Adaptive Detection and Tracking using Multimodal Information

Ciarán Ó Conaire, B.Eng.

School of Electronic Engineering &
The Centre for Digital Video Processing

Dublin City University

Supervisor: Dr. Noel E. O'Connor

A thesis submitted for the degree of

Doctor of Philosophy

September 2007

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

ID No.: 99426439

Date:

Ba mhaith liom an tráchtas seo a thiomniú do mo thuistí
Rhóda agus Breandán Ó Conaire

Bhíos an-óg nuair a rugadh mé, ach bhí an t-ádh orm, mar thugadar aire
agus grá dom, agus ní dóigh liom go mbeidh a leithéidí arís ann.

Acknowledgements

There are a number of people to whom I am greatly indebted for their help, inspiration and patience. First of all I would like to acknowledge and thank my supervisor, Dr. Noel O'Connor, for his guidance, his advice and his ability to quickly resolve any issues I sought his help on. I would also like to thank him for giving me the freedom and the time to pursue the many paths this research has followed.

I would like to thank the Irish Research Council for Science, Engineering and Technology (IRCSET) and Science Foundation Ireland (SFI) for providing the funding to pursue this PhD work.

I would like to express my gratitude to my colleagues in the Centre for Digital Video Processing, and especially to Tomasz Adamek, Sorin Sav, Philip Kelly, Mike McHugh, Noel Murphy and Alan Smeaton. It has been a pleasure to work within the group with these people on so many interesting projects.

And finally, to all those people who helped keep me sane for the past year, a big thank you to Michael Cotter, Charlie Daly, Marian Flanagan, Susi Ghassempur, Ailbhe Jordan, Colm Mac Oireachtaigh, John Monaghan, Declan Nolan, Aisling Ní Chonaire, Cormac Ó Conaire, Aonghus Ó Conaire, Lisa O'Reilly, Karl Podesta and Sudanthi Wijewickrema.

Abstract

This thesis describes work on fusing data from multiple sources of information, and focusses on two main areas: adaptive detection and adaptive object tracking in automated vision scenarios. The work on adaptive object detection explores a new paradigm in dynamic parameter selection, by selecting thresholds for object detection to maximise agreement between pairs of sources. Object tracking, a complementary technique to object detection, is also explored in a multi-source context and an efficient framework for robust tracking, termed the Spatiogram Bank tracker, is proposed as a means to overcome the difficulties of traditional histogram tracking. As well as performing theoretical analysis of the proposed methods, specific example applications are given for both the detection and the tracking aspects, using thermal infrared and visible spectrum video data, as well as other multi-modal information sources.

List of publications

- C. Ó Conaire, N. E. O'Connor, and A. F. Smeaton. Detector adaptation by maximising agreement between independent data sources. In IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum, June 2007.
- C. Ó Conaire, N. E. O'Connor, and A. Smeaton. Thermo-visual feature fusion for object tracking using multiple spatiogram trackers. Journal of Machine Vision and Applications, May 2007.
- C. Ó Conaire, N. E. O'Connor, and A. F. Smeaton. An improved spatiogram similarity measure for robust object localisation. In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), April 2007.
- C. Ó Conaire, N. E. O'Connor, A. Smeaton, and G. J. F. Jones. Organising a daily visual diary using multi-feature clustering. In Proc. of 19th annual Symposium on Electronic Imaging, Jan 2007.
- C. Ó Conaire, N. E. O'Connor, E. Cooke, and A. F. Smeaton. Multispectral object segmentation and retrieval in surveillance video. IEEE International Conference on Image Processing (ICIP), Oct 2006.
- C. Ó Conaire, N. E. O'Connor, E. Cooke, and A. F. Smeaton. Comparison of fusion methods for thermo-visual surveillance tracking. In International Conference on Information Fusion, July 2006.
- C. Ó Conaire, N. O'Connor, E. Cooke, and A. Smeaton. Detection thresholding using mutual information. In VISAPP: International Conference on Computer Vision Theory and Applications, Setúbal, Portugal, Feb 2006.
- C. Ó Conaire, E. Cooke, N. O'Connor, N. Murphy, and A. Smeaton. Background modelling in infrared and visible spectrum video for people tracking. In 2nd Joint IEEE International Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum (OTCBVS), San Diego, CA, USA, June 2005.
- C. Ó Conaire, E. Cooke, N. O'Connor, N. Murphy, and A. F. Smeaton. Fusion of infrared and visible spectrum video for indoor surveillance. In International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Montreux, Switzerland, April 2005.

- G. M. P. O'Hare, M. J. O'Grady, R. Jurdak, C. Muldoon, N. O'Connor, C. Ó Conaire and P. Kelly. Engineering Ambient Visual Sensors. IMAI'07 - 5th International Conference on Intelligent Multimedia and Ambient Intelligence, Salt Lake City, Utah, 22-24 July 2007.

- D. Byrne, P. Kehoe, H. Lee, C. Ó Conaire, A. F. Smeaton, N. O'Connor and G. Jones. A User-Centered Approach to Rushes Summarisation Via Highlight-Detected Keyframes. TVS 2007 - TRECVID BBC Rushes Summarization Workshop, ACM Multimedia 2007, Augsburg, Germany, 24-29 September 2007.

Contents

List of Figures

Chapter 1

Introduction

Detection and tracking are two essential components of automated vision systems in many application areas, specifically in scenarios involving people. These areas of application include automated surveillance, human-computer interfaces, human body tracking, automated monitoring of the elderly and meeting activity analysis. Traditional approaches, using only a single visible spectrum camera as an input device, face many difficult problems such as ambiguities due to occlusion, camouflage due to similar properties of the objects and the background and changing lighting conditions.

Thermal infrared cameras are very powerful devices for monitoring people and can overcome many of these problems. Due to advances in technology, these devices have become cheaper and more widespread in recent years, a trend that is expected to continue. Firstly, they detect emitted thermal radiation, so are robust to fast lighting changes. Secondly, people are easily distinguished from the surrounding background clutter due to their large temperature differences. However, they have their own drawbacks, including an inability to distinguish similar temperature objects and a low image quality due to high noise, as well as technology-specific problems such as the *halo-effect* and periodic shuttering.

In recent years, research efforts have focussed on using multiple input devices to mitigate the problems associated with traditional methods of visual analysis. In this vein, the combination of thermal infrared video data with traditional visible spectrum data would seem like a worthwhile endeavor for a number of reasons. Firstly, these sources are inherently complementary, as visible imagery is generated primarily from reflected radiation, whereas thermal infrared images are mainly caused by emitted radiation. Secondly, using multiple sources of data allows better discrimination of target objects from the background clutter. For example, while an object may resemble the background in one data source, it is unlikely to resemble it in all other sources. And thirdly, a suitably formulated combination of these sources allows their strengths to be exploited and their weaknesses overcome. The properties of thermal infrared can make a visual analysis system robust to fast lighting changes, while its use of visible spectrum information allows it to exploit the richness of multi-band colour information.

The main contributions of this thesis are to develop general-purpose techniques for fusing information from multiple sources of data, specifically targetting the fusion of visual data with data from thermal infrared video imagery. This work's contribution is twofold. Firstly, a new paradigm in dynamic thresholding is proposed that is termed *mutual information thresholding*. This technique is a general method of selecting parameters for multiple detection modules in order to maximise their agreement. Numerous applications using real-world data are demonstrated, including foreground object detection in multimodal video, event detection using audio-visual data and skin detection in thermo-visual video.

Secondly, the recently proposed spatiogram-based tracker is extended in order to efficiently handle multiple data sources. This tracking framework is shown to outperform traditional trackers in a variety of difficult tracking scenarios. As well as extending it to handle multiple data sources, an architecture is proposed for dynamically weighting the various sources, so as to best distinguish the object from the background clutter, providing robust tracking.

1.1 Thesis overview

In chapter 2, a literature review is conducted on a broad range of topics relevant to the work in this thesis. As this work focusses on combining information from multiple sources, specifically visual and thermal infrared sources, prior work in using thermal infrared in automated visual applications is reviewed. Next, previously described uses of combined visual and thermal infrared imagery are explored, as well as a general discussion on the broad families of techniques that have been widely used for data fusion. To set the context for this thesis' contribution to object and event detection, detailed in chapter 3, previous research on scene background modelling and unsupervised dynamic thresholding is described. Finally, with regard to the work described in chapter 5 and chapter 6, a review of relevant prior work in object tracking is conducted, along with a discussion of the open issues in the field.

The next two chapters concern the contribution of this work to dynamic object and event detection. In chapter 3, the proposed approach of *mutual information thresh-*

olding for automated detection of events and objects is described and evaluated using synthetic data and compared to the leading dynamic thresholding algorithms. Chapter 4 continues this evaluation using real data from publicly available datasets, as well as data captured privately using the developed thermo-visual camera rig. Numerous applications of this technique are demonstrated.

Next, chapters 5 and 6 describe the work's contribution to general object tracking. Chapter 5 introduces the concept of a Spatiogram, which generalises the commonly-used histogram, and then details the proposed extension of the spatiogram for efficient multi-modal tracking. Chapter 6 details the work on adaptive tracking, which uses the spatiogram bank tracking framework presented in chapter 5. The experiments on adaptively combining features to best separate the object from the clutter demonstrate robust tracking on a variety of difficult thermo-visual tracking sequences.

In the final chapter, the contents of the thesis is summarised and a detailed discussion of the work is conducted, providing a roadmap for future work leading on from the experiments conducted here.

Chapter 2

Related Research

2.1 Introduction

This chapter covers a broad range of computer vision topics, giving an overview of the methods related to the core work in this thesis. Firstly, thermal infrared technology and research are explored, illustrating its advantages over visible spectrum analysis. The discussion is then expanded to multi-modal analysis by examining research on data fusion, particularly in fusing visible spectrum and thermal infrared. Background modelling and automatic threshold selection approaches are reviewed, with a view to prepare the groundwork for developing our new technique for *mutual information thresholding*. Similarly, a broad review of the extensive literature on object tracking is conducted as a precursor to this thesis' contribution to work in this field using adaptive banks of spatiograms. Each section concludes with an overview of research in the field and discusses open problems, future research directions, as well as how the work in this thesis makes a contribution to the research goals.

2.2 Thermal Infrared

Infrared (IR) radiation covers a large span of the electromagnetic spectrum, beginning just beyond the red portion of the visible spectrum. The visible spectrum region lies only between 400 and 780nm, whereas the infrared region spans 780nm up to 100,000nm. As this region is so large, infrared radiation can have different properties depending on the section in which it lies. It is also often subdivided into smaller ranges. The exact names for these sub-bands vary in the literature. The most common classification is as follows: The portion just outside the visible region (780nm to 1.3μm) is referred to as near-IR (NIR) or short-wave infrared (SWIR). Other region classifications are mid-wave IR (MWIR; 3 to 5μm), long-wave IR (LWIR; 8 to 14μm), and very-long-wave IR (30μm and above). The term *thermal infrared* is generally used to refer to LWIR, but may also apply to MWIR. Why these particular sub-band regions are used is due to atmospheric absorption and is discussed further below in reference to figure 2.2. Forward-looking infrared (FLIR) is another term used, primarily in North America, to refer to imagers of thermal infrared radiation.

Most of the visible light that is seen by human eyes is reflected light. The primary source of visible light is the sun or artificial lighting. The complex array of colours we can see comes from the reflection of some subbands of the white light that falls on objects, while the rest is absorbed. Thermal infrared on the other hand is mostly emitted radiation and as such is inherently complementary to the visible spectrum. Planck's law of black-body radiation approximates the emitted electromagnetic radiation from an object at a given temperature, T. His law is given by:

$$I(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{(hc)/(kT\lambda)} - 1} \tag{2.1}$$

where λ is the radiation wavelength and h,c and k are physical constants, namely Planck's constant, the speed of light and Boltzmann's constant respectively. Figure 2.1 demonstrates some emission spectra for black-body objects of various temperatures. The peak radiation emission occurs in the thermal infrared band for objects at room temperature.

In terms of its transmission through air, figure 2.2 shows the fraction of thermal radiation that passes through the atmosphere at sea-level over a distance of 1km. Certain wavelengths are absorbed by chemicals in the air, such as water and CO_2. As can be seen from the graph, there are two spectral windows that transmit radiation with low absorption: 3-5μm and 8-14μm. This accounts for the sub-band classification into MWIR and LWIR, as they account for distinctly separate wavelength bands.

2.2.1 Thermal imaging technology

History of IR technology Research into thermal imaging began in the military domain, with potential to improve combat performance in many areas. Firstly, thermal imaging works in total darkness, therefore giving the ability to literally *see in the dark*. It also has the ability to penetrate smoke, which is useful on the battlefield. Secondly, military vehicles and aeroplanes, when operating, as well as combat personnel, usually have a significantly different temperature to their surroundings, allowing them

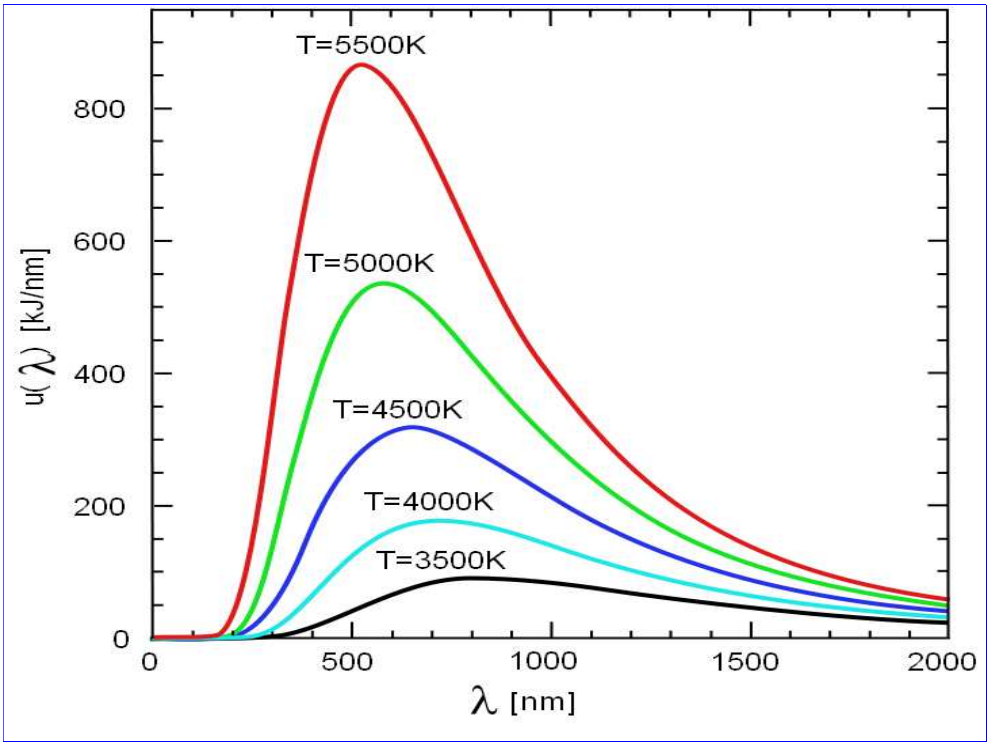


Figure 2.1: Planck's law of blackbody radiation, which governs the emission spectrum of a black-body object at a fixed temperature, T.

to be detected, despite the use of camouflage, which only conceals them in the visible spectrum.

In the late 1950s and 1960s, three companies, Texas Instruments, Hughes Aircraft, and Honeywell developed single element detectors that produced thermal line images by scanning scenes. The technology was not commercialised until decades later due to the high costs and because it was classified due to the sensitive military nature of its applications. However, these basic detectors led to the development of modern thermal imaging devices [14].

In the late 1980s the United States federal government awarded large classified contracts, known as HIDAD (HIgh-Density Array Development) contracts, to two companies, Texas Instruments and Honeywell. Their purpose was the development of uncooled infrared sensor technology with a very short turn-on time, which would make thermal imaging useful for practical military applications, unlike the earlier cooled systems and line-scanning devices. Both companies produced successful devices: the pyroelectric

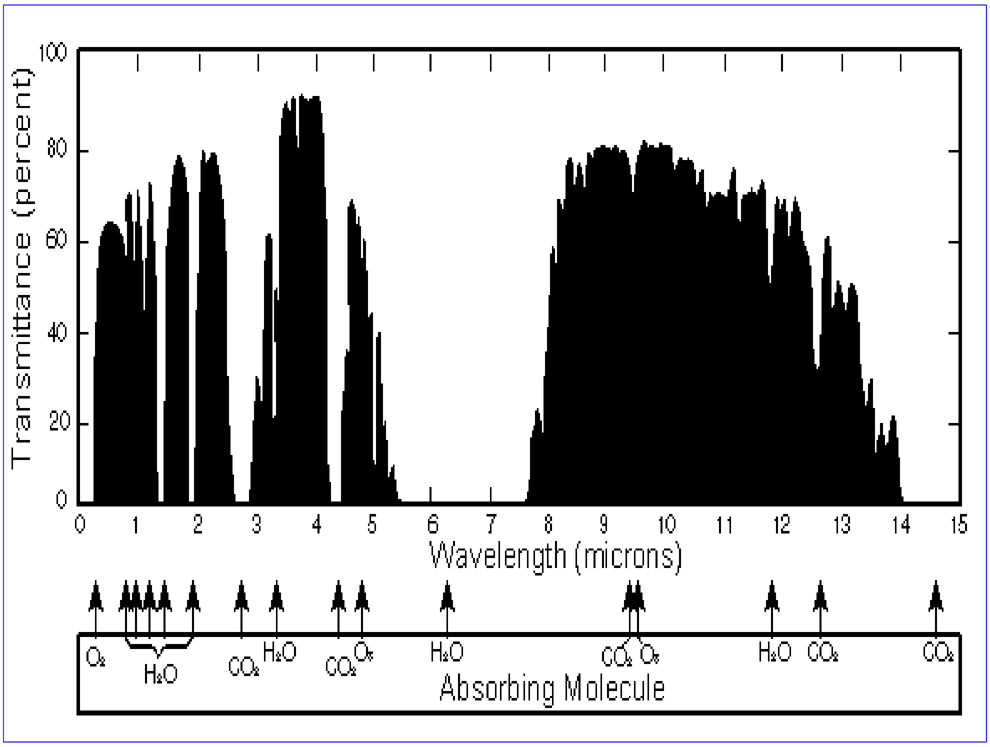


Figure 2.2: Transmittance of infrared radiation through air at sea-level. Various chemical compounds in the atmosphere absorb significant amounts of infrared in certain bands, making these wavelengths impractical for long-range imaging (Source: Wikipedia; a similar graph appears in [64] pg. 23).

sensor using using barium strontium titanate (BST) by Texas Instruments [43] and the microbolometer using vanadium oxide (VOx) by Honeywell [14] [134].

In 1994 Honeywell was granted a patent on their microbolometer technology. Honeywell licensed their microbolometer sensor technology to other companies. Originally, four companies bought licenses for VOx technology from Honeywell but these licences have since changed hands and other companies have since purchased licences. Companies that are (or were) involved in thermal imaging include Raytheon, Boeing, Lockheed-Martin, British Aerospace (BAE), Lorcal, Rockwell, Santa Barbara Research, DRS Technologies, Indigo Systems, InfraredVision Technologies Corporation, NEC, and Institut National dOptique [14] [134].

In 1992 the US Government de-classified the use of Infrared Technology for commercial products but maintained control of the technology [134]. Large thermal-imaging manufacturers, such as Raytheon and Lockheed-Martin, are allowed to sell their devices to foreign countries but not however to divulge their manufacturing techniques. The current ban applies to two of the three uncooled IR technologies, microbolometer and pyroelectric/ferroelectric, but exempts thermoelectric devices [44].

Modern uncooled technology Thermal images are inherently very noisy. This is due to the fact that every object above absolute zero temperature emits thermal radiation, including the camera itself! Early thermal imagers (as well as some modern systems) require cryogenic cooling systems to prevent the capture device from interfering with the imaging process. A thorough review of cooled infrared imaging systems can be found in [120].

Technological advances have made it possible to capture thermal imagery without cooling systems. These cameras are known as *uncooled* detectors and the two dominant imaging technologies in this field are pyro/ferro-electric and micro-bolometer. Both technologies rely on the thermal radiation to produce a measurable change in a property of each element of the focal plane arrays; a change in capacitance, in the case of the ferroelectric sensor, and a change in resistance, in the microbolometer. The images produced by both technologies are visually quite different. As well as this, there are a number of other differences that should be taken into account when choosing a technology to use in a real application.

Microbolometer sensors have the advantage that they can measure the absolute temperature of pixels in a scene. Ferroelectric sensors suffer from pixel *crosstalk* causing a *halo-effect* so cannot measure absolute temperature. This halo appears as a bright glow around dark objects and a dark glow around bright objects. This is explained further below. Disadvantages of microbolometer sensors are that they must perform periodic shuttering of the system, which may last from a few seconds to a few minutes, and results in the image freezing during shuttering. Also, fixed-pattern noise may appear on the imagery due to DC-drift. The ferroelectric technology does not have this problem because AC-coupling is used.

The halo effect (also known as pixel cross-talk) is caused by the *chopper*, a spinning blade within the camera housing, that acts as a reference temperature source for each pixel. The pixels will measure the difference in temperature between the scene and the chopper and will return this value. The assumption is that the chopper is of uniform temperature. The assumption is often violated when there are hot objects in the scene that emit radiation onto the chopper, causing it to heat up and this heat diffuses through the chopper. The end result is that pixels beside hotter pixels will appear colder and pixels beside colder pixels will appear hotter, producing the halo-effect. Two examples of this effect are shown in figure 2.3.

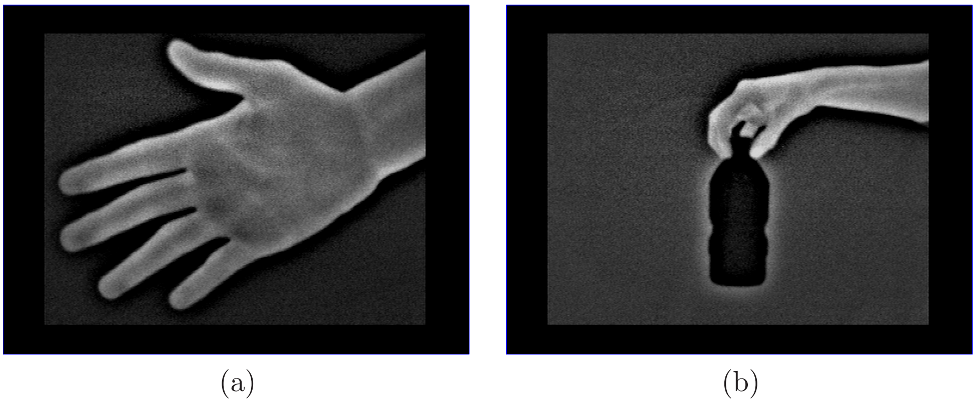


Figure 2.3: Examples of the halo-effect that occurs with hot and cold objects in images captured by pyrolectric thermal imagers.

2.2.2 Real-world applications

Thermal infrared technology has found a diverse range of real-world applications. These include governmental uses (for military, law-enforcement and anti-terrorism), industrial applications (predictive maintenance for early-failure-warning on mechanical and electrical equipment, process monitoring, evaluation of insulation) and medical applications (patient diagnosis, thermal abnormality detection), as well as in a host of other fields such as aerial archaeology and pollution effluent detection.

In [84], applications of thermal imaging are detailed, such as non-destructive testing (NDT) of products, buildings and structures maintenance (heating networks, sewer systems, waste-water pipes, water canals), quality control of insulation, moisture damage detection, detection of cracks in exterior walls, air-tightness testing, wastewater pipe damage detection, detection of road surface deterioration, printed-circuit board (PCB) testing, anomaly detection in the electric utilities and nuclear power industry, real-time weld control, military applications of automatic target recognition (ATR) systems, guided ordinance, weapon sights and thermal imagers on the AH-64A Apache helicopter, many applications in the aerospace industry, and forest fire detection.

More applications are given in [15], such as the detection of vapour and gas leaks, oil pollution control, inspection of machinery in chemical, petrochemical and steel industries, inspection of electronics, inspection of vehicle tyres, testing of buildings (thermal losses), land survey applications using specialised satellites (Landsat, ERTS, HCMM, SPOT), medical (identification of varicosities, assessment of arterial disease, therapy monitoring) and stress analysis in structures.

Further details are given in [64] of applications such as electrical monitoring (detecting inductive currents, energised grounds and open circuits), identifying excessive friction, pipe blockage, moisture detection in buildings, and industrial process monitoring and control.

2.2.3 Thermal infrared research

The various research directions of infrared imaging are now briefly reviewed. The dominant areas of research are in the military, medical and security (surveillance) domains.

Since the technology for thermal imaging originated in military research labs, many of the uses of thermal technology are for defence related applications. Especially useful for battlefield scenarios, thermal radiation penetrates smoke and fog, thereby providing the ability to locate opposing units. Its lighting independence allows it to be used to total darkness, where visible spectrum-based methods are ineffective. The main advantage of thermal technology is that is allows objects to be detected by their temperature difference from their surroundings. This is ideally suited to detecting people and moving vehicles, as they are usually have significantly different thermal properties than the environment. In [90], a system is described for recognising military vehicles in thermal imagery. Forward-looking infrared (FLIR) is used in heat-seeking missiles, such as the AIM-9 Sidewinder missile that is carried on fighter aircraft for air-to-air combat. It is named after the Sidewinder snake which has the ability to detect thermal radiation in its pit organ [93].

In the medical field, thermal imaging has been cited as a useful and non-invasive diagnostic tool [61]. Usually, uncooled cameras are not effective enough to be used in medical applications where precise temperature measurement is required, so more expensive cryogenic cooling systems are needed. Thermal imagery can be used by medical personnel to identify abnormal heat patterns that may indicate maladies. These maladies include the identification of varicosities, assessment of arterial disease and therapy monitoring [15].

However, the usefulness of thermal infrared for medical applications has been questioned [84]. While it is stated that *there is renewed interest in using infrared imaging to detect breast tumors* [61], in a thorough review of the effectiveness of infrared thermal imaging for breast cancer screening and testing, [68] Kerr concludes that *"The evidence that is currently available does not provide enough support for the role of infrared ther-*

mography for either the population screening or adjuvant diagnostic testing of breast cancer". Her conclusions are consistent with recommendations of the Royal Australian and New Zealand College of Radiologists Breast Imaging Reference Group who do not recommend the use of thermography for the early detection of breast cancer in their 2001 policy.

In considering the use of image processing techniques for thermal images of human skin [61], Jones and Plassmann state that "Care is needed in interpreting thermograms since they are nonspecific and may reveal past traumas as well as current problems" and that "Standard protocols are necessary to produce repeatable and meaningful thermograms". Jones [60] also states that thermal imagery *"is nonspecific"* and that *"patterns of temperature need to be interpreted by a trained eye. A hot or cold spot in the temperature distribution may be an indicator of the presence of a tumor forming within the body, or of inflammation as in the case of arthritic joints, of infection, of loss or over-activity of sympathetic nerve function, or of a host of other dysfunctions"*. These kind of statements do not add much credibility to the use of thermal imagery for widespread use in medical diagnosis.

As people are usually of significantly different temperature to their surroundings, thermal imaging is an ideal technology for surveillance of people and their activities. In [28], Davis and Sharma present a new background-subtraction algorithm for thermal imagery that uses contour information to overcome to problem of the halo-effect. A contour saliency map is formed using foreground and background gradients. This map is then thinned and broken contour segments are completed by combining the watershed algorithm with a path-constrained search. People silhouettes are produced by flood-filling the contour image. In [29] a template-based method is presented for detecting people in thermal imagery from different seasons. First, a fast screening procedure is used, using a generalized template to locate potential person locations. Finally, an AdaBoosted ensemble classifier tests the hypotheses to determine if people are present in the scene.

One particular application of person detection that has inspired much research is in the area of pedestrian safety. Given the high number of night-time road fatalities due to vehicles colliding with pedestrians, efforts have been underway to use thermal imaging to automatically detect people and warn drivers of their presence [10, 34, 91, 156].

Of the many other applications of thermal imaging, a few are mentioned here. Thermal imagery has been used for automatic face detection [136], as well as face tracking [32]. Face recognition is also found to be improved by using additional spectral information from near-IR bands [100]. Another important piece of research to mention is the work of Lin on extending the current computer vision techniques, that were designed with the visible spectrum in mind, to infrared band images [79].

2.2.4 Discussion: the future of thermal imaging

Widespread adoption of thermal infrared technology may not be as fast as many predict. The adoption by Deville of Raytheon's NIGHTDRIVER thermal vision system in their 2000 model is often cited as evidence of its commercial viability. The sales of this model fell quickly however, with only 600 systems sold in 2004 and it was finally dropped in 2005. Some believe the next step in thermal technology development is the mass-production of low-cost, low-resolution thermal imagers, for wide-area distributed sensing and efforts are already underway in this direction [44].

The introduction of a workshop on Object Tracking and Classification in and Beyond the Visible Spectrum (OTCBVS) in 2004 at the IEEE Conference on Computer Vision and Pattern Recognition is a sign of the increased levels of research on thermal imaging. The workshop has continued each year and is now in its 4^{th} year.

Thermal infrared imaging has been shown to be a very useful modality in many areas of computer vision research. As thermal radiation comes primarily from emitted radiation, it is very robust to adverse lighting condition that plague many visible spectrum vision systems. It is also very useful for distinguishing people and vehicles from complex backgrounds.

The drawbacks of thermal imaging, such as the high levels of image noise, the halo effect in ferroelectric cameras and its inability to distinguish objects of similar temperature, such as people, may limit its applicability in many application areas. It would seem plausible therefore to try to leverage the advantages of both thermal imagers and traditional visible spectrum cameras, to act as complementary sources of data and to overcome the limitations of both these data sources simultaneously, thereby improving the performance of vision systems by using multi-modal input, instead of opting for one or the other. In the next section, the various research directions on combining these two modalities are reviewed.

2.3 Combining infrared and visible

This section reviews research on combining thermal infrared and visible spectrum information to improve the performance of computer vision systems. The capturing of this multi-modal data is first examined, then the discussion proceeds to how it is aligned and finally exploring the uses to which this rich data has been put. Also detailed is the hardware and the procedure used to capture thermal and visual data for this work.

2.3.1 Capture rigs

In order to jointly capture thermal infrared and visible spectrum imagery, most approaches use two separate cameras. A single camera that provides capture and alignment of both modalities is commercially available from Equinox Sensors, although it may be prohibitively expensive for some projects. When two separate cameras are used, they can either be in a beam-splitter (BS) configuration or a semi-parallel (SP) setup. The BS configuration is where the cameras are positioned at ninety degrees to each other with a semi-reflective medium placed between them so that they appear to have the same optical axis. This medium will reflect one type of radiation (e.g. infrared) and will be transparent to the other (e.g. visible light). The SP configuration is where both cameras are pointing in roughly the same direction and see a similar scene, but due to parallax effects, there may be some parts only seen by one camera. The BS configuration is superior to the SP setup, but is more difficult to construct, as it requires a beam-splitter medium that will transmit visible radiation but reflects thermal (or vice versa). The SP setup is most useful when the scene is far enough away from the cameras so that it can be assumed to be planar. BS configurations have been used in [159] and in this work. A SP configuration was used in [28].

In the thermo-visible capture rig used for this work a Raytheon Control-IR 2000B thermal imaging video camera was used, along with a Panasonic WV-CP470 colour video camera. The thermal camera is sensitive to wavelengths in the range 7μm-14μm. The frame capture times of both cameras were synchronised using a *gen-lock* signal to ensure that they captured images simultaneously. Both channels of analogue video output are captured and digitised by a Falcon Quattro multi-channel frame-grabber. A pane of thermally-reflective glass was used to act as a beam-splitter. Figure 2.4 shows the configuration of the visible and thermal cameras.

2.3.2 Data Alignment

A number of different methods have been proposed in the research literature on how to automatically align multimodal imagery, many of which are based on maximising the mutual information between the two sources [104, 150]. This is especially useful for medical imagery taken from two different scanners. It is usually the case that there are a small number of dominant pixel classes in the data (corresponding to different

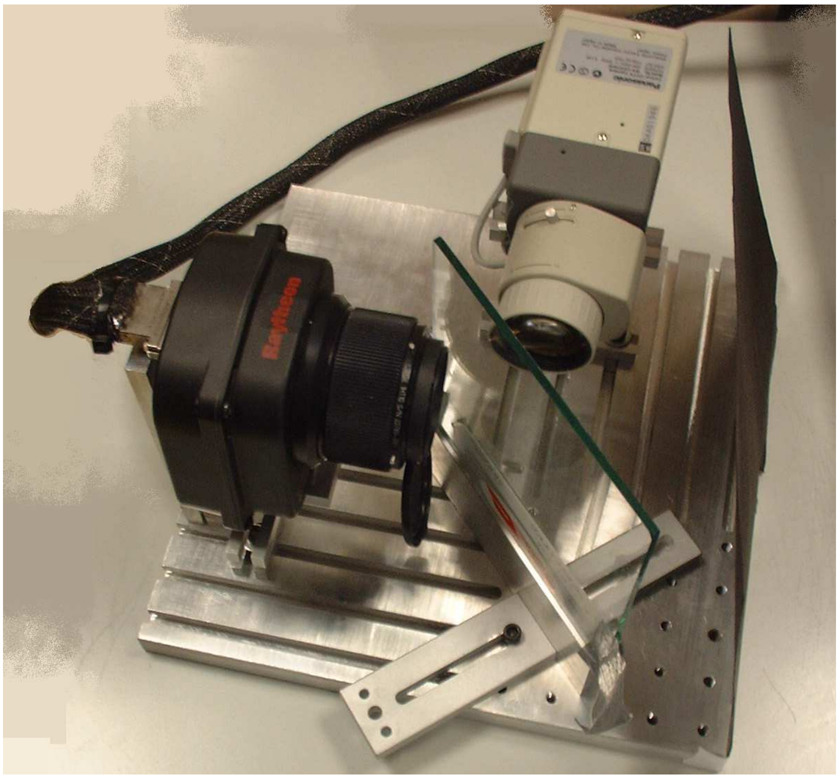


Figure 2.4: Thermal and visible capture rig

skin tissues) and therefore there is strong statistical correlation between the brightness values in both modalities. For general thermal and visible imagery, this is not usually the case. In [54], a more general method to align multimodal data is described which uses a block-based cross correlation approach to maximise edge orientation similarity between the modalities, since this is one feature that is common to both modalities.

In this work, to align pixels in the thermal and visible spectrum, an appropriate planar homography [46] is determined and this image warping is applied to all thermal infrared frames in order to align their pixels with the corresponding pixels in the visible spectrum image. This homography is determined by manually selecting many corresponding points in both modalities and computing the homography with least-squared-error. There is no correlation between visible spectrum brightness and thermal infrared brightness values, so many of the automatic mutual-information based alignment methods would not be appropriate. With enough edge information in the scene, the unsupervised alignment method of [54] using edge orientation similarity could be used. However, as only one warping needs to be computed for a fixed camera rig, no automatic method is required, but automating the process using such a method could be useful and is a direction for future work.

2.3.3 Image fusion

By capturing images of a scene using different spectral bands, features that may be undetectable in one band can be seen in another. This seems like an obvious benefit of jointly capturing thermal infrared and visible spectrum imagery. However, in order to present multimodal imagery or video to humans, the amount of information must be reduced in order for it to be processable by the human visual system. *Image fusion* refers to the reduction of a hyper-spectral (multi-band) image to a single band (or three-band colour) image for visual inspection. The goal of image fusion is to retain, in the *fused* image, the useful information contained in all bands. The success of an image fusion procedure can be assessed using perceptual tests; for example, by gauging the speed or accuracy at which a user can locate objects in the scene presented in the fused imagery. Multi-scale decomposition-based image fusion schemes have been shown to perform well for image fusion [160]. A method for the fusion of multimodal video frames is proposed and evaluated in [114].

In this thesis, the targetted applications involve automated vision data processing, so the techniques of image fusion are not considered since they must, by necessity, discard potentially useful information. Any practical system for displaying complex data to humans must reduce the information burden on the user by discarding some of the data. Automated systems, on the other hand, have no such restriction.

2.3.4 Surveillance

The use of multispectral data for automated surveillance is another area that has received much attention. In a review of video surveillance and sensor networks research [25], Cucchiara argues that the integration of video technology with sensors and other media streams will constitute the fundamental infrastructure for new generations of multimedia surveillance systems. Also reviewing surveillance research [50], Hu et al. conclude in their section on Future Developments in Surveillance that *"Surveillance using multiple different sensors seems to be a very interesting subject. The main problem is how to make use of their respective merits and fuse information from such kinds of sensors"*.

Using low-resolution thermal images, Jones et al propose a Bayesian approach for combining them with high-resolution visual information for the surveillance of *sterile* *zones where an alarm is required should a person enter the zone* [62]. Their aim is to reduce the high false alarm rate present when using the visual images only, due to changing environmental conditions. They show how the low-resolution thermal images can suggest an area of analysis and then Markov Random Fields are used to segment the highlighted object.

A surveillance system that detects foreground regions in infrared and visual imagery is described in [141]. Object regions are detected and tracked separately in each modality and then a series of merging rules are used associate regions between the modalities.

In [40], Goubet et al. illustrate quite well the advantages and disadvantages of thermal and visible images for daytime pedestrian detection and tracking. However their very simple fusion method does not improve performance over using infrared alone.

In [27], Davis and Sharma build on their earlier work which extracted person contours from infrared images only and investigate how the addition of visible spectrum information improves the person segmentation performance. Using regions of interest obtained from background modelling, contour segments are extracted from these regions in each modality using a thinned contour saliency map. The contour segments in each modality are aligned and completed by performing a path-constrained search on the watershed boundaries. Person silhouettes are produced by flood-filling the completed contour image.

Sharma and Davis [124] continue their work in this vein, but use a mutual information based approach to choose the contour segments in one modality (visual) in such a way as to maximise a mutual-information-based measure between these contours and detected contours in the thermal image. The contour selection is based on a heuristic selection scheme. Results on segmenting people from the background are quantitatively evaluated using manually segmented ground truth and shown to outperform either visual or infrared analysis alone.

2.3.5 Face recognition

Face recognition research is another area of computer vision that has benefitted from research that combined thermal infrared and visible spectrum imagery.

Heo et al. use thermal images of faces to detect, and subsequently remove, eyeglasses from the visual images [49]. This is shown to improve face recognition performance.

In a similar study of face recognition, Chen et al. compare visible only, infrared only and combined face recognition results, using both principal component analysis (PCA) and a commercial face recognition system [18]. They conclude that *"multimodal IR and visible recognition has the potential to improve performance over the current commercially available state of the art."*

Also in work related to multispectral face recognition, Selenger and Socolinsky [121] conclude from their experiments that *"It becomes clear from our analysis, that LWIR imagery of human faces is not only a valid biometric, but almost surely a superior one to comparable visible imagery."* Though they add a note of caution that perhaps their thermal images were not challenging enough for the recognition task.

2.3.6 Discussion

The combination of thermal infrared and visual data has shown to be useful in a wide variety of application areas, usually out-performing single modality analysis. Future directions for the fusion of visible and thermal infrared information lie in providing automatic mechanisms of adaptation for these systems to cater for changes in the multi-modal data due to situational or environment changes.

We will examine, in the next section, the main schools of thought with regard to fusing the data from multiple sources of information.

2.4 Data fusion

The exact definition of term *data fusion* is often debated. On a website dedicated to the subject (http://www.data-fusion.org/), in an article discussing the search for a concise meaning of data-fusion, the author admits *"it is very difficult to provide a precise definition of data fusion"* and that *"If one looks for a definition, one will be stunned by the poverty of the few definitions, the lack of clarity and consensus, and by the battle of words. The exact meaning of data fusion [varies] from one scientist to another"*. Fusion could, for example, refer to *image fusion* as mentioned in section 2.3.3. Generally, the acid-test for a desired fusion system is that it should ensure that no single data source provides better information than the fused combination of all data sources. In this thesis, the term fusion, or data fusion, is used to refer to *the combination of evidence from multiple sources of data in an attempt to improve the accuracy of decisions made on the basis of this evidence.*

In this section, the two most mature approaches for evidence fusion are briefly reviewed; namely, Bayesian fusion and fusion using the Transferable Belief Model. We also review other commonly used ad-hoc approaches to data fusion.

2.4.1 Bayesian fusion

Bayesian fusion refers to the use of Bayes theory, which is strongly founded classical probability. In Bayesian decision making, probability distributions are used to make decisions on the most likely possibility. These distributions must be known beforehand and are usually computed from *training data*, which is a large collection of previously annotated examples.

To provide a more concrete understanding of the Bayes approach, we look at a simple classification example. We have uncovered a rib bone from some animal, but are unsure of which species it belongs to. However, it has been narrowed down to three possibilities: $C = \{Rat, Squirrel, Possum\}$. We can further evaluate how likely each species is using some features of the bone. These features are $F = \{length, density, smoothness\}$. We the use Bayes' rule, which is stated as

$$P(C_k|f) = \frac{P(f|C_k)P(C_k)}{P(f)} = \frac{P(f|C_k)P(C_k)}{\sum_i P(f|C_i)P(C_i)} \tag{2.2}$$

where f represents the measured values of the feature set and C_k is the object class. Intuitively, this equation provides a probability of the bone belonging to species C_k , given the three measurements of the bone features, f. The values of $P(C_k)$ are known as the *priors* and are the probabilities of each species before measurements are taken. In this example, the priors would be computed from the relative population sizes of each species, with $P_{rat} + P_{squirrel} + P_{possum} = 1$. For example, *possums* are quite rare in Ireland, so if the bone were found in Ireland, then $P_{possum} << P_{squirrel}$. In the absence of other information, priors are often assumed to be equal. Annotated training data is required in order to have the probability distributions from which the values of $P(f|C_k)$ are obtained. In this example, a large number of rib bones from all three species would have already been collected and their features measured. Using such training samples, the probability distributions can be estimated using histograms or neural networks [41].

In order to avoid high dimensional density estimation in fusing multiple features, it is sometimes valid to assume that the features are independent. If the feature set F has three distinct features such that $F = \{F_1, F_2, F_3\}$, then the assumption of independence leads to

$$P(f|C_k) = P(f_1|C_k)P(f_2|C_k)P(f_3|C_k). \tag{2.3}$$

This simplifies the training stage, since much less samples are required to estimate one-dimensional distributions accurately compared to higher-dimensional distributions.

2.4.2 Transferable Belief Model based fusion

Over a decade ago, Luo and Kay [83] reviewed the many approaches to multi-sensor integration. They concluded that the methods used for modeling the error or uncertainty are central to formulating a general methodology for multisensor integration.

The Transferable Belief Model (TBM) concept introduced by Smets [129] explicitly models the error, or uncertainly, associated with data fusion. Specifically, the uncertainty is catered for by dividing it into two classes: imprecision and ignorance. Imprecision refers to the data returned from sensors and fuzzy-set theory can be used to account for sensor imprecision. Ignorance refers to missing information, a lack of knowledge or ambiguity between alternative hypotheses. As an example, if Bayes theory assigns probabilities to the two possible propositions (outcomes), the TBM assigns a *belief mass* to the same propositions, but also to the "unknown" proposition, which represents total ignorance of the true outcome.

The TBM defines Δ as the set of possible outcomes and Ω as the power set, which contains sets of all the combinations of items in Δ, therefore Ω has $2^{|\Delta|}$ elements. A belief mass function m is defined such that

$$\sum_{A \in \Omega} m(A) = 1. \tag{2.4}$$

To fuse the belief masses of two sensors, or sources of information, Dempsters rule of combination gives the fused belief mass function as

$$m(C) = \sum_{A \cap B = C} m_1(A) m_2(B) \tag{2.5}$$

The TBM concept is derived from Dempster-Shafer theory and the two terms are sometimes used inter-changeably. The main difference is that no normalisation takes place in the TBM. In the Dempster-Shafer theory, the belief masses are normalised, so that the belief mass of the null proposition $m(\emptyset) = 0$. The normalised version of the fusion equation is shown in equation (2.6). This normalization corresponds to the *closed-world* assumption, that the evidence may support only propositions from Ω. A discussion of the problems associated with normalisation is given in [129], where he argues that the belief mass of the null proposition, $m(\emptyset)$, should be interpreted as evidence of a possible outcome outside of Ω (the *open-world* assumption).

$$m(C) = \frac{\sum_{A \cap B = C} m_1(A) m_2(B)}{1 - \sum_{A \cap B = \emptyset} m_1(A) m_2(B)} \tag{2.6}$$

In terms of how sensor data is mapped onto the belief masses, this is usually done using fuzzy sets. These fuzzy sets are created using statistical or expert knowledge and can be an intuitive representation of the belief given the sensor's value.

A comparison between the TBM and Bayes theory for fusion is presented in [16], along with a literature review of work in data fusion applications such as target detection and tracking, and robot navigation.

2.4.3 Ad-hoc fusion schemes

In the absence of previously trained probability distributions or fuzzy belief sets, other fusion approaches have been found to be useful. These approaches include linear combinations of evidence and democratic integration.

Linearly combining the sources of evidence has been used in many works, where each source is given a weighting. In order to robustly detect people in cluttered images, segmentation and edge-based information from chamfer matching scores are linearly combined in [74] using a fixed weighting of the features. A linear combination of shape and appearance is used for person tracking in [78], also using fixed weights. The feature weights are adaptively updated in [125] to improve object tracking using colour and edge histograms.

Democratic integration is proposed in [144] as a framework for combining evidence from multiple sources by adaptively weighting the sum of saliency maps from each source. In the experiments, a face is tracked using information from colour, motion, prediction, contrast and shape cues. Sudden changes in the testing videos cause some of the cues to fail, but as long as the changes disrupt only a small number of the cues simultaneously, the tracking can survive failure of each cue at different times. A number of methods for valuing a source's information quality are proposed, including uniform quality for all cues, quality based on the correlation between the cue's saliency map and the fused map, and quality based on the source's saliency at the correct face location, compared to the average saliency over the entire map. This quality measure is then used in updating the fusion weights.

2.4.4 Discussion

Both the Bayesian method and the TBM use pre-learned probability distributions or pre-computed fuzzy belief sets in order to provide a good judgement of the most likely outcome given the present data. There are cases where it is not possible to pre-learn this information, either because of a lack of resources, such as time or memory, or because the environment in which the system will operate is unknown. There are also cases in which it is possible to pre-learn this information but whereby the environment may change such that the learned knowledge no longer represents a useful likelihood model for the system. The core ideas in this thesis tackle these issues focussing on the idea of adaptation to different and unknown environments, specifically in the areas of parameter selection for object detection and in adaptive object tracking.

2.5 Background modelling

Many captured scenes, especially for visual surveillance, involve static camera settings where the camera is in a fixed position relative to the scene. In some moving camera sequences, such as some unmanned aerial vehicle (UAV) data, the scene may be considered planar and when the motion is compensated, the camera can be considered static. In these static camera settings, the object detection problem is made much simpler than in the general case, as most of the scene remains relatively fixed for long periods of time. A *background model* is a model of these static scene elements. In its simplest form, a background model is an image of the scene when no moving objects are present. At each time step, the new data can be compared to the background model. Anything that is considered significantly different from the background model is said to be detected as *foreground*. The rest of the scene is unimportant. This notion of change is a vague one and is difficult to accurately define in any broad sense. As Radke et al. [110] note, *"The notions of 'significantly different' and 'unimportant' vary by application, which sometimes makes it difficult to directly compare algorithms"*.

For example, if the application targets the detection of people, it is not desirable to detect trees moving in the breeze. However, another target application may be to visually determine the weather conditions, and it would therefore be desirable to detect this motion. Most background modelling approaches in the literature usually have a target application in mind (such as surveillance), and therefore define "significantly different" in that context.

Modelling the background in static camera scenarios remains a very active research topic. To review the work in this area, we begin by describing a simple approach and discussing its drawbacks, then build upon this discussion to consider more complex systems designed to deal with its limitations.

2.5.1 Simple approaches and their problems

The simplest approach to background modelling is to capture an image of the scene when no moving objects are present. To then detect moving objects, the background image is subtracted from the current image and the absolute difference between the two images is computed (hence the term *background subtraction* is often used interchangeably with background modelling and foreground detection). This difference image is then thresholded. The threshold can simply be fixed at a value relative to the camera pixel noise (e.g. a brightness difference value of 10). Pixels whose absolute difference is greater than this threshold are considered *foreground*. These foreground pixels are then clustered into *blobs*, using a connected components algorithm, for further analysis (such as tracking or classification).

Although this method may work in certain scenes, there are numerous drawbacks to this simplistic approach. Firstly, a fixed image is usually not a good model for scenes that change over time or which contain moving background objects, such as flags, trees, leaves, CRT monitors etc. Secondly, slow lighting changes (especially in outdoor scenes) and abrupt lighting changes (usually in indoor scenes) can cause the background image to be an insufficient model for the background of the scene. Thirdly, objects that should be considered part of the background (such as chairs) may be moved and again, would render this simplistic model insufficient.

Other challenges that pose difficulties for background modelling include: scenes of high traffic in which moving objects are always present, reflections from smooth surfaces (such as glass) and camouflage, where foreground objects are similar to the background.

The literature on how the above issues should be tackled for background modelling is extensive. Most methods initially model each pixel independently, so we review the approaches to pixel-based background modelling in the next subsection and proceed to describe more complex models later.

2.5.2 Pixel-based models

To model the background of a scene, the vast majority of approaches begin by modelling each pixel separately. Pixels are usually either monochrome (single-valued) or colour, with three values to represent the colour; usually red, green and blue (RGB) are used, but these can easily be transposed to other colour spaces such as YUV or HSV.

An incremental improvement on the simple value-based pixel models of the last subsection is to use a Gaussian distribution to model the pixel's colour. This model has two parameters: mean and (co-)variance. For monochrome data, the variance is a scalar. For colour images, the co-variance is a 3×3 matrix. For simplicity, this matrix is often assumed to be diagonal or even a scalar multiple of the identity matrix. These parameters can be learned during a training period when no moving objects are present and then they remain fixed while the system is operating. Alternatively, to account for changes in the background, the parameters can be updated continuously in an online manner. However, care must be taken not to allow moving objects to corrupt the background model. The Pfinder system described in [154] uses a Gaussian model per pixel with a full covariance matrix in the YUV colour-space.

In their seminal work, Stauffer and Grimson [137] demonstrate the limitations of using single Gaussians to model pixels. For many scenes, both indoor and outdoor, background pixel values have multi-modal distributions and cannot be accurately modelled using unimodal distributions. Examples of multi-modal distributions include water, computer screens and outdoor foliage. They propose to model each pixel using a mixture of Gaussian distributions and to update them in an online manner.

Many later works improved upon their approach, making the model updating faster and accounting for moving shadows [63] as well as catering for moved background objects [140].

Finding distributions that use mixtures of Gaussians limiting, a data-driven non-parametric distribution for background modelling is proposed in [30]. This is cited as a better performing algorithm than the mixture of Gaussians in a small benchmarking test [148]. An alternative approach to using distributions is given in the WallFlower system [143], where a Wiener filter is used as a linear predictor to determine a pixel's background value, based on a recent history of values. Exploiting the temporal periodicity of pixel values, a frequency-based representation is used to model each pixel in [155]. This is shown to perform well in natural scenes with water or trees, but the model has the side-effect of leaving a trail after foreground objects. In the W4 surveillance system [53], each monochrome pixel is modelled as a minimum and maximum brightness value, along with a value of maximum temporal change between frames. If the pixel value goes outside these limits, it is considered foreground.

A novel method of background modelling is proposed in [77] where temporal differencing is used to classify each pixel into either a *moving pixel* or a *static pixel.* Moving pixels are analysed using colour co-occurrence statistics in order to determine whether it belongs to a foreground object or is part of some moving background, such as an escalator, water fountain or moving curtains. Static pixels are classified as foreground or background using colour distribution statistics. The results show improved performance on complex videos when compared to two widely-cited algorithms. Additionally, learning strategies are proposed to cater for sudden and gradual changes in the scene and to adaptively select the learning rate.

Lighting changes frequently plague background models for outdoor scenes that are based on colour alone. The use of gradient based models is the most frequently used method to overcome this difficulty.

Li and Leung propose a background model that integrates both intensity and texture differences [76]. Two methods of integration are described: one method is adaptive, based on the weighting of texture evidence and the other method attempts to minimise an energy function that encourages the final result to be spatially smooth. Mathematical analysis and tests with real world data demonstrate the effectiveness of their approach with respect to noise and illumination changes.

Also motivated by the need for robust background modelling even in the presence of non-stationary scenes with sudden lighting changes or periodic background motion, Matsuyama et al. [85] propose to use pixel-block-comparison measures based on the normalised vector distance (NVD). Using such a measure is robust to illumination changes but can be unstable in homogenous image areas, so they propose a more robust alternative which they term *spatially modulated NVD.* To cater for periodic background motion, a temporal co-occurrence NVD matrix is used.

The BRAMBLE system [56] is quite different from most other background modelling approaches in that it also models the foreground. During a training phase, foreground and background colour and edge distributions are learnt. These distributions are subsequently used to determine the likelihood of foreground of each image block, and these likelihoods are then used in a particle filter to assess the likelihoods of each particle, which represent possible system states. The system state describes how many objects (people) are in the scene, their size, position and velocity.

In [126], non-parametric densities are used to model both the background *and* the foreground. The dependencies between pixels are exploited by modelling the density in colour-position space. Foreground is first identified using temporal persistence is used as a detection criteria. Subsequently foreground pixels are detected using the *graph-cuts* algorithm.

In order to decide when to update the background model for a pixel, the EM-SWITCH model [153] use the Expectation Maximisation (EM) algorithm, comparing foreground and background probabilities to make the update decision. They show how the commonly used updating rule, that only updates when the current pixel is within k standard deviations from the mean, results in the variance decreasing at every iteration.

Shadow Detection Shadows caused by moving objects change pixel colours and are often detected as foreground but it is generally desired that shadows not be included in the foreground detection process. This has led to much research in the detection of moving shadows. Shadow detection is often a post-processing step. After foreground detection, each foreground pixel is evaluated to determine if it could actually be part of a moving shadow. This is difficult to do reliably in monochrome images. In colour sequences, shadow pixels are distinguished by observing a decrease in the pixel's luminance (brightness) component and a negligible change in its chrominance component. The colour saturation has also been found to be decreased by a cast shadow. A good evaluation of various algorithms for shadow detection can be found in [108].

2.5.3 More than just pixels (Hierarchical models)

Many of the more difficult scenarios in background modelling cannot be handled at the pixel-level alone and benefit from using feedback from higher-level reasoning in the system [52] or using multi-modal data, such as stereo [39, 48] or infrared. In this subsection, we examine hierarchical background models that build upon the pixel level background modelling methods of the last subsection. In the next subsection, we will discuss the use of multi-modal information in background modelling.

The WallFlower system [143] performs processing at region level and at a frame level, as well as the basic pixel level. The region level processing helps fill in camouflaged pixels in the interior of an object, while the frame level analysis determines if a global change has occurred, such as a light being turned on, causing the algorithm to switch to another background model.

To cater for quick illumination changes, relocation of background objects and initialization with moving objects, Javed et al [57] also use a hierarchical approach. Background pixels are modelled using both a colour and a gradient model. The colour model is used to detect foreground regions, which are then validated using the gradient foreground at the boundary of the region. Quick illumination changes and *ghosts* (moved background objects) cause foreground regions to appear but they are do not have a strong gradient on the boundary, which allows them to be distinguished from true foreground regions. If the colour model results in more than 50% of the image being classified as foreground, the frame level processing will ignore the colour model and will use gradient only.

In [140], using a mixture of Gaussians model, texture is integrated with colour to detect false foreground regions caused by lighting changes, as in [57], but it is done in a local pixel neighbourhood, instead of at region level. Additionally, they detect static objects and *push* the whole object into the background, avoiding the foreground fragmentation problem which occurs when updating is done at a pixel level, since different parts of the static object will update faster than others.

In order to cater for difficult high-traffic situations, such as in a busy shopping district, where the background is never fully visible, the approach described in [35] is to process the video sequence offline and first identify periods of motion. These motion block areas are discarded and the remaining blocks are clustered by correlated them with all other blocks in the same position. The largest cluster is the dominant block in the video and becomes the background for that area. Their method performs favourably against the standard median filter. However, the output of their algorithm is an image for the background, which will not take multimodal pixel distributions into account, as mentioned previously.

In order to discount moving background, such as water, trees and flags, [148] use motion consistency as a measure of object detection, discarding objects whose motion does not have a consistent direction.

When foreground objects enter the scene then stop moving, they should not be expected to remain as foreground forever, but eventually become part of the background. For example, a car entering the scene and parking. This causes *ghosting* problems when the object again begins to move. A relatively recent approach to this problem is to use a *layered background*, where the different layers correspond to background at different times. The lowest layers correspond to the most stable background, and higher layers correspond to more recent objects that entered the scene and became static [151].

2.5.4 Multimodal background modelling

The problem of camouflage is one that cannot be easily solved. If the background is black in colour and a person wearing black clothing walks over it, the person will more than likely not be detected. When one feature, such as colour, cannot separate two classes, such as background and foreground, then introducing new features that can discriminate between them is a useful way to proceed. Thermal infrared is one such feature that is very helpful in discriminating between people and the background when colour fails.

Davis and Sharma describe their approach to foreground object detection using thermal and visual imagery in [27]. While lighting changes and shadows affect the visual domain, the halo-effect can cause unwanted detections in the thermal imagery that is captured with a BST (ferroelectric) thermal camera. Using a contour-based approach, they overcome these difficulties in both modalities, since lighting changes and the halo-effect usually cause only weak changes in gradient magnitude and are uncorrelated in the two modalities. Salient contour pixels are detected in both modalities and then fused using a binary OR operation, followed by contour thinning using gradient information from both images. A contour-completion algorithm is run on the fused output and then the contours are flood-filled to provide the final silhouette.

As well as infrared, information from a second visible spectrum camera has been found to aid in foreground detection. By building a depth model of the background using stereo vision techniques, many of the difficult problems in single camera analysis become tractable. For example, stereo analysis is robust to sudden illumination changes and shadows [45]. Depth alone can have its own problems however, as the range data is prone to high noise [47] causing people who are at a similar depth to the background to be ignored. Standard colour cues can be combined with stereo depth to overcome this limitation. Krumm et al. use a depth and colour based background model to robustly detect foreground pixels [70]. These pixels are then clustered into blob regions within discrete disparity bounds. The blobs are grouped into people-shaped regions by searching through the space of possible clusterings. A similar approach is described by Zhao and Thorpe [161] where adjacent foreground pixels are grouped if they have similar depth values and the region size does not exceed that of a person. Darrell et al. combine colour, stereo and face detection in their person tracking system [26].

Our early work in the domain of multimodal background modelling is detailed in two prior publications. The works in these publications are not directly related to this thesis, but we briefly review their contents here.

In [96], we modelled the backgrounds separately in the thermal infrared and visible domains. The thresholds for foreground detection in both modalities are adaptively computed based on maximising agreement measures between the foreground of both modalities. This early work led to our work in mutual information thresholding in chapter 3. Fusion of the detected foreground regions was achieved using a region-based method. First the foregrounds from both modalities are merged using a simple binary OR operation. Then any region that does not include foreground pixels from both modalities is discarded. Failure of the visible modality, such as when the light is turned off, was detected by measuring if a large part of the visible image was returned as foreground and switching to an *infrared-only* mode. In this mode, IR foreground was detected using hysteresis segmentation.

In [95], the non-parametric background model of Elgammal et al. [30] was adapted for multimodal background modelling by using 4 bands instead of 3, modelling each pixel's distribution in 4-dimensional RGB-IR space. Further, the model was extended to handle occluded background pixels by allowing model pixels to be marked as *unknown* when they are believed to be occluded. This belief comes from a person detection module which is used in infrared to pre-detect people in the scene and prevent them from being stored in the background model. A rule-based background-update scheme is used to encourage the updating of regions that are static, cold, small or have low gradient values, suggesting a lighting change. Shadow detection is performed using an object based approach, detecting only those shadow regions that overlap with a detected foreground object.

2.5.5 Discussion: open issues and future directions

In this section, a thorough review of the state-of-the-art research in background modelling was presented, covering the wide array of approaches to pixel modelling and how these models can be combined in hierarchical systems to cater for problems that cannot be accounted for at pixel level. Implementation details of many of the most common background models are given in McIvor's survey of background modelling techniques [87].

The problem of camouflage is not one that can be easily remedied. When the distributions of two classes, such as background and foreground, significantly overlap in the feature space, there is no way to accurately distinguish between them. The introduction of additional discriminative features, such as thermal infrared data, is a viable solution, especially since the objects of interest, usually people and vehicles, are sufficiently different from most backgrounds in the thermal modality.

Regardless of the background model used, the detection of foreground can be adversely affected by the incorrect detection threshold. In this context, the next section examines the extensive literature on adaptive threshold selection.

2.6 Thresholding

In the previous section, approaches to background modelling were examined, in order to disregard the commonplace appearance and motion in the scene and to detect salient objects. The background model allows us to compute the difference or distance between what is in the current scene and the background. To actually determine if something significant appears in the scene, this distance must be thresholded. This is sometimes referred to as *change detection* in the literature.

The selection of an appropriate threshold can dramatically effect the vision system's performance. A threshold set too high will result in many missed detections; set too low, there will be many false positives. A fixed threshold, even if carefully selected, may not perform well if there is a change in the properties of the scene, environment or objects of interest. A change in brightness or contrast is an example of this. By dynamically adapting the threshold to cater for different scenarios, these limitations can be addressed.

There is a wide range of research on the subject of dynamic (or adaptive) thresholding. The majority of approaches observe some signal property and determine the best threshold to suit this property. In an extensive survey of image thresholding techniques [123], where 40 different thresholding approaches are evaluated, six categories of thresholding algorithm are identified, each using a different measure to determine the optimal threshold. The measures used in each of the thresholding categories were: (i) histogram shape information, (ii) measurement space clustering, (iii) histogram entropy information, (iv) image attribute information, (v) spatial information and (vi) local characteristics. One major difference between change detection and the applications that were evaluated in this study (thresholding nondestructive testing (NDT) images and document binarisation) is that the majority of pixels will belong to the "no change" class. This can cause problems for thresholding algorithms that try to fit models to the data, as the 'change' class has a low number of samples, hence leading to a poor fit of the model.

The change detection problem was specifically addressed in [118], where a number of methods were tested on real data. It was possible for the ground truth to be automatically generated since the object of interest was a spherical ball that was easy to track. Of the eight methods tested, Kapur's method [65] showed the best performance. Kapur's method also performed very well in [6] and was ranked highly for NDT image thresholding in [123]. Kapur's method is an entropy-based algorithm that selects the threshold in order to maximise the sum of entropies of the two classes (change and no-change).

The Euler number of a binary image is the number of regions minus the number of holes. This feature was found to be useful in determining a good threshold using an image's spatial information [115]. The initial implementation was inefficient but a real-time implementation is described in [133].

Noting the difficulty in fitting a model to the *change* class, Rosin proposes to treat the histogram as uni-modal [116]. Since there are usually very few samples in the *change* class, this is a valid assumption. The algorithm is based on an intuitive geometric analysis of the difference image histogram. First, a line connecting the histogram peak and the last non-empty bin is drawn. The threshold is selected as the bin whose point on the histogram curve is furthest from this line.

Otsu proposed to select the optimum threshold such that it would minimise the weighted sum of within-class variances of the foreground and background pixels, which is equivalent to the maximisation of between-class scatter. This method performs best when the numbers of pixels in each class are roughly equal. Otsu's method [99] performed well in [123]. In change detection, as expected, Otsu was found to perform poorly [118].

Kittler and Illingworth treat threshold-selection as a minimum error Gaussian density-fitting on the histogram [69]. Their method was ranked first in [123] for both NDT images and document binarisation. It also performed well in [6], where is was concluded to conform very closely to the ideal equal-error case, where equal amounts of background and foreground pixels are mis-classified.

Ridler and Calvard [113] use an iterative clustering approach to threshold selection. The mean image intensity serves as an initial estimate. Pixels are classified as foreground and background using this threshold and the threshold is iteratively re-estimated as the average of the two class means. In the conclusion of thresholding tests using synthetic data [6], it is noted that the methods of Ridler and Calvard, as well as Otsu's method, fail when the number of background pixels is more than 10 times greater than the number of foreground pixels.

Tsai [145] models the difference image as a blurred version of the ideal thresholded binary image. The best threshold is selected so that the first three gray-level moments are equal to the first three moments of the thresholded image.

In recent work, an interesting approach to thresholding is described by Rahnamayan et al. [112] where instead of defining a new thresholding criteria, a combination of thresholding algorithms are used in order to improve performance. Using the most successful thresholding algorithms, including Kapur, Kittler and Otsu's methods, the threshold for each method is computed. A weight is assigned to each method, based on their performance in previous tests [123]. Finally, to *fuse* their results, the weighted median of all thresholds is selected. This is found to outperform any individual algorithm on average using a small collection of 15 test images.

2.6.1 Local thresholding

While most studies of thresholding focus on selecting a single threshold for the entire image, it may be beneficial to adapt the threshold spatially, as well as temporally. In fact, if the spatial and temporal indices of the data elements are ignored, the data becomes one dimensional, and the issue now is to choose an appropriate window size to use for performing the thresholding.

In [3], Adamek et al. use thresholding as a first step in extracting words from handwritten documents for use in a word-matching retrieval system. A local thresholding approach is used to account for variations in the pressure used to apply ink to the page. The method used is a variation of Niblack's algorithm [94], using the threshold selection method proposed by Sauvola et al. [119]

Thresholds selected locally were used to aid global threshold selection in [139]. The image is first split into K blocks; $K = 9$ was used in the paper. A *region of change scatter-algorithm* is then applied to each block to determine if it contains a regions of change. Two algorithms are used to select each block's threshold: one for regions of change blocks, based on histogram partitioning, and one for background blocks, based on a Gaussian noise assumption. Finally the global threshold is an average of all local block thresholds. The proposed method seems overly complex however, and includes a number of parameters, at least one of which is set empirically, while some others are selected adaptively.

2.6.2 Discussion

By thresholding a signal, some information in the signal is lost. It would be desirable to retain all the information input to a system. However, it is unfeasible to maintain the likelihood of every possible outcome and without thresholding, some sacrifices must be made. For example, the BRAMBLE system, mentioned earlier in the previous section on background modelling, does not explicitly detect foreground and hence, does not require thresholding. Instead the likelihoods of foreground for each pixel are used. The drawback is that, since no explicit foreground detection is performed, the colour distributions of all objects must be merged into one *foreground distribution* and this makes it difficult to distinguish different people. Their sizes and velocities are the only way in which they may be distinguished, which are not very discriminative features given typical image resolution, the similarity of people in size and the high likelihood that their velocities can suddenly become zero, if two people were to stop and talk to each other, for example.

Thresholding is useful for reducing the numerous possibilities a system must remain aware of and is a necessary step when a decision needs to be taken, such as when to initiate a tracker. The goal in thresholding therefore is to reduce the amount of useful information that is lost due to the thresholding process. The current methods for dynamic threshold selection cannot exploit the redundancy present in multiple data sources to aid their analysis. Analysis is performed on individual data sources without consideration for how they relate to the information in other sources of data. Having knowledge of how the sources relate can help determine whether or not an event or pixel is relevant. In chapter 3, the concept of *mutual information thresholding* is introduced, whereby data sources can assist each other in optimal threshold selection. This is shown in many cases to outperform existing dynamic thresholding algorithms.

2.7 Object tracking

Object tracking is important for many applications in computer vision, such as traffic monitoring, human-computer interfaces and remote surveillance. The term “tracking” can refer to either two or three dimensional object tracking.

In three-dimensional tracking, the purpose of tracking is to estimate the 3D position and pose parameters of the tracked object. Unless multiple cameras are used, there can often be ambiguity in the estimation process due to the limited nature of the 2D input images [89]. Also, for complex deformable objects, such as people, the number of parameters to estimate is quite large, further exacerbating the problem.

Two-dimensional tracking aims to identify pixels that are part of the object of interest, often fitting either a bounding box (quadrilateral) or an ellipse to the tracked object.

In order to perform accurate tracking, some features of the object are selected to be tracked. Ideally, these features should be stable (relatively constant for that object) and also discriminative (help in separating it from the background and other objects).

In general, tracking can be considered an ill-posed problem, as in certain situations it is not clear what exactly should be tracked. For example, while tracking a caterpillar’s cocoon, it changes into a butterfly; should the cocoon or the butterfly be tracked? If a tracked object splits in two, which half should be followed? While it is true that what should be tracked it not always clear in the general case, for particular applications the problem is more well-posed. Similarly, while the background modelling problem may be considered ill-posed in general, since it is not always clear which parts of a moving scene should be considered as background, in specific applications, such as site surveillance, the distinction between background and relevant objects is clearer.

2.7.1 Traditional approaches

The vast majority of object tracking approaches in the literature follow a similar structure. Firstly, a method to model the object is selected. Next, a similarity measure is chosen to compare the model to candidate objects. Finally, a method to search for the best match to this model in consecutive frames is needed. This general approach is used here to broadly categorise the tracking literature. In the next two subsections, we review the various approaches to *object modelling and matching* and *model locating* for object tracking. There follows a discussion on the main problems in object tracking and an examination of the extensive range of approaches in the literature to tackle these issues.

Object modelling and matching In order to track an object, some kind of representation (or model) of the object is required. This model should ideally be stable enough to capture the different appearances of the object, and it should be discriminative, allowing the object to be distinguished from other features in the visual scene. The chosen similarity measure, used to compare the model to object hypotheses, can also affect how discriminative the model is. Here we briefly discuss the main approaches to object modelling and matching.

Feature histograms have been shown to be robust and efficient for object modelling for use in surveillance tracking, as they capture stable object properties that are resilient to changes in object pose due to local object motion (e.g. walking) and small changes in perspective. In their seminal work, Comaniciu and Ramesh [22] [23] derived a mean-shift formulation for histogram tracking allowing real-time tracking that requires only a few iterations per frame to converge on the correct target. Adaptation to scale changes is performed by examining windows that are 10% larger and smaller than the current size. Collins [19] improved upon this scale selection heuristic, deriving a two-stage mean-shift procedure that interleaves spatial and scale mode-seeking using differential scale-space filters. In [163], scale adaptation is formulated as an EM-based approach. A method to perform very fast exhaustive histogram matching to locate the tracked object position is proposed in [105], where integral histograms are computed using a dynamic programming approach, allowing it to search the entire image efficiently instead of just a small search window. The drawback to this method is that it may require a large memory overhead. Birchfield and Rangarajan [11] generalise the histogram formulation by introducing spatial histograms, or *spatiograms*, that are histograms with higher-order moments. Like histograms, spatiograms allow comparisons between image regions without explicitly computing any explicit geometric transformation between them. However, unlike histograms, spatiograms retain some information about the geometry of object feature distributions, allowing them to remain more tightly locked onto their targets and less likely to be distracted.

The Bhattacharyya coefficient is the most frequently used similarity measure to compare histograms in tracking. Huet and Hancock found that the Bhattacharyya distance outperformed the standard L_1 and L_2 distance measures between histograms for the task of aerial image retrieval [51]. In [80], Ling and Okada introduce the diffusion distance measure for histograms, comparing it to 8 other standard measures. High performance and efficiency is demonstrated in shape and feature matching applications. Meaures such as Bhattacharyya and the L_1 and L_2 distance measures perform only a bin-wise comparison, whereas the diffusion distance and *earth-mover's distance* take adjacent bin similarity into account. This means that they are more robust to small histogram changes, but are computationally more expensive.

Instead of modelling the object's appearance, the object boundary is another useful feature, especially for textureless objects or for medical applications. Active contours, or *snakes*, have frequently been used to track the boundary of an object [2, 24, 101]. The similarity measure used to match the active contour in consecutive frames uses a formulation that applies a tradeoff between having a smooth boundary and the importance of the boundary corresponding to real image edges. The CONDENSATION algorithm (CONditional DENSity propogATION) [55] proposed by Isard and Blake is another contour-based tracking algorithm. Instead of using the energy-minimisation strategy of standard snakes, however, particle filtering is used to efficiently search the high-dimensional parameter space of the object's contour and to perform tracking using a probabilistic model of its shape.

Appearance models [59, 98, 122, 162] try to model the visual appearance of the tracked object are also popular for object modelling for tracking. In [162], Zhou et al. introduce an adaptive appearance model for robust tracking. Using image brightness values in their appearance model, results are shown on tracking a car, a frontal face and an aerial view of a tank. The objects they tracked do not alter significantly in appearance, although the pose does change.

Image templates have also been used as object models [86], where an image of the object is used as a model. The *sum of squared differences* (SSD) is a very common similarity measure used for comparing templates to potential object locations.

In fixed camera scenarios, a background model can be estimated and subtracted so that moving objects are modelled as foreground blobs. In this case, the model is usually just the bounding box surrounding the object in the previous frame. Matching a blob in consecutive frames can be done by simply finding the blob that overlaps it in the next frame. For more complex scenarios, where the blobs split and merge, such as during an occlusion, rule-based procedures are needed to determine the identity of each blob.

Model Locating In subsequent frames, tracking proceeds by matching the model to the most likely position in the image. In order to efficiently search for this optimal position, a number of different strategies have been proposed.

A brute-force exhaustive search can be used in a search range around a predicted object position. This search range can be fixed or dynamic based on the object's velocity or on how well the model matches.

Since most similarity measure surfaces are smooth, gradient ascent type methods can be used [86] to efficiently ascend the surface and find the local maximum. If the method is initialised close to where the object is expected to be, the local maximum will correspond to where the object is located.

Mean-shift [19, 23, 157] is another widely used model locating strategy in object tracking. Similar in some ways to gradient ascent, the model is initialised close to where the object is expected to be, and the pixels in that area vote to move the model towards a solution with a higher similarity score.

In high-dimensional tracking problems, or where there are many potential model distractors, particle filters [162] are a useful tool for tracking. Each *particle* can be thought of as representing a possible object position or state. Each time step, new particles are generated close to particles with high likelihood scores. In this way, multiple hypotheses can be maintained and the search space can be more efficiently covered. The CONDENSATION algorithm is another example of tracking using a particle filter [55].

Open problems in tracking There are three main reasons for tracking failure. Firstly, *partial occlusion* of the object may cause the tracking to fail. In cases of complete occlusion, higher-level reasoning is usually needed to hypothesise the unseen object's position. Secondly, *model failure* can occur due to some change in the object or in the environment, making the object model used for tracking unsuitable for accurate object localisation. An example of this is when a colour histogram model is used after a change in lighting. Thirdly, *feature failure* is when the features used for tracking are insufficient to distinguish the tracked object from the background (or from another passing object). For example, using colour features to track a green object in a forest environment. These three causes of tracking failure are now discussed in detail.

During the tracking of an object, it may become partially occluded, either by another object, or by self-occlusion (common in deformable objects such as people). Therefore, a robust tracking system should account for this, either by explicitly detecting occlusion and adapting its search strategy, or by using an object model that is robust to partial occlusion. A review of occlusion handling is conducted in the next subsection.

As discussed in the beginning of this section, it is necessary to use stable (relatively constant) features for tracking. When features violate this assumption, it can cause *model failure*, as defined above. Usually, the stability of features cannot always be guaranteed, therefore many papers make the assumption that the change in features is gradual. That is, the change is small between consecutive frames, therefore adaptation to these slow changes is possible by updating the object model. In a later subsection, the various approaches to model updating are reviewed.

Feature failure results from an inability to distinguish the object from its surroundings. The use of multiple features somewhat reduces the likelihood of ambiguity between the object and potential distractors. In the subsection on multi-modal tracking, a review of methods for combining multiple features in the context of tracking is conducted. Another method, related to multi-modal tracking, is to selectively adapt (or choose) the features that will be used for tracking, in order to best discriminate between the tracked object and other distractions. These approaches have much in common with *feature selection* approaches from the data classification literature. A review of these methods is conducted in the subsection titled *feature adaptation*.

The following subsection examines the more advanced tracking approaches that have been proposed to handle the above mentioned scenarios. These approaches include occlusion handling, object model updating, multi-feature tracking, feature adaptation and fusing multiple trackers.

2.7.2 Robust tracking

Occlusion handling Occlusion can be handled indirectly, by noting when objects become occluded and then assigning identities after the occlusion, or it can be handled directly, by attempting to track the object during the partial occlusion.

In [122], all objects in the camera's field of view are tracked; appearance models and linear velocity prediction are used to cater for situations where objects occlude one another. In [162], occlusion is handled using robust statistics and occlusion is declared when over 15% of pixels are determined to be outliers.

Model Updating No feature is so stable that it can provide perfect tracking in all circumstances. For example, while colour histograms are usually quite stable features for object tracking and insensitive to pose changes, this assumption of stability is violated during lighting changes, where the pixel colour features may change abruptly. Lighting changes are caused by a number of factors: changes in the ambient lighting (the sun setting, rising or going behind a cloud, a light switch turned on/off), camera auto-iris adaptation (the video capture device allows more/less light onto the sensor), cast shadows (a shadow from another object). Similarly, edge-based features are usually more robust to lighting changes, but can be affected by changes in object pose. Because a given feature used for tracking can change over time, it is important to update the model of the tracked object to account for this.

No tracking system can be expected to track a target whose features all change completely in one frame to an unseen configuration. Some assumptions need to be made in order for model updating to be practical in successful tracking. Examples of assumptions for model updating could be that any change should happen gradually over a number of frames or that when sudden changes occur, only a subset of features will be affected. These assumptions allow model updating to be a viable solution to tracking complex objects.

The extreme alternative to using a fixed object model, is to update the model for each frame to the best match found in that frame. A solution that lies in between these two extreme cases is to use a gradual updating scheme with an update parameter α. The updated model at time $t+1$ is computed from the current model and the best match in the current frame, as follows:

$$M_{t+1} = \alpha M_t + (1-\alpha)B_t. \tag{2.7}$$

It can be seen that the two extreme cases, of 'no update' and 'instant update', are specific cases of using this model, with $\alpha = 1$ and $\alpha = 0$, respectively. The extreme update strategies, along with a more reasonable $\alpha = 0.95$, were used in [162] where three separate appearance models were updated using these three strategies and combined using an adaptive weighting scheme.

The main problem associated with model updating is known as the *drift problem.*

Although the initial model for the object may be a very good representation of it for a number of frames, by updating the model using imperfect knowledge of its true location, the model is gradually corrupted, eventually deviating completely from the true object's appearance. One proposed soution to overcoming drift is to combine the initial object model, along with the updated model in the matching stage [86]. This prevents the updated model from drifting too far from the initial model [20].

Multimodal Tracking It is generally accepted that "no single visual cue will be robust and general enough to deal successfully with the wide variety of conditions occurring in real-world scenarios" [135]. Therefore, to create robust systems, multiple features (or cues) need to be used in such a way that they can, together, compensate for their individual weaknesses. The use of feature combination for tracking is an active research area and many approaches have been proposed to combine the information from multiple sources in order to provide more accurate and robust detection and tracking.

Probabilistic methods are commonly used to fuse information sources for tracking. In [82], Bayesian probability theory is used to fuse the tracking information available from a suite of cues to track a person in 3D space. A Bayesian tracking framework using particle filters is described in [103] for fusing colour cues with stereo or motion information. A Bayesian multi-object tracker is described in [130] that fuses binary information from foreground detection with colour tracking cues. Linear combinations of sources have also been widely used to fuse information from multiple sources. Lim and Kriegman [78] use a linear combination of shape and appearance to track people in an indoor environment. In [74], information from image segmentation is fused with chamfer matching scores to robustly detect people in cluttered images. Both [74] and [78] use fixed weighting for the data sources. In [125], the weightings for each tracking cue (colour and edge histograms) are adaptively updated using the Bhattacharyya coefficients in order to robustly track moving vehicles. Fumera and Roli [38] consider linear combinations of classifiers and conduct a theoretical analysis, as well as performing experiments on real data sets. Their conclusions were that weighted average combinations usually only provide a marginal improvement over simple averaging, even with optimal weights. In [141], Torresan et al. describe a surveillance system that fuses standard visible spectrum and thermal infrared video to detect and track pedestrians. Foreground regions in consecutive frames are linked using ad-hoc rules to account for splitting and merging. In [27], the benefits of fusing colour and thermal infrared information are demonstrated using a contour based approach. They compute a *contour* *saliency map* in each modality and from these maps, binary contour fragments are obtained and then fused. Silhouettes of the detected people are obtained by completing and closing the fused contour segments. In [135], the *democratic integration* scheme is used to fuse tracking cues from intensity, motion, skin-colour, shape and contrast for robust face tracking in changing environments.

As mentioned previously, feature histograms are commonly used for tracking stable features of objects. In the context of combining object features however, the main drawback of using histograms or spatiograms is that their memory requirements (and hence their computational load) increase exponentially as more features are added and they do not scale well to higher dimensions [8]. For example, an RGB colour histogram with 32 bins per channel requires a total of $32^3 = 32768$ bins. If an extra channel, such as thermal infrared, is added, this increases to $32^4 = 1048576$, which increases the memory requirements and decreases the tracking speed due to increased computation. There is also the issue of the *curse of dimensionality* [9] which states that it is more difficult to accurately estimate feature distributions for higher dimensional spaces, since exponentially more samples are required. It has also been shown in [157] that the Bhattacharyya coefficient, a similarity measure for histograms often used in tracking, is not very discriminative in higher dimensions.

Feature adaptation While the use of multiple features can help in tracking, the case may arise where some of the features being used are redundant or even harmful to tracking. Redundant features are those that provide no separability between the object and non-object classes. For example, thermal infrared might be redundant if both the object and background were at the same temperature. Using thermal features here would produce a uniform similarity surface, providing no tracking information. Harmful features, on the other hand, provide a similarity surface with multiple peaks. One peak corresponds to the tracked object, which the other peaks are background *distractors*. Avoiding redundant features makes tracking more efficient and avoiding harmful features makes it more robust.

Loy et al. describe a multi-cue tracking system that dynamically allocates computing resources by measuring cue performance [82]. In their chosen application, a Bayesian particle filter is used to fuse the cues and track a person in 3D space. The fused probability density function (PDF) computed from all cues to assumed to give the best estimate of the true PDF and the quality of each cue is measured on how well their individual PDFs match the fused PDF, in terms of the Kullback-Leibler distance.

In [125], vehicles are tracking using an adaptive fusion of colour and edge features.

Using a histogram to describe the object in each feature-space, the position of the vehicle in the current frame is computed as a weighted average of the positions returned by each feature. The weighting are derived from the Bhattacharyya coefficient returned after matching each histogram to candidate positions.

Stern and Efros used five colour space models for face tracking and switched between them adaptively [138]. The switching is based on a colour space quality measure that indicates how well each colour space separates the face from its immediate surroundings.

Object tracking is formulated as a classification problem in Avidan's work on *Ensemble tracking* [8]. The bounding box of the tracked object contains the pixel samples of the *object class* and a larger outer rectangle surrounding the object contains the background samples. Using the AdaBoost framework, an ensemble of linear classifiers are trained on the pixel samples in a least-squares manner in order to efficiently separate the object and background classes. Training takes place each frame, so the features used for tracking adapt to changes in the object and in the background clutter. Tracking is performed by creating a weighting image using the classifier *margin* and using the mean-shift algorithm on this weighting image. Avidan's framework is very efficient and can easily incorporate multiple pixel-based features, such as local edge orientation histograms.

Similar to the work of Avidan is the work of Collins et al. in the online selection of discriminative tracking features [20]. Given any single feature, a methodology is devised, similar to that of Stern and Efros [138], that allows the feature's tracking quality to be measured in accordance with how well it separates the object's true location from other potential background distractors. Collins et al. use a set of *seed features* that comprise of features obtained using linear combinations of R,G,B pixel values. These features are ranked according to their tracking quality and then the top N feature are used for tracking. Tracking is done by forming a new set of candidate features tailored to the local object/background discrimination task using the log likelihood ratios of class conditional sample densities from object and background. The mean-shift algorithm is then run on the log likelihood image to locate the object.

While the method of Avidan is quite fast, since it selects the weights for each feature in one pass, the method of Collins et al. requires that each feature must be assessed separately in order to rank it. However, the new set of candidate features computed by Collins et al. are more flexible than the linear combinations used by Avidan. Secondly, Collins et al. specifically target distractors, the main cause of tracking failure, whereas Avidan selects features that best separate the two classes on average.

The general approache of Avidan and of Collins et al. to adaptively selecting the feature used for tracking is a relatively new approach in this field. However, the selection of features for classification has been studied in detail. A good introduction to variable and feature selection can be found in the thorough review of Guyon and Elisseeff [42]. Mutual information is often used as a criteria for feature selection [102]. Recent work in that area tackles the problem of rapid feature selection from huge collections [73].

Tracker fusion As well as combining features to robustify tracking, it has been proposed to combine trackers to assist in this task as well. A probabilistic framework for combining tracking algorithms is described by Leichter [75]. They make the assumption that the algorithms are conditionally independent, and that each one outputs a PDF of the target's likely state or position. In this case, they show that simply multiplying the PDFs produces an analytically justified PDF of the combined tracker. Robust results are shown using their method for tracking a person, a ball, human eyes and human heads.

Toyama and Hager [142] examine the problem of how to fuse multiple different simple trackers to produce a more robust fused tracker. They first outline the different causes of failure of simple trackers, including poor localisation, fast motion, distraction, occlusion, brightness changes and the aperture effect. Using a rule-based fusion, trackers based on intensity, edge, hue and motion are combined. They conclude that by using a variety of simple trackers, and designing fusion algorithms that take into account the potential causes of failure of each tracker, significant gains in tracking robustness can be made with minor computational cost.

2.7.3 Future directions in automated object tracking

Both the adaptive tracking methods of Avidan [8] and Collins et al. [20] exclude spatial information, and treat objects simply as *bags of pixels*. Therefore when the background resembles the object in some fashion, such as when it has similar colours, their methods can easily concentrate only on one part of the tracked object (the most distinctive part) therefore mistaking the object's scale. In this thesis, spatial information is retained in the tracking framework we outline in chapter 5 using banks of *spatiograms* for multi-feature tracking. By focussing on *distractors*, Collins et al. target the most significant threats to robustly tracking the object. We adopt a similar approach in chapter 6 where we detail how the tracking features can be adaptively weighted within the spatiogram bank framework of chapter 5.

Chapter 3

Mutual Information Thresholding

3.1 Introduction

The unifying thread that runs throughout this thesis is the combined use of multiple sources of information. In this chapter, the use of two data sources in adaptive parameter selection is investigated; specifically in the application of dynamic thresholding.

Thresholding means throwing away information and discarding the rich continuous-valued signal in favour of a discrete (usually binary) representation, but there are many cases when thresholding is a necessary operation. Firstly, in cases when a decision must be made, to determine whether an alarm should be sounded or whether a tracker should be initialised, for example. Secondly, due to memory constraints or real-time considerations, thresholding is required to reduce the size of the solution search space. As Sezgin and Sankur remark in their conclusion of their extensive evaluation of thresholding algorithms [123]

> One should keep an eye on the fact that thresholding should be opted for [in] two-class segmentation problems due to their simplicity whenever they achieve performance similar to more sophisticated methods, like Bayesian schemes and random Markov models.

The selection of an appropriate threshold is an important aspect in many computer vision systems. In the many processing steps (or subsystems) in a vision system, it is often too costly to maintain the likelihoods for every possible correct output of the subsystem. Thresholding must be performed to prune the search tree and reduce the total number of possibilities. Thresholding is often an end in itself, in applications such as event and object detection where it is important to immediately raise an alarm or initiate tracking. The precise value of this threshold can strongly effect the system's performance.

A threshold set too high will result in many missed detections; set too low, there will be many false positives. A fixed threshold may not perform well if there is a change in the properties of the scene, environment or objects of interest. For example, the same threshold is unlikely to be optimised for both daytime and night time scenes. By dynamically adapting the threshold to cater for different scenarios, these limitations can be addressed.

In this chapter, the work primarily targets *change detection* and will use related terminology in describing the proposed approach. However, the proposed method is a generic method and is not limited to this single application, as the numerous examples in the next chapter shall demonstrate. A brief review of related research is now conducted, followed by an overview of this chapter's contribution.

3.1.1 Related research

Dynamic thresholding In the literature, dynamic (or adaptive) thresholding research focusses on three main applications areas: (i) non-destructive testing (NDT), (ii) document binariation and (iii) change detection in satellite/aerial imagery and surveillance images. Our main focus in this chapter is on the last category: change detection. In change detection we must threshold a distance (or difference) image, where values close to zero indicate that no change has occurred, with higher values indicating a higher likelihood of change.

One major difference between change detection and the first two applications (NDT and document binariation) is that the majority of pixels will belong to the *no change* class. This can cause problems for thresholding algorithms that try to fit models to the data, as the *change* class may have a low number of samples, leading to a poor fit of the model.

In the previous chapter, the review of thresholding research covered the wide range of attributes that were used in the various algorithms in order to determine the optimal value; Histogram entropy, as used by Kapur [65], histogram shape used in [116], image moments in [145], spatial information using the Euler number in [115] etc. Regardless of the approach, the structure common to all methods is that some optimality measure is defined based on some signal feature and then the threshold that maximises this measure is selected. In this work, our approach is different in that we do not use a measure based on the properties of a single signal. Instead we observe how the choice of thresholds for two signals will affect their relationship with each other. Specifically, we try to maximise agreement between the resulting binary signals and use mutual information as a robust measure of agreement.

Maximising mutual information Mutual information has been used in computer vision and machine learning for various applications, including data alignment [150], particularly in medical imaging [104]. In medical images, such as MRI scans of brain tissue, there are usually a small number of dominant pixel classes in the data (corresponding to different tissues) and therefore there is strong statistical correlation between the brightness values in multiple modalities. Mutual information can be used to measure the strength of this correlation and therefore maximising it leads to accurate alignment. Feature selection for classifier training [102] is also an application where mutual information has proven useful. In feature selection however, the features that should be selected are those with the *minimum* mutual information between them, as complementary features are needed for good classification.

The work of Kruppa and Schiele [71, 72] in maximising mutual information is most similar to the work described in this thesis. Their work primarily concerns the reliable detection of elliptical areas, representing skin or face regions. These regions are detected iteratively using a greedy algorithm by finding configurations that maximise the mutual information between detection modules. The work in this thesis on mutual information thresholding seeks a single parameter-set that will maximise the mutual information, instead of finding multiple such sets, and thereby avoids the use of an ad-hoc stopping rule. Additionally, this work specifically targets the selection of thresholds, which makes the search through parameter-space highly efficient.

3.1.2 Chapter overview

The following sections of this chapter describe this work's contribution to dynamic thresholding, termed *mutual information thresholding.* The three assumptions that underlie the algorithm are examined and its tolerance to deviations from these assumptions is investigated. The approach is evaluated using synthetic data and shown to outperform the leading dynamic thresholding algorithms that take into account only single signal information. Next, following an examination of the assumptions of the method, an extended version of the algorithm is presented to cater more robustly for correlated noise in the input sources. An online version of the *mutual information thresholding* algorithm is then described and shown to be suitable for wireless sensor networks. The chapter concludes with a discussion of the approach and some ideas for future research in this area. Numerous applications of the method are demonstrated on real data in the next chapter.

3.2 Mutual Information Thresholding

This section contains the algorithmic and implementation details of the proposed mutual information thresholding algorithm. The algorithm is described, along with two efficient implementations that avoid a brute force search for the best thresholds, using the integral image technique [149] and gradient ascent respectively. This work was first described in [97].

In the approach adopted here, two thresholds, not one, are selected for two separate data sources such that the mutual information between the two binary thresholded signals is maximised. This encourages high agreement between detectors, as well as high information content.

As an illustrative example, figure 3.1(a) and (e) shows two synthetic difference images and alongside them, a series of thresholding results using three methods. The Kapur method [65], shown in the second column, is a histogram-based method and chooses the dynamic threshold based on histogram properties of a single image. Similarly, the Euler method [115], shown in the third column, chooses the threshold based on spatial layout properties of a single image. By exploiting the relationship between the two difference images, the proposed mutual information thresholding method, shown in the rightmost column, removes almost all of the false positives.

The assumptions made when performing mutual information thresholding are, firstly, that the noise in the sources is uncorrelated, secondly, that both sources are aligned (spatially and temporally) and thirdly, that the sources have some common information relating to some event (or object) detected in both sources. The proposed algorithm is now described, but in the following section, these assumptions will be further investigated, specifically how performance is affected when these assumptions are not met.

3.2.1 Algorithmic details

Formally, we describe the algorithm as follows. We define a *detection score* as a confidence measure indicating the presence of an event. Similar to the distance measure defined by Smits and Annoni [131], the detection score is such that values close to zero are evidence for *no event* and higher values indicate a *possible event*. For example, a detection score could be (i) a difference image (detecting foreground), (ii) a change

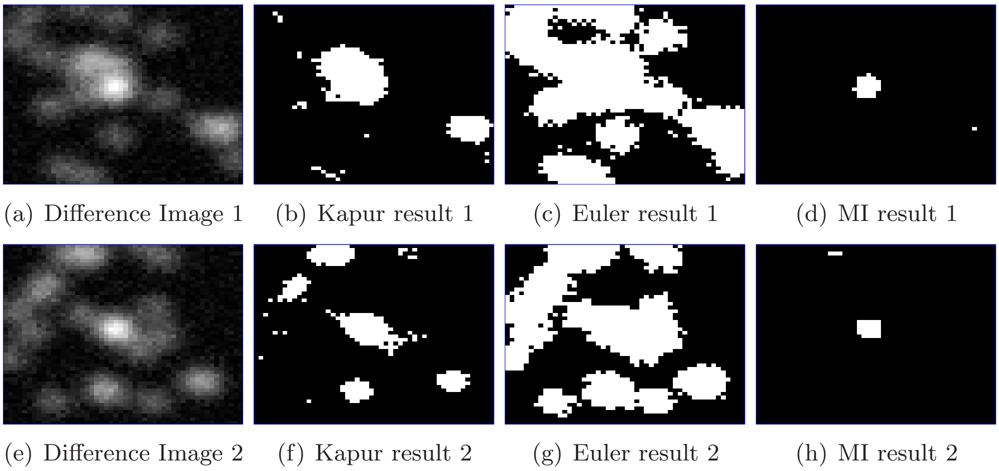


Figure 3.1: Illustration of thresholding approach. Shown in (a) and (e) are two synthetic difference images of the same scene. The *true change* has occurred in the centre of the image, but there are many potential false positives. Standard thresholding methods apply some model to the data histogram (e.g. the Kapur method shown in (b) and (f)) or the spatial layout (e.g. the Euler method shown in (c) and (g)) of single images. By exploiting the information from both images, the proposed method (d)/(h) selects superior thresholds.

detection mask or any signal that is expected to have high values in the presence of events/objects of interest or (iii) a likelihood image with high values indicating high likelihood of the presence of the sought event/object.

Given two sets of detection scores, X and Y, with $X = \{x_1, x_2, ..., x_N\}$ and $Y = \{y_1, y_2, ..., y_N\}$, that are aligned (spatially and temporally), we can choose thresholds, T_X and T_Y, to decide whether the event was present at a particular point, according to each set. By thresholding each set, we obtain the *binary* event detection sets, $X^{'}$ and $Y^{'}$, with $X^{'} = \{x_1^{'}, x_2^{'}, ..., x_N^{'}\}$ and $Y^{'} = \{y_1^{'}, y_2^{'}, ..., y_N^{'}\}$, given by

$$x_i^{'} = \begin{cases} 1 & \text{if } x_i \geq T_X \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

$$y_i^{'} = \begin{cases} 1 & \text{if } y_i \geq T_Y \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

These thresholds, T_X and T_Y, are chosen so as to maximise the mutual information between the distributions of $X^{'}$ and $Y^{'}$, expressed as

$$I(X;Y) = \sum_{u\in\{0,1\}} \sum_{v\in\{0,1\}} p_{XY}(u,v) log \frac{p_{XY}(u,v)}{p_X(u)p_Y(v)} \tag{3.3}$$

where $p_{XY}(u,v)$ is the probability that $x_i^{'} = u$ and $y_i^{'} = v$, $p_X(u)$ is probability that $x_i^{'} = u$ and $p_Y(v)$ is the probability that $y_i^{'} = v$. These probabilities are easily estimated by counting occurrences and dividing by N. We compute $p_{XY}(u,v) = C_{u,v}^{XY}/N$, where $C_{u,v}^{XY} = \#\{i; x_i^{'} = u, y_i^{'} = v\}$. Similarly, $p_X(u) = C_u^X/N$, where $C_u^X = \#\{i; x_i^{'} = u\}$ and $p_Y(v) = C_v^Y/N$, where $C_v^Y = \#\{i; y_i^{'} = v\}$.

Choosing the thresholds in this way leads to two desirable benefits. Firstly, it encourages agreement between the two detection sets, so that they often agree on whether the event has been detected or not. Secondly, it leads to high information content (or entropy). Without this consideration, agreement could be maximised by setting both thresholds very high (or very low) but the detectors would always return the same answer, regardless of the data they are analysing. The entropy of a single-valued image is zero, so the use of mutual information as a measure of agreement avoids this extreme case.

Given the detection scores, X and Y, and a pair of thresholds, T_X and T_Y, a mutual information score can be computed. By testing every possible pair of thresholds from a discrete set, each resulting in a different mutual information score, a *mutual information surface* is created, with the height at point (T_X, T_Y) equal to the mutual information score. These surfaces were found to be smooth and often convex and unimodal when the assumptions made earlier about noise independence and common information were valid.

The appropriate thresholds can be selected using an exhaustive search of all threshold pairs, but this is inefficient. Two much faster algorithms have been developed: one based on efficiently performing a full search using the integral image technique [149] and the other performing gradient ascent on the mutual information surface to locate the peak value.

3.2.2 Fast exhaustive search

Every pair of thresholds used for thresholding two signals will provide a corresponding mutual information (MI) value. By exhaustively computing the MI value for every pair of thresholds, a *MI surface* is obtained. In this section, how the integral image technique can be used to compute the entire MI surface using all pairs of thresholds (chosen from two discrete sets) is shown. The second method is to use the simplex algorithm and perform gradient ascent to find the maximum MI value, under the assumption of surface convexity.

A brute-force approach to computing the MI surface involves iterating over all pairs of thresholds (chosen from two discrete sets), using them to threshold both signals, then computing the MI between the thresholded signals. If T_c thresholds are tried for each signal, this results in T_c^2 pairs and a computation in the order of $O(T_c^2 N)$, where N is signal size (e.g. the number of pixels in an image). The proposed integral-image-based algorithm achieves the same results in time $O(T_c^2 + N)$.

Firstly, we denote $A = \{a_1, a_2, ..., a_P\}$ as the set of thresholds we wish to evaluate for the first signal and $B = \{b_1, b_2, ..., b_Q\}$ as the set of thresholds we wish to evaluate for the second signal. These sets are ordered in ascending order, with a_1 and b_1 set to a value lower or equal to the smallest value in X and Y respectively. The value of a_P and b_Q are set larger than any value in X and Y respectively. Next, we note that equation (3.3) requires the four values for $p_{XY}(u, v)$, with $u, v \in \{0, 1\}$. $p_X(u)$ and $p_Y(v)$ can be obtained from these values (e.g. $p_X(1) = p_{XY}(1, 0) + p_{XY}(1, 1)$). Each of these four values are computed by counting the number of occurrences where $x_i^{'} = u$ and $y_i^{'} = v$, then dividing by the total number of values, N. Therefore, we wish to compute these four counts for each pair of thresholds we wish to evaluate. We denote the counts as $C_{u,v}(i, j)$, which equals the number of occurrences where $x_k^{'} = u$ and $y_k^{'} = v$, when the thresholds are set at $T_X = a_i$ and $T_Y = b_j$. Initially the counts are all set to zero. For each data point we have the values x_k and y_k. From these values, we can deduce that $C_{0,0}(i, j)$ will be increased by one when both $a_i > x_k$ and $b_j > y_k$. Similarly, $C_{0,1}(i, j)$ will be increased by one when both $a_i > x_k$ and $b_j \leq y_k$. Count maps $C_{1,0}$ and $C_{1,1}$ have similar rules. For each data point, we could increase the counters in each map by iterating over all thresholds that should be increased. A faster method is to store markers at the positions in the map where the count increases or decreases, and integrate afterwards. This is a similar, complementary technique to the standard integral image method used in [149] to quickly find the sum of all pixels in a rectangular area of an image. The pseudo-code describing how to update the count map markers for a data-point is shown in figure 3.2. Finally, we integrate all the counts horizontally, as follows:

$$C_{u,v}(i, j) \leftarrow C_{u,v}(i, j) + C_{u,v}(i - 1, j) \tag{3.4}$$

```
Init: $C_{u,v}(i,j) = 0$ for all $u, v, i, j$

For all data points $(x_k,\, y_k)$
    Find largest threshold $a_i$ such that $a_i \leq x_k$
    Find largest threshold $b_j$ such that $b_j \leq y_k$
    $C_{1,1}(1,1) + +$
    $C_{1,1}(i+1,1) - -$
    $C_{1,1}(1,j+1) - -$
    $C_{1,1}(i+1,j+1) + +$

    $C_{1,0}(1,j+1) + +$
    $C_{1,0}(i+1,j+1) - -$

    $C_{0,1}(i+1,1) + +$
    $C_{0,1}(i+1,j+1) - -$

    $C_{0,0}(i+1,j+1) + +$
```

Figure 3.2: Pseudocode for algorithm in subsection 3.2.2 to effciently compute the entire MI surface, relating mutual information to selected thresholds.

and then afterwards, we integrate vertically,

$$C_{u,v}(i,j) \leftarrow C_{u,v}(i,j) + C_{u,v}(i,j-1) \tag{3.5}$$

This array now stores, at location $C_{u,v}(i,j)$, the number of occurrences where $x_k^{'} = u$ and $y_k^{'} = v$, when the thresholds are set at $T_X = a_i$ and $T_Y = b_j$. Using the obtained values, the entire MI surface can then be computed using equation (3.3).

3.2.3 Gradient-ascent search

Any gradient ascent method will be very computationally efficient, compared to a full search, as its complexity is of order $O(IN)$, where I is the average number of iterations and N is the number of signal samples, as before. Using a gradient ascent approach (such as the Simplex algorithm) also has the advantage that the thresholds do not need to be quantised into discrete values. Any full-search approach will require a finite set of pairs of thresholds, therefore demanding a quantisation of the values. This means that the Simplex search finds a more precise optimum solution. Simplex (or another gradient ascent method) can also be used efficiently for higher dimensional thresholding. For example, if we wished to choose P thresholds that would maximise the mutual information between P thresholded signals, a full-search would usually be unfeasible for large values of P.

Simplex initialisation and Scale In order to use Simplex, the initial position and simplex size needs to be specified. The choice of these parameters may depend on the application. One proposed approach to initialising Simplex for video processing is to do the following. In the first two video frames, a full search is performed, using as fine a quantisation as is possible within the time constraints. The thresholds found using the full search can be used to initialise the Simplex search in subsequent frames (i.e. the thresholds found in the previous frame are used as the starting position for the current frame). The simplex size can be determined by setting it to be a fraction (e.g. 10%) of the change in thresholds between the first two frames. This size can be left fixed or adapted to minimise convergence time. Alternatively, multiple initialisation positions and scales can be evaluated to choose the one that provides the greatest MI value.

Convexity Assumption If there are multiple peaks in the MI surface, simplex will not be guaranteed to find the global maximum. However, by initialising the simplex using the thresholds of the previous frame, the temporal coherence of the thresholds is enforced, rather than tolerating the thresholds jumping between two similarly MI valued peaks. It was also found that multiple peaks were only likely to occur in two scenarios: either there was a correlation between the detectors false positives/negatives or the signals did not share much mutual information, in which case the peaks were caused by random noise. This is discussed further in the next section when violations of the assumptions underpinning the MI thresholding approach are examined.

Efficiency Analysis In order to gauge how efficient the gradient ascent approach is compared to the full-search, the number of iterations required to converge to the correct foreground-detection thresholds was calculated for each of 200 frames in a multimodal (thermal infrared and visible spectrum) video sequence. A median background image was used for both the visible and infrared sequences. The simplex was initialised at 10 simplex sizes, from 1 to 10. In only two tests (out of 2000) did it converge to a sub-optimum solution. This occurred at the two smallest sizes. It was found that larger sizes, in general, required more iterations to converge, but were more likely to converge to a more precise solution. The average number of iterations to convergence was 26.72. When compared to a full-search, using 256 thresholds for each signal, the Simplex method is over 2400 times faster.

3.2.4 Quality Measure

The value of mutual information at the peak gives some indication as to how well the method worked and the strength of agreement between the sources. However, despite strong agreement between sources, the peak MI value may be low because there is not much common information present. For example, a very small foreground object may be detected strongly in both modalities (high agreement) but because the size of the object is small, the MI value will be low.

A potentially better way to measure performance is to compare the resulting MI score to the maximum MI score that would be achieve if there was no disagreement. This *perfect agreement map* is constructed by setting all pixels that do not have the same binary value in both map to zero. This is a simple binary *AND* operation on the two maps. The MI between two identical *perfect agreement maps* is computed as $I_{max}(X;Y)$. The quality score measure is the ratio of observed MI to maximum MI, given this level of agreed foreground pixels, $C_{1,1}$. In the practical implementation of this measure, no new maps need to be created, and the maximum MI can simply be computed using:

$$I_{max}(X;Y) = -\frac{C_{1,1}}{N}\log(\frac{C_{1,1}}{N}) - (1-\frac{C_{1,1}}{N})\log(1-\frac{C_{1,1}}{N}). \tag{3.6}$$

The quality score is then given by:

$$Q = \frac{I(X;Y)}{I_{max}(X;Y)}. \tag{3.7}$$

The quality measure, Q, will lie between 0 and 1 when $(\frac{C_{1,1}}{N}+\frac{C_{0,1}}{N})(\frac{C_{1,1}}{N}+\frac{C_{1,0}}{N}) \leq \frac{C_{1,1}}{N}$. A proof is given in Appendix A. As well as providing a good indicator for when the method has failed, it can also be used when the MI surface contains multiple peaks, for peak evaluation and selection, as we shall see later in section 3.3.3.

3.2.5 Discussion

As stated earlier, three things are assumed in the mutual information thresholding process. Firstly, the sources should have some common information relating to some event or object detected in both sources. Secondly, it is assumed that that both sources are aligned (spatially and temporally). Thirdly, it is assumed that the noise in the sources is uncorrelated. In the following experiments, the sensitivity of the proposed approach to these assumptions is investigated.

The first assumption is broken in the case of the Null hypothesis. The Null hypothesis is the case where the sensors have no common information, either because no object/event is present or because no common object/event can be detected, and any correlation is due to random noise. Experiments are performed on synthetic data to ascertain the distribution of the maximum MI value in the null hypothesis case. This permits reasoned judgements to be made about the validity of the results on real data, and whether any common information is actually present. Essentially, this allows the method to detect a violation of this assumption and thereby indicate that the conditions for successful operation are not present.

The second assumption, of *aligned data*, is a stronger one. Without aligned data, any detection in the first source cannot be corresponded to a detection in the second source. However, given that objects and events are usually spatially and temporally broad, MI thresholding can handle small mis-alignments of the data sources. The effects of mis-alignment on performance is evaluated, as well as developing a spatial smoothing procedure to best reduce these effects.

In some scenarios it may be the case that only one source of data is available and it must be split, somehow, to provide two sources for our method to work. An example of this scenario is if the data is a colour image, it could be split into a red and green band, or split into luminance and gradient values. Then it is likely that the third assumption of *noise independence* is invalid. The effects of violating this assumption are evaluated in later experiments.

3.3 Synthetic analysis of assumptions

In this section, the assumptions of mutual information threholding are examined to determine the method's behaviour when these assumptions are not met. Synthetic data is used to investigate the effects of having no common information, mis-alignment and correlated noise.

3.3.1 No common information

It may be the case that there is no common information detected in both sources. This could commonly be the case if there was no object/event to detect, for example. It is important to detect if this happens, as the method relies on common information being present. In this subsection, we examine the mutual information values produced in the case of *no common information* and we show that their distribution is well separated from the case when common information is present. This shows that it is possible to detect when no common information is present in the data sources.

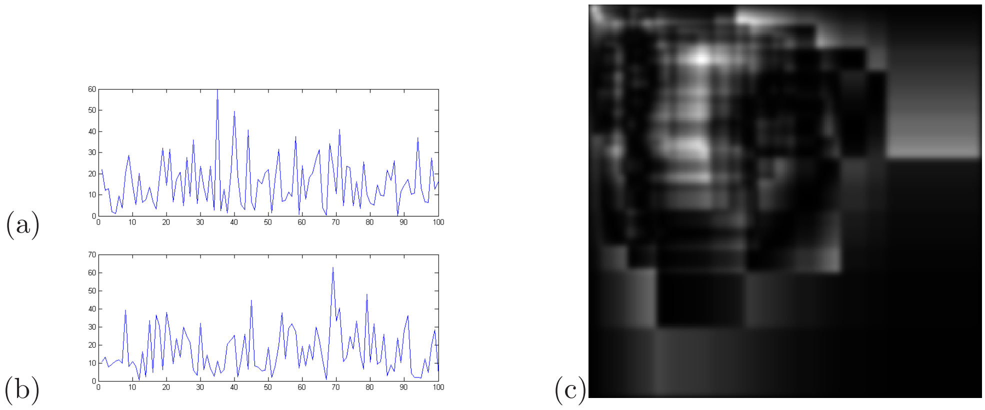
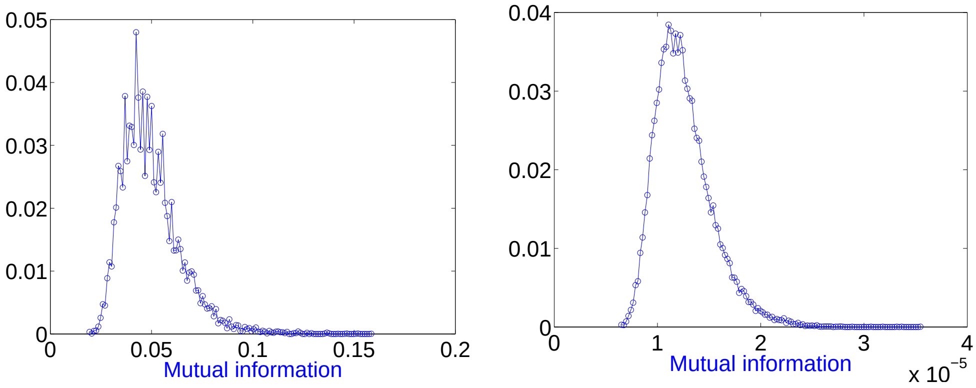


Figure 3.3: An example of two synthetic data sources (a) and (b) with no common information. The resulting mutual information surface is shown in (c).



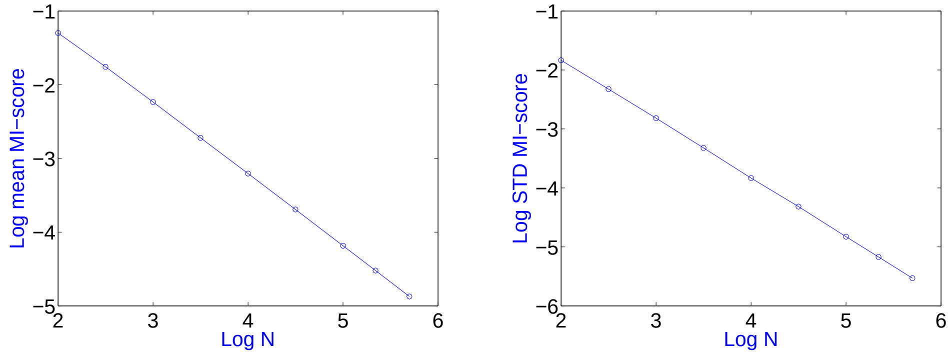
(a) $N = 100$

(b) $N = 500000$

Figure 3.4: Distributions of peak MI value in the case of the two sources having no common information, for two different numbers of samples (a)N=100, (b)N=500000.

To determine the distribution of mutual information scores when no common information is present, we generate two synthetic data sources composed of uncorrelated Gaussian noise. Over a large number of tests, we vary the sample size and examine how the distribution of MI values changes.

In this experiment, the detection scores of both sources were computed as the absolute value of a normally distributed variable with zero mean. The variance is not relevant, as 256 thresholds were selected for each source at equally spaced intervals up to the maximum value of the detection score. The absolute value is taken



(a) (b)

Figure 3.5: This plot shows the relationship between the logarithm (base 10) of the number of samples to (a) the mean and (b) the standard deviation of the peak MI distribution

to simulate a typical detection signal where most values are low. An example of the two input signals and the corresponding mutual information surface is shown in figure 3.3. It can be seen that the surface is multi-peaked due to the lack of common information. Signals of different numbers of samples were used, with $N \in \{100, 316, 1000, 3160, 10000, 31600, 100000, 220000, 500000\}$ and $10,000$ trials were performed per N value.

For a particular value of N, it was found that the peak mutual information values had a Gaussian-like distribution with a heavy tail, shown in figure 3.4. It was also found that the mean and variance of this distribution are related to the number of samples, N. In the log-log plot in figure 3.5, the logarithm of the mean and standard deviation were found to be linearly related to the logarithm of N. Therefore, using the number of samples, N, the mean and standard deviation of the distribution of MI values in the null hypothesis case of no common information between the sources are approximated by:

$$\mu = 4.5964N^{-0.9693} \tag{3.8}$$

$$\sigma = 1.4959N^{-1.0003} \tag{3.9}$$

It is likely that these formulae can be derived mathematically, without the need to resort to large amounts of empirical data. However, this derivation is left for future work.

Common information present We now simulate the case where there is a small amount of information common to both sources. The model we use for this is given in figure 3.6. Data source 1 and 2 agree on a small number of correct foreground pixels, denoted A. They also each contain foreground not found in the other source, with $2B$ disagreed pixels in total. We examine two cases: (i) when there is a reasonable amount of common information present in the data, and (ii) when there is only a very low level of common information present. In both cases, we use an image size of 256×256, therefore $N = 65536$ and the brightness of all foreground pixels is 100 (before the addition of noise).

In the first case, we set $A = 80^2$, $B = 57^2$ and added Gaussian Noise with a standard deviation of 50. This means that the data is made up of approximately 10% agreed foreground pixels, 10% disagreed foreground pixels and a signal-to-noise-ratio (SNR) of 2. An example of the data sources are shown in figure 3.7.

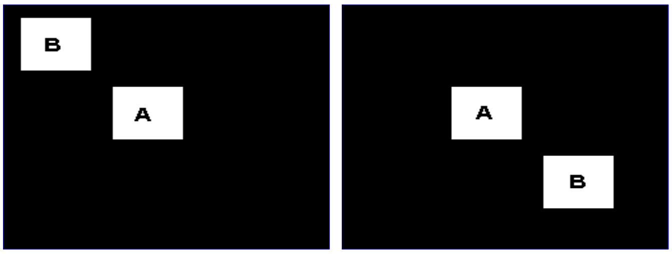
In the second case, we set $A = 25^2$, $B = 35^2$ and added Gaussian Noise with a standard deviation of 50. In this case, the data is made up of approximately 1% agreed foreground pixels, 4% disagreed foreground pixels and a signal-to-noise-ratio (SNR) of 2. An example of the data sources are shown in figure 3.8.

For both cases, we performed mutual information thresholding on the sources and computed the peak value of mutual information surface, m. We then computed a separation value, S, that measured how close m was to the distribution of no common information.

$$S = \frac{m - \mu}{\sigma} \tag{3.10}$$

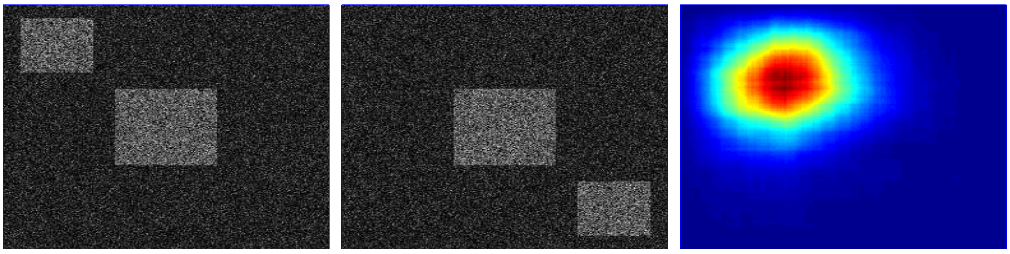
This was done $100,000$ times for each of the two cases, and the distributions of the separation values are shown in figure 3.9. Since the distributions in figure 3.4 are Gaussian-like, a separability value of about 4 would place the MI value far outside the distribution of no common information. It can be clearly seen that, even when there is only a very small amount of common information shared by the sources, as in figure 3.9(b), the separation values are greater than 10. This suggests that detecting the presence or absence of common information should be possible.

Conclusions From these synthetic tests, we have shown that, the distributions of *no common information* and *some common information* are well separated. Even when there is very little common information, the mutual information values returned by the algorithm are more than sufficient to separate it from the case of having no common information.



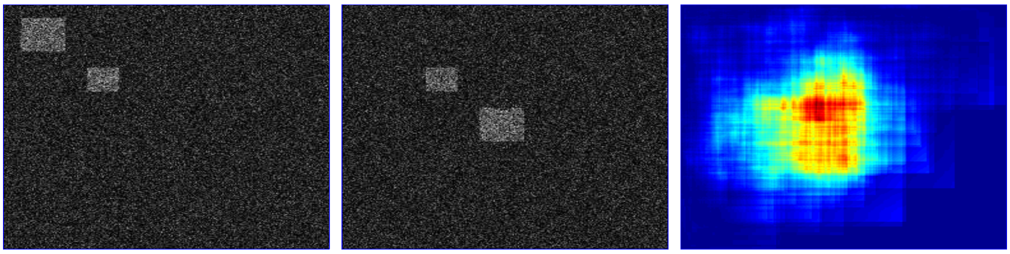
(a) Data source 1 (b) Data source 2

Figure 3.6: Model for sources with common information present: block 'A' represents agreement, and 'B' represents disagreement.



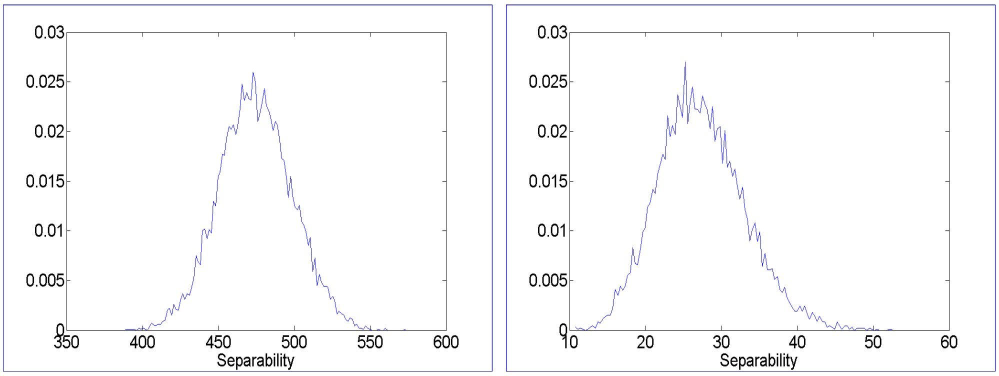
(a) Data source 1 (b) Data source 2 (c) MI

Figure 3.7: Examples of the two synthetically generated data sources when there is a reasonable amount of common information shared by the sources



(a) Data source 1 (b) Data source 2 (c) MI

Figure 3.8: Examples of the two synthetically generated data sources when there is a very small amount of common information shared by the sources



(a) (b)

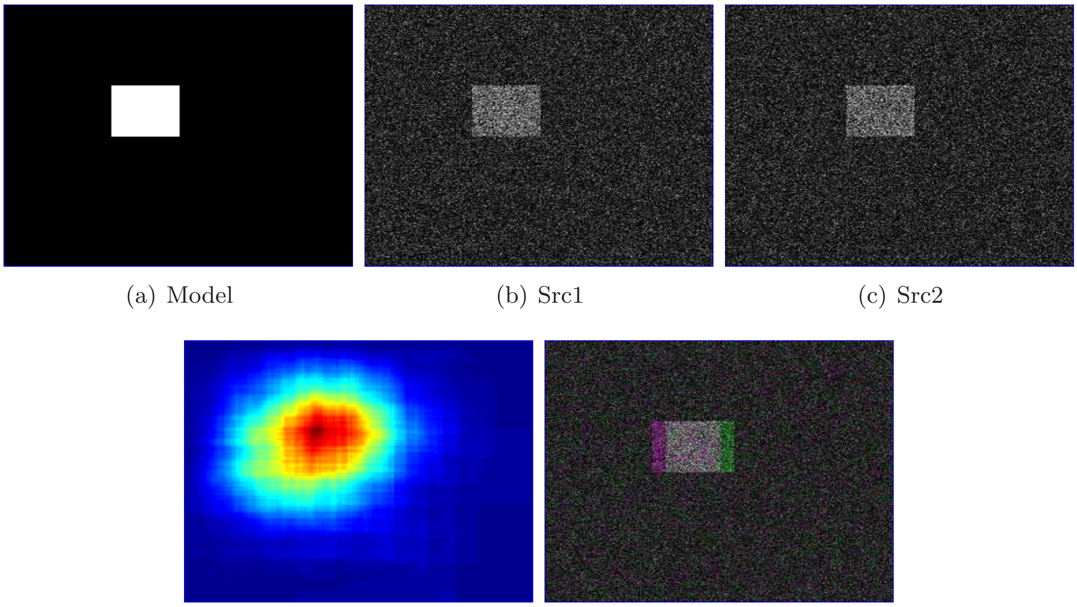
Figure 3.9: Distributions of separation values for (a)some common information and (b)low levels of common information between data sources

3.3.2 Poor alignment

The method of MI thresholding relies on the data sources being aligned to detect common information. Small misalignments may not affect the performance considerably, since objects/events are usually spatially/temporally broad and would still overlap in the two data sources. We now investigate how misalignment affects the performance of MI thresholding.

Figure 3.10(a) shows the model we used for this experiment. A simulated square foreground object, 50 pixels on its side, with a brightness of 100, is placed in the centre of both data sources. The square is then shifted horizontally in data source 2 and Gaussian noise is added to both sources with a standard deviation of 50. Figures 3.10(b) and 3.10(c) show examples of the data sources.

The peak MI surface value is computed for different amounts of horizontal shift. Figure 3.11 shows how the peak MI value decreases as the misalignment shift is increased. When the shift is greater than 80% of the width of the foreground object, the peak MI value cannot distinguish any common information in the data sources. Any misalignment of less than 80% reduces the MI score but the method still performs well and selects acceptable thresholds. This can also been seen by observing the returned thresholds, as shown in figure 3.12. The figure of "80% of the width" is not general to all cases and smaller foreground objects would not be expected to tolerate misalignments up to this fraction of their width, at similar noise levels to this example.



(a) Model

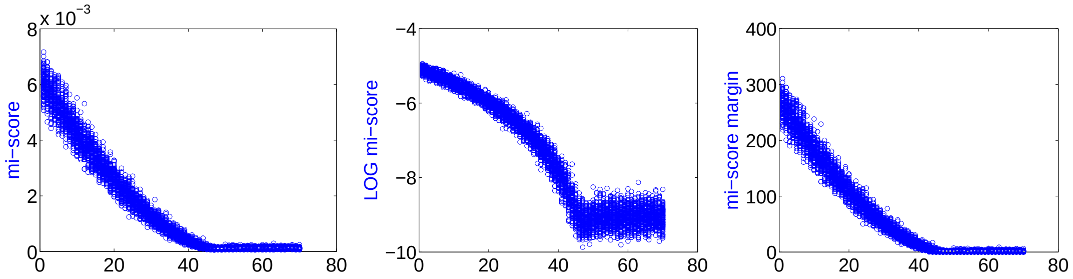
(b) Src1

(c) Src2

(d) MI

(e) Overlap

Figure 3.10: Misalignment Experiment: (a) Model image used, (b) and (c) show examples of two synthetic misaligned data sources with a shift of 10 pixels between them, (d) MI surface, (e) Overlap of images in (b) and (c).

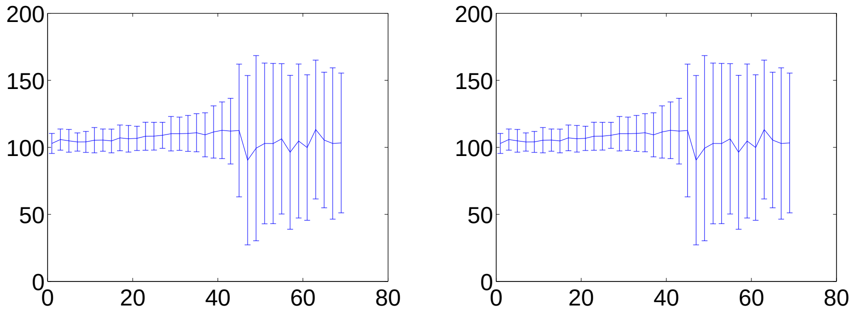


(a) Linear Scale

(b) Log Scale

(c) Separation

Figure 3.11: Plot of peak MI value versus the misalignment shift in (a)linear and (b)log scale. Subfigure (c) shows the separation value for the MI score. The x-axis represents the horizontal shift in pixels.



(a) source 1

(b) source 2

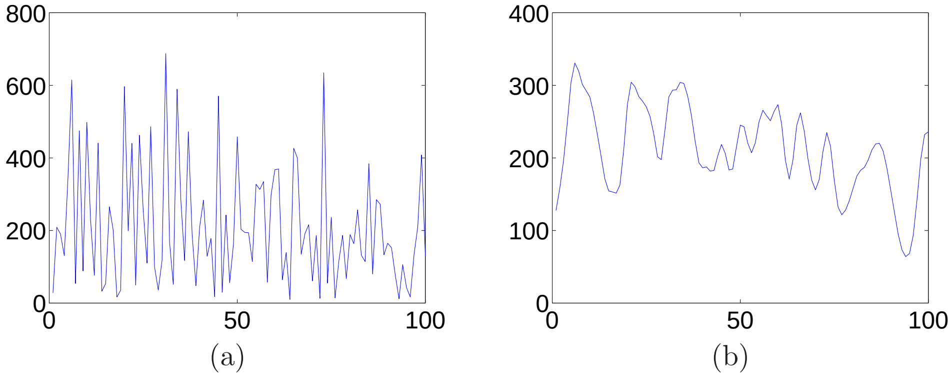
Figure 3.12: Plot of threshold values (mean and one standard deviation) for (a) source 1 and (b) source 2.

Data smoothing effects In order to cater for data sources that are not perfectly aligned, one may wish to perform data smoothing on the two sources before computing their mutual information thresholds, in order to better compensate for the spatial misregistration. Figure 3.13 shows an example of the effects of smoothing on a signal. The original signal in 3.13(a) becomes the smoothed signal 3.13(b). Visually, the smoothing operation can be seen to increase the correlation between neighbouring samples and to reduce the information content of the signal. Reducing the information content is similar to reducing the number of samples, N.

As can be seen from equation (3.8), a lower number of samples (N) means a higher average value of MI in this case where there is no common information. Experiments on synthetic data showed this to be true. Using two synthetically-generated absolute Gaussian noise signals, as used previously, with $N = 100$ samples each, both signals were filtered using a Gaussian filter of width σ. By varying the value of σ, and repeating this procedure on $250,000$ signal pairs, one can obtain an understanding of how smoothing affected the results of MI thresholding. For this experiment, σ was varied from 0 to 6 in increments of 0.25, and $10,000$ tests were run for each σ value.

Figure 3.14 shows how the average maximum MI value obtained between two sources changes when both sources are filtered with a Gaussian filter of width, σ, as σ increases. It can be seen that at $\sigma \leq 0.5$, the average MI value is almost constant. However, when $\sigma \geq 0.5$, the average MI value increases linearly with increasing σ.

Optimal data smoothing Since smoothing seems to increase the MI score, regardless of whether of not there is any common information present, an important consideration is if there is a way to determine the optimal σ value when common information is



(a)

(b)

Figure 3.13: (a) Raw data signal with N=100 samples, (b) Signal smoothed with Gaussian filter of $\sigma = 2.5$

present. Imagine a scenario whereby an event is detected using a motion sensor and an audio sensor. The event could be people entering a room and talking. The event may last a few minutes, but neither motion nor talking would be present in every sample, so those samples would be interpreted as disagreement. We simulate such a scenario using the sources in figure 3.15(a) where the *event* is represented by a square that is *punctured by random holes* to represent that motion and audio do not always happen together. The experimental parameters were as follows: The image size was 250×250 pixels, the square event was 50×50 pixels in size and the holes were created by setting the event's pixels to zero with a probability of 0.5. The event pixels' brightness was set to 5 and random noise of standard deviation 1 was then added. For various values of σ between 0 and 10, the two sources were convolved with a Gaussian kernel of this width, as shown in figure 3.15 (b) and (c). The resulting plots of MI score and quality, shown in figure 3.16, distinctly identify a unique σ value. This value is approximately $\sigma = 1.5$ and produces the maximum quality score, as well as a sharp *corner* in the MI score plot. In this experiment, the smoothing is exploiting the spatial information due to the close proximity of detections. The MI score rises sharply when there are spatial correlations between the detections. At the graph's corner, the spatial correlations have been exhausted and the MI rises only slowly.

3.3.3 Correlated noise in sources

In this section, the effects of correlated noise on the algorithm are investigated. We examine the case of when the variance in some of the pixels is correlated and see how the MI surface is affected. This could be the case when both data sources are derived from the same sensor, or it could also be present in the pixels corresponding to fluctuating

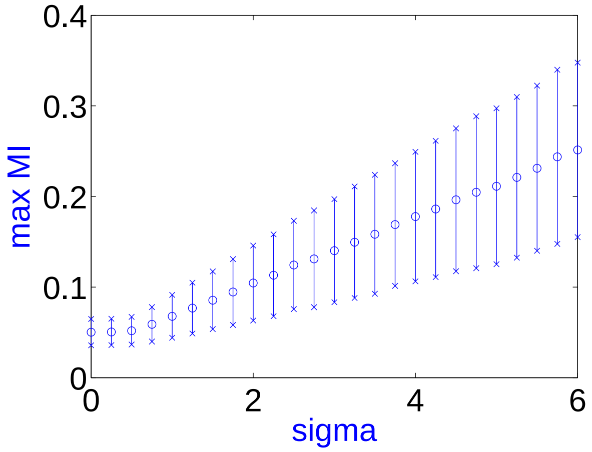


Figure 3.14: Plot of average MI value returned using MI thresholding when two noise signals with no common information are Gaussian filtered using a width of σ. Vertical bars indicate one standard deviation. The signals used has $N = 100$ samples.

background objects such as trees or water. The synthetic model we use for this is made up of three parts; two parts background (A,B) and one part constant foreground (C). Part A models normal background, part B models highly variant background and part C models the foreground. Gaussian noise of variance N_1 is superimposed on A and C. Gaussian noise of variance N_2 is superimposed on part B. The absolute value of the signal is then computed. Initially we set $N_1 = N_2$. Figure 3.17(a) shows examples of the synthetic data used at various levels of noise. With low noise, there is a strong peak in the MI surface at the correct thresholds. As N_2 is increased, another peak emerges in the surface, eventually overtaking the correct peak as the maximum. In part (b) of Figure 3.17 the MI surfaces corresponding to the signals in (a) are shown.

By selecting the thresholds corresponding to the peak, figure 3.18(a) shows the thresholds that will be selected. It is clear from this plot that, at a certain level of noise (approximately $N_2 = 125$), the incorrect peak will be selected when it becomes higher than the correct one. This is indicated by the sharp drop in threshold value. However, it was found that by examining multiple peaks and choosing the one with the greatest quality score, the correct peak can be identified. In this experiment, the surface was smoothed with a Gaussian filter with $\sigma = 2.0$ to eliminate spurious maxima. Figure 3.18(b) shows the thresholds selected at the second highest peak. The corresponding quality scores for the top 2 peaks are shown in figures 3.18(c) and 3.18(d). The highest quality score is a strong indicator of the correct peak and this will be used later in this chapter, in section 3.4, to develop an extended version of our algorithm that is more

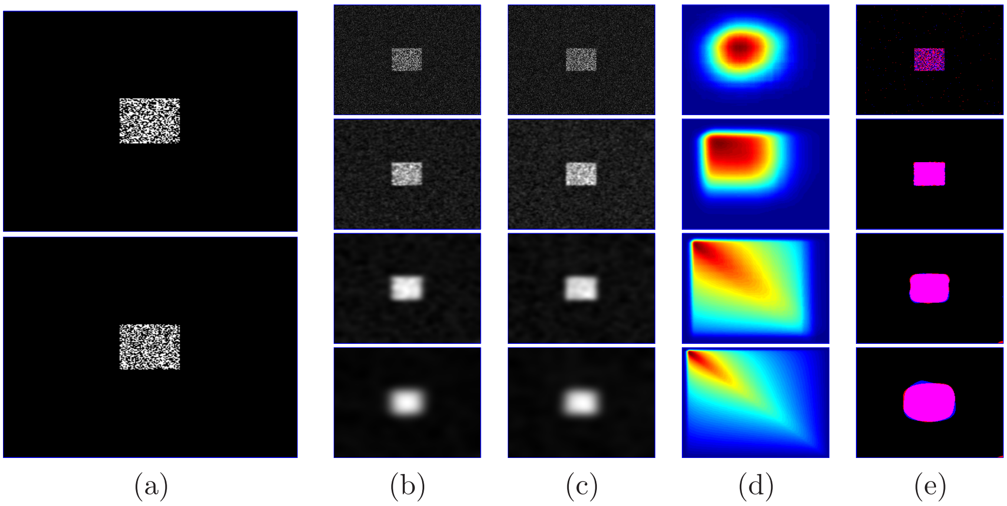


Figure 3.15: Optimal smoothing experiment: (a) shows the two input data sources without noise, (b) and (c) show these sources at various degrees of smoothing ($\sigma \in \{0, 1.5, 5, 10\}$), (d) the corresponding MI surface and (e) the resulting thresholded signals superimposed.

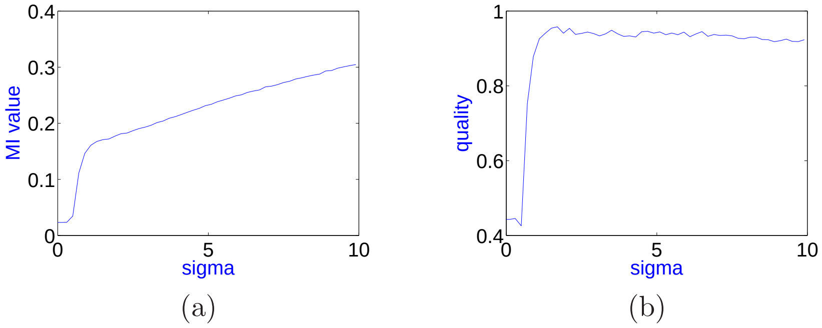
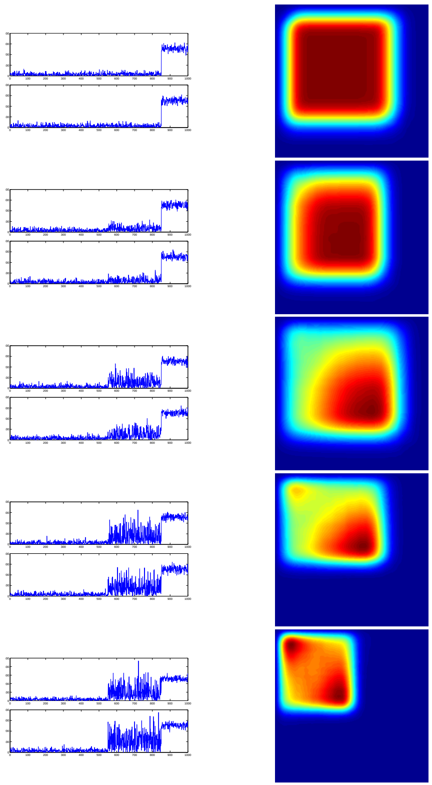


Figure 3.16: These graphs show (a) the MI score and (b) the quality measure resulting from using the smoothed sources at various different values of σ.



(a) (b)

Figure 3.17: The left column shows examples of input signal pairs and the right column shows their corresponding mutual information surfaces. As the variance of the correlated noise increases, a new peak can be seen to rise in MI surface.

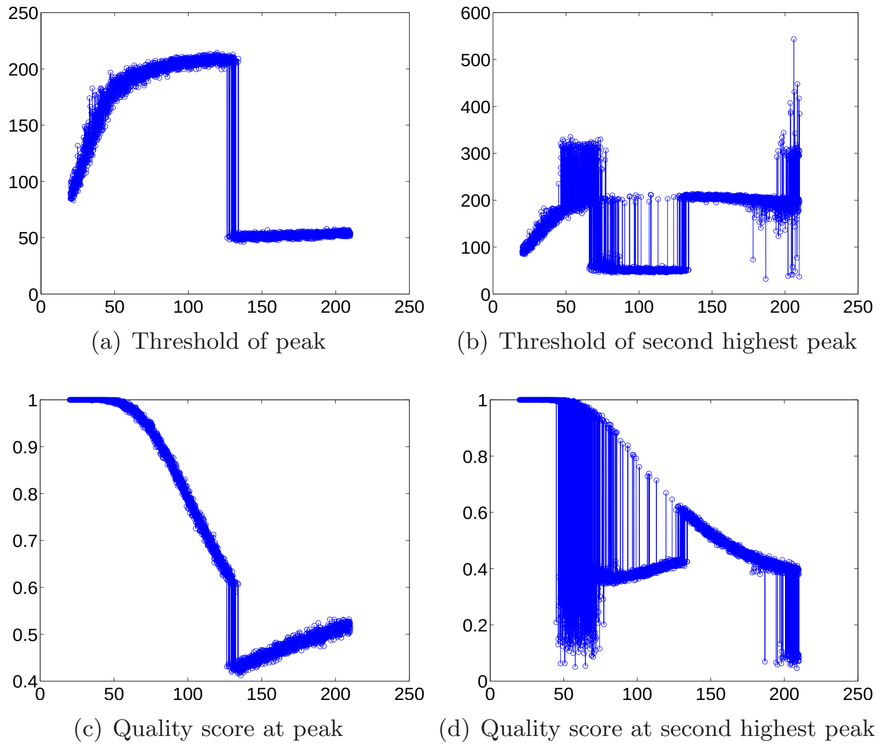
robust to multiple peaks caused by correlated noise. As the graphs in figure 3.18(c) and (d) show however, beyond a certain high level of noise, the quality value will not be able to distinguish the peaks.

3.3.4 Discussion

This section questioned the assumptions of MI thresholding and examined the effects of using data that did not comply with these assumptions.

Firstly, the distribution of the MI score in the case of no common information was approximated and shown to be well separated from the case where common information is present.

Secondly, the method's robustness to misalignment was investigated. It was shown that if the size of the misalignment is smaller than the typical object width, then this is usually tolerated and acceptable thresholds are produced. A technique for optimally



(a) Threshold of peak

(b) Threshold of second highest peak

(c) Quality score at peak

(d) Quality score at second highest peak

Figure 3.18: The threshold selected (for source 1) using the first and second highest peaks in the MI surface is shown in (a) and (b) respectively. The corresponding quality score plots are shown in (c) and (d). On the x-axis is shown N_2, the amount of correlated noise in part B of the signal.

smoothing the data sources was also developed, in order to exploit the spatial proximity of detections.

Thirdly, it was shown that if correlated noise was present in the sources of data, it was manifested as an additional peak in the MI surface. If this peak is greater than the correct peak, the synthetic data indicated that it is usually possible to distinguish the peaks using the performance quality measure.

Real data might contain small violations of all three assumptions and this could lead to additional problems that are difficult to fully analyse with synthetic data. For example, there may be nothing in the scene, and hence no common information to exploit, but if correlated noise were present, this could be very difficult to distinguish from common information. The next chapter more fully explores how closely real data complies with these assumptions.

3.4 Extended algorithm

As shown in section 3.3.3, the use of the quality measure to evaluate multiple peaks in the MI surface can make the algorithm robust to correlated noise in the sources. The extended version of the algorithm that we term *extended mutual information thresholding* (EMIT) is now described in detail.

First, the MI surface is computed as normal. All local peaks on this surface are detected and stored as P_{orig}. The surface is then smoothed with a Gaussian filter of width σ. All peaks on this smoothed surface are then extracted as P_{new} and then each peak in P_{new} is replaced by its nearest neighbour in P_{orig}. Next, any threshold pairs in P_{new} that do not have $(\frac{C_{1,1}}{N} + \frac{C_{0,1}}{N})(\frac{C_{1,1}}{N} + \frac{C_{1,0}}{N}) \leq \frac{C_{1,1}}{N}$ are removed from consideration, as this could cause the quality score to fall outside the 0..1 range. Low scoring peaks are then removed by computing the Rosin threshold of a histogram of the MI values of the entire surface and retaining only those peaks with MI scores above this threshold. Finally, of the remaining peaks, the peak with the highest quality score is used to determine the final thresholds.

The EMIT algorithm is essentially a heuristic to counter the problem of multiple peaks in the MI surface. With regards to the σ smoothing parameter, it was found that values of 1 and above produced similar results, whereas values less than this often led to the selection of high thresholds, with high precision but low recall. Values between 1 and 2 were found to be sufficient to remove spurious peaks causes by surface noise.

3.5 Method comparisons using synthetic data

In this section, experiments are conducted to compare mutual information thresholding to the other most frequently cited and most common dynamic thresholding methods.

3.5.1 Selected non-parametric thresholding approaches

To evaluate the MI thresholding algorithm, experiments comparing this algorithm to other dynamic thresholding algorithms are now described. Four of the most frequently cited thresholding methods were chosen and they are briefly described below. The methods used are those of Kapur, Kittler and Illingworth, Rosin and Otsu.

Kapur's method [65] chooses a threshold to maximise the sum of entropies in a two class system (the change and no-change classes). It was ranked first in an evaluation of change detection methods on both synthetic and real data [118]. Additionally, it was also ranked highly in other thresholding studies [6, 123]. A minimum error thresholding method was proposed by Kittler and Illingworth [69]. It was ranked first in an extensive image thresholding survey [123]. The unimodal thresholding method of Rosin [116] specifically targets the change detection task by noting that most pixels come from the no-change class. Otsu's thresholding method [99] chooses a threshold to maximise the between-class scatter between the background and foreground classes. Otsu's method performed well in [123], though not explicitly for change detection. Otsu's method is also a standard function in MATLAB, under the function name *graythresh*.

These 4 methods are compared to the MI thresholding algorithm using synthetically generated data in this chapter. The next chapter compares their performance on real data applications.

3.5.2 Evaluation measures

Any particular detection score can be thresholded using all possible thresholds and this, along with the ground truth data, can produce a *Precision-Recall* (PR) curve. This PR curve relates to the tradeoff between false positives and missed detections depending on the chosen threshold. Each dynamic thresholding algorithm will essentially position the system at a particular point on the curve. Although there is no consensus on what the absolute optimum threshold is, there are a number of approaches in the literature proposing *optimality* measures that are useful for this task.

As an intuitive example, position A in figure 3.19 is not a good result, as we can lower the threshold to achieve a higher recall with a negligible decrease in precision. Position B would seem to be closer to the optimum choice.

In order to ascertain whether the algorithms have performed well, five different optimality measures are considered. The optimality measures used are (i) the minimisation of a cost function, relating the probabilities of false positives and missed detections, (ii) the F-1 measure that balances the precision and recall trade-off, (iii) equalising false positive and missed detection rates, (iv) minimising the distance between the ideal solution and the chosen point on the precision-recall curve and (v) maximising the Jaccard coefficient. These five measures are described in the following paragraphs.

Cost function minimisation If we illustrate our two-class discrimination problem as in figure 3.20, we see that any threshold will cause some portion of each distribution X and Y to fall on the incorrect side of the threshold divide. For a given application, the optimal threshold will be one which minimises a cost function relating the cost of

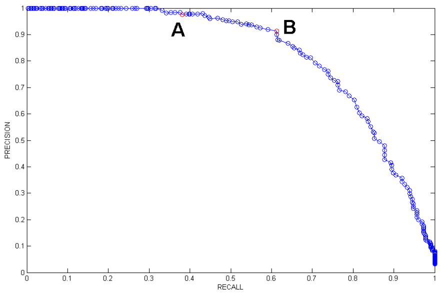


Figure 3.19: Precision-Recall (PR) curve example with two possible operating points marked, A and B.

false positives with the cost of missed detections. This cost function can be written as:

$$K = C_1(bP(Y)) + C_2(cP(X)) \tag{3.11}$$

where C_1 and C_2 are the costs associated with missed detections and false positives respectively. The two terms b and c fractions of the distributions that fall on the incorrect side of the threshold. $P(X)$ and $P(Y)$ are the a priori probabilities of a sample coming from X and Y.

For a given threshold, the precision is defined as the fraction of detections that are correct. The recall is defined as the fraction of the total number of events that are detected. The precision and recall can be computed as follows:

$$p = \frac{dP(Y)}{dP(Y) + cP(X)} \tag{3.12}$$

$$r = \frac{dP(Y)}{dP(Y) + bP(Y)} \tag{3.13}$$

Noting that each probability distribution's area is equal to 1, we get

$$a + c = 1 \tag{3.14}$$

$$d + b = 1 \tag{3.15}$$

We can rewrite equations 3.13 and 3.12 as

$$r = \frac{d}{d+b} = d = 1 - b \tag{3.16}$$

$$cP(X) = \frac{dP(Y) - dpP(Y)}{p} = \frac{rP(Y)(1-p)}{p} \tag{3.17}$$

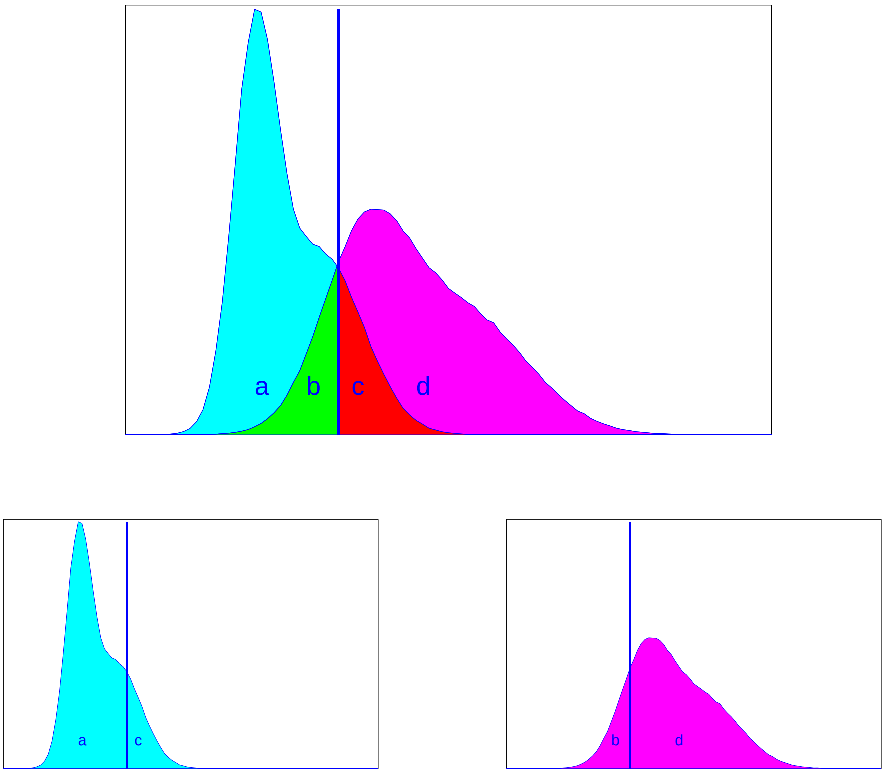


Figure 3.20: Illustration of thresholding to distinguish two distributions: The cyan distribution represents the no-change class, whereas the magenta distribution represents the change class. The areas b and c indicate the probability of obtaining a false negative and a false positive respectively.

These equations can be inserted into our original cost equation as follows:

$$\begin{aligned} K &= C_1(1-r)P(Y) + C_2\frac{rP(Y)(1-p)}{p} && (3.18)\\ \Rightarrow K &= P(Y)\left[C_1(1-r) + C_2\frac{r(1-p)}{p}\right] && (3.19)\\ \Rightarrow K &= P(Y)C_1\left[(1-r) + C\frac{r(1-p)}{p}\right] && (3.20)\end{aligned}$$

where $C = C_2/C_1$ is the ratio of the cost of false positives to the cost of missed detections. A more compact form of the cost function can then be obtained by noting that the optimal choice of threshold is not affected by dividing the cost function by a positive constant. Dividing by $P(Y)C_1$ gives our final cost equation in terms of precision, p, recall, r, and trade-off cost ratio, C:

$$K = r(\frac{C}{p} - C - 1) + 1 \tag{3.21}$$

This cost has its minimum value of zero when $p = r = 1$. In the experiments conducted, C was set equal to one, giving the cost function to minimise as:

$$K_1 = r(\frac{1}{p} - 2) + 1 \tag{3.22}$$

F-1 measure A measure used frequently in information retrieval to measure performance is the F_1-measure. According to Yang and Liu [158], this measure was first introduced by C. J. van Rijsbergen [147]. This measure equally weights precision, p, and recall, r, in the following form:

$$F_1 = \frac{2pr}{p + r} \tag{3.23}$$

This can be interpreted as the harmonic mean of precision and recall. Relating to figure 3.20, if the X distribution is considered background and the Y distribution foreground, then using equations (3.12) and (3.13) the F_1-measure can be expressed as

$$F_1 = \frac{2dP(Y)}{P(Y) + dP(Y) + cP(X)} \tag{3.24}$$

with $P(X)$ and $P(Y)$ being the a priori probabilities of background and foreground.

Equal Error Rates Another approach commonly used in threshold selection is to equalise the error rates. That is, to choose the threshold such that the probability of getting a false positive (FP) is equal to the probability of getting a missed detection (MD). These rates can be written as follows:

$$P_{FP} = P(X)\frac{c}{a + c} = P(X)c \tag{3.25}$$

$$P_{MD} = P(Y)\frac{b}{b + d} = P(Y)b. \tag{3.26}$$

Using equations (3.12), (3.13) and (3.15), the condition of equal error rates can be expressed in a much simpler form with some simple manipulation.

$$P(X)c = P(Y)b = P(Y)(1 - d) \tag{3.27}$$

$$\Rightarrow P(X)c + P(Y)d = P(Y) \tag{3.28}$$

$$\Rightarrow 1 = \frac{P(Y)}{P(X)c + P(Y)d} \tag{3.29}$$

$$\Rightarrow d = \frac{P(Y)d}{P(X)c + P(Y)d} \tag{3.30}$$

$$\Rightarrow r = p \tag{3.31}$$

Therefore by selecting the threshold that sets the operating point on the PR curve to where it crosses the line connecting (0,0) to (1,1), the equal error criterion is achieved.

Optimal distance The final measure we used in optimal threshold selection is to minimise the distance between the point (p,r) and the ideal solution of (1,1), where precision and recall are perfect. So the optimal threshold, according to this measure, is the one that selects the position on the p-r curve closest to (1,1), thus minimising D, given by:

$$D(p, r) = \sqrt{(p-1)^2 + (r-1)^2} \tag{3.32}$$

The Jaccard coefficient The Jaccard coefficient is a performance measure proposed in [132]. In terms of true-positives, false positives and false negatives (missed detections), the Jaccard coefficient, J, is written as TP/(TP + FP + FN). In terms of the variables in figure 3.20, we write it as:

$$J = \frac{P(Y)d}{P(Y)d + P(X)c + P(Y)b} \tag{3.33}$$

Using equations (3.12) and (3.13), it can be rewritten simply as:

$$J = \frac{pr}{p + r - pr} \tag{3.34}$$

Measure visualisation To summarise, the five measures used to assess the quality of the selected thresholds are : (i) minimising a cost function, (ii) Maximising the F-1 measure, (iii) Equalising the error rates, (iv) minimising the distance to the optimal point and (v) Maximising the Jaccard coefficient. These five criteria can easily be reformulated into the maximisation of one of five score measures:

$$M_1(p, r) = \frac{p}{r - 2rp + 2p} \tag{3.35}$$

$$M_2(p, r) = \frac{2pr}{p + r} \tag{3.36}$$

$$M_3(p,r) = \frac{p+r}{2\max(p,r)} \tag{3.37}$$

$$M_4(p,r) = \frac{2-(p-1)^2-(r-1)^2}{2} \tag{3.38}$$

$$M_5(p,r) = \frac{pr}{p+r-pr} \tag{3.39}$$

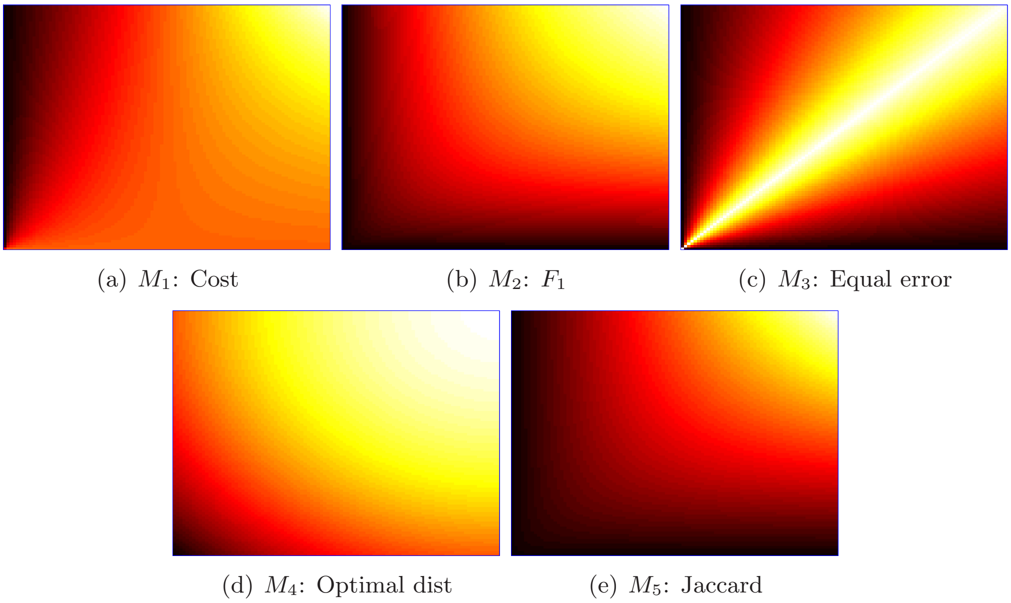
each of which is bounded between 0 and 1, and takes on its maximum value when $p = r = 1$. Indeterminate values (0/0) are mapped to zero.

In figure 3.21, a visualisation of the five measures is shown as a more intuitive aid to how they assess precision and recall quality. Subfigures 3.21 (b), (d) and (e) have the most obvious interpretation, since they give higher scores to points with higher precision or recall. The first measure, in 3.21(a) seems counter-intuitive, since at low values of precision, higher values of recall are actually punished. This is because when recall increases, it means that more true positives are found but because precision is kept constant is means that the number of false positives also increases. For example, if 5 more true positives are found by lowering the threshold, recall will increase, but if $p = 0.1$ then it will mean that we get 45 more false positives. Figure 3.21(c) is also counter-intuitive, since it gives the same score to point(0.2, 0.2) and to point (1,1). This can be countered by the fact that any PR curve will only pass through the equal-error line once, so that kind of ambiguity would not arise.

3.5.3 Synthetic data experiments

Synthetic data are useful for quantitative comparison of algorithms, since the ground truth can be computed without manual annotation. Synthetic difference images were generated of similar form to figure 3.22(a). Two foreground regions of different sizes and different constant signal strengths (brightnesses) were used. Pairs of these images were created with identical size and signal parameters but with independent Gaussian noise super-imposed. Each pair was fed into the MI thresholding algorithm to compute appropriate thresholds. Only one of the images was used for the other methods, since they cannot take advantage of multiple sources of information.

The parameters for the experiment were as follows. Each image was of size 100×100 pixels. Gaussian noise with standard deviation 5 was added to all images and the absolute value of each pixel was computed. The sizes of the two square foreground regions were varied from a side of 5 pixels to 40 pixels, in increments of 5 pixels. The brightness, or signal strength, of the regions varies from 10 to 25 in increments of 1. In total, $16,384$ tests were run (8 sizes and 16 brightnesses for 2 regions $\rightarrow 8 \times 8 \times 16 \times 16 =$



(a) M_1: Cost

(b) M_2: F_1

(c) M_3: Equal error

(d) M_4: Optimal dist

(e) M_5: Jaccard

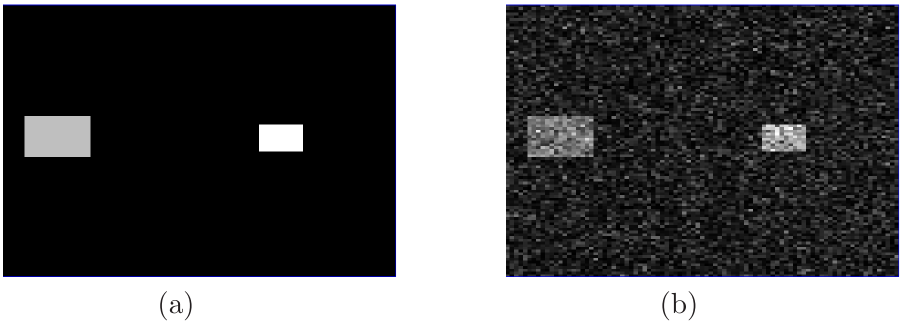
Figure 3.21: Visualisation of threshold score measures: (a) cost function minimisation, (b) F-1 measure maximisation, (c) error rate equalisation, (d) distance minimisation and (e) Jaccard coefficient. Precision and recall are on the x-axis and y-axis respectively.

$16,384$). The results of this experiment are shown in table 3.1. The mutual information thresholding method was found, on average, to outperform all other methods using any of the five optimality criteria. The Kapur and Kittler methods both performed well also.

Measure	Rosin	Kapur	Otsu	Kittler	MI
M_1	0.6676	0.7241	0.6058	0.7046	**0.7505**
M_2	0.7502	0.7792	0.6901	0.7807	**0.8077**
M_3	0.8876	0.9059	0.8426	0.9208	**0.9469**
M_4	0.9166	0.9311	0.8790	0.9333	**0.9444**
M_5	0.7770	0.7975	0.7675	0.7920	**0.8155**

Table 3.1: Mean results for each method for all five measures.

Gradient as second data source If a second data source were not available, it could be created by taking the gradient magnitude of the first data source. We investigate the performance of this approach by rerunning the previous experiment, except this time replacing source 2 with the magnitude of the gradient of source 1. The gradient was computed using the Sobel operator. Results are shown in table 3.2 and show that



(a)

(b)

Figure 3.22: (a)An example of synthetic input image before noise addition, and (b)after noise is added

MI thresholding again outperformed all other methods.

Measure	Rosin	Kapur	Otsu	Kittler	MI
M_1	0.6667	0.7241	0.6059	0.7046	**0.7496**
M_2	0.7498	0.7794	0.6902	0.7806	**0.8071**
M_3	0.8877	0.9065	0.8426	0.9205	**0.9462**
M_4	0.9164	0.9312	0.8791	0.9332	**0.9448**
M_5	0.7766	0.7974	0.7675	0.7919	**0.8142**

Table 3.2: Mean results for each method for all five measures using gradient as a second data source.

3.6 Online MI thresholding

As previously mentioned, the use of MI thresholding is not confined to foreground detection. We can examine the realm of wireless sensor networks, where the goal is to reduce the cost of individual processing units by reducing its hardware capabilities, and to improve performance by distributing the sensing capabilities over a large area [5]. Therefore in order to reduce sensor node costs, the processing power of each distributed sensor is limited in terms of speed and storage capacity. Additionally, most sensor nodes are not connected to a power supply, thus have limited battery life which must be used sparingly [111]. Thresholding is a process that fits well into this scheme.

For example, in an audio event detection scenario, where each wireless sensor has a microphone, it is impractical for every node to broadcast the audio is receives back to a central hub for processing. The power consumption would be too great. Ideally, the node should only transmit when an important event occurs, such as when the audio volume exceeds a threshold, indicating that some event is taking place [128].

In this scenario, it would seem that mutual information thresholding is impractical, since the algorithm requires both signals in order to produce the mutual information surface. This is not the case however. In fact, it is possible for two sensors to adjust their respective thresholds to maximise the mutual information between their detected events, without knowledge of the signal data of the other, but simply by observing the binary detection results of each other. In this section, this procedure is demonstrated, along with an example of it in operation.

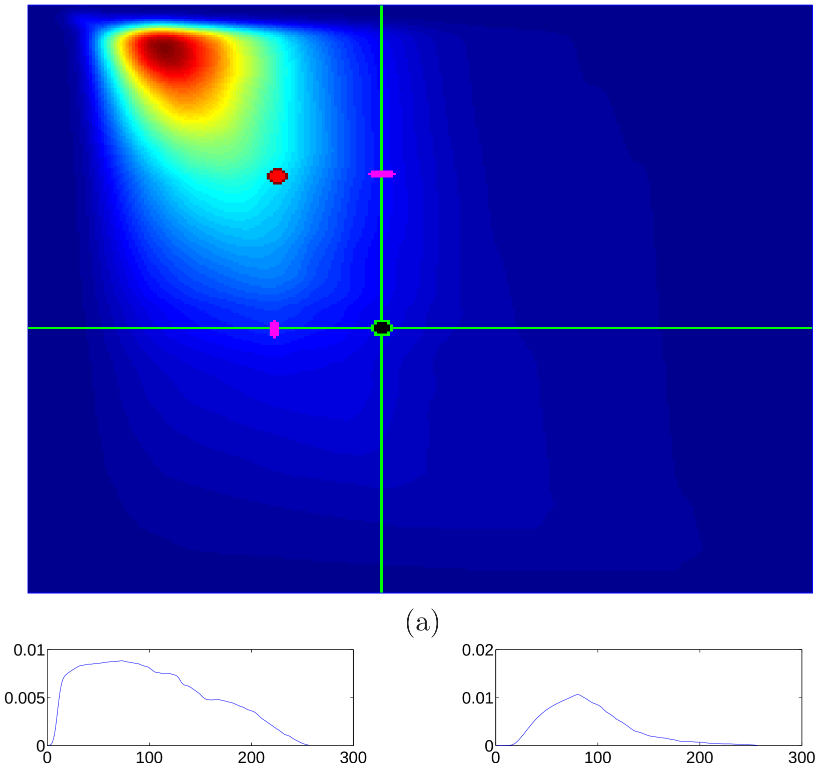
A sensor node receives only the binary detection signal from another node. It can choose its threshold by performing the exact same MI thresholding algorithm as before, but this time only using a single threshold for the second source (e.g. 0.5). It need not store its entire data signal, but only two histograms: one for its data when the other node outputted 0 and the other for its data when the other node outputted 1. The number of bins can be dictated by the memory available to the node. Each node adapts it own threshold to maximise the mutual information between its own output and the other node. In terms of movement across the MI surface, this is either a horizontal or vertical movement to the highest point along that line.

Figure 3.23 demonstrates how the procedure works. The system begins at operating point $(115, 140)$, shown by the green circle. This corresponds to a threshold of 115 for sensor 1 and a threshold of 140 for sensor 2. Each sensor adapts their own threshold by observing the binary output of the other, and computing the MI for each possible threshold it could use. For sensor 1, it wishes to know how to vary its threshold (the x-coordinate), and this means computing the plot of the horizontal line through the operating point, shown in figure 3.23(c). Similarly, sensor 2 computes the plot shown in figure 3.23(b), which is the vertical line through the operating point. The magenta bars indicate where the maxima are located on each linear plot. The new operating point becomes $(80, 73)$, indicated by the red circle.

3.6.1 Experiment

Using synthetic data, we simulate the procedure of online MI thresholding for two nodes. A separate data stream is created for both nodes. Each stream consists of 95% zeros (representing no event) and 5% ones (representing events). Gaussian noise of standard deviation σ_{Noise} is added to the data signal.

Both node thresholds are initialised at value 1. Each node will utilise the MI thresholding algorithm, using its own data signal and the binary signal of the other node as input to the algorithm. The threshold list for the other data source will be



(a)

(b)

(c)

Figure 3.23: Online MI: illustration of the algorithm. (a) Entire MI surface, (b) and (c) show the vertical and horizontal cross-sections, respectively, through the operating point. By choosing thresholds to maximise MI separately along each dimension, the new operating point (shown in red) lies closer to the optimum.

$A = \{0, 1\}$. In terms of storage requirements, each node can simply maintain two histograms. The first histogram counts occurrences of its own values when the other node outputted zero. The second histogram counts occurrences of its own values when the other node outputted one. After N samples have been received, each node will execute the MI thresholding algorithm and adapt its threshold to maximise MI. If either histogram is empty, the algorithm is not executed, since there is no *information* present to exploit. An empty histogram could result from there being no events to detect, or if the other node has set its threshold very high or very low.

Two experiments were performed to evaluate the performance of online MI and to investigate the correct choice of N. As we show, the correct choice of window size, N, can be crucial in optimising performance. The first experiment uses *static noise* and the second uses *variable noise*. For example, the case of *variable noise* might arise when using audio sensors within a building whose air-conditioning is scheduled to turn on and off automatically, causing a periodic change in the noise variance. Both experiments were performed using a set of values for N, each set using the exact same pairs of input data. The values of N used were $\{10, 20, 50, 100, 250, 500, 1000, 2500, 5000\}$. A total of $250,000$ samples were used in both the *static noise* and the *variable noise* experiment.

Static Noise In this test, we set $\sigma_{Noise} = 0.25$. The precision and recall for each different value of window size, N, were combined using the F_1 measure and the results are shown in figure 3.24. As expected, since the noise level is static, more accurate results are obtained using a greater number of samples (larger window size). The dashed line indicates the F_1 value if both signals were received at a single node and processed using the full MI thresholding algorithm.

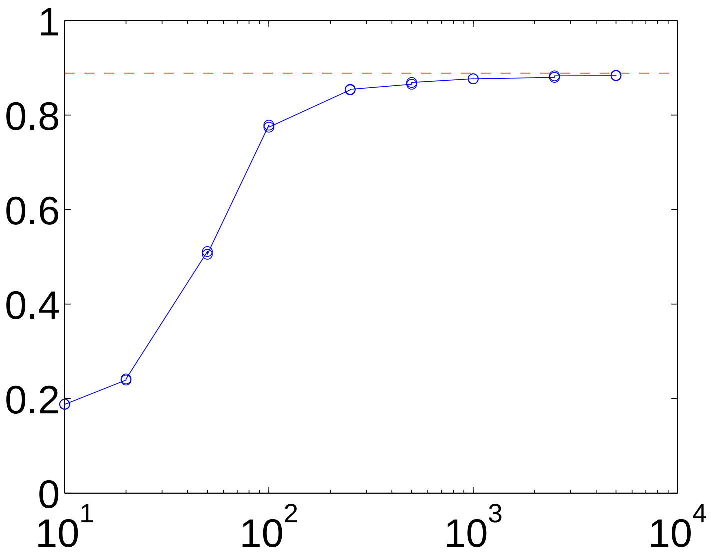


Figure 3.24: Performance of Online MI Thresholding on static noise, plotted using the F_1 measure. Dashed red line indicates the performance of MI thresholding.

Variable Noise In this test, the noise standard deviation would alternate every 5000 samples, from $\sigma_{Noise} = 0.25$ to $\sigma_{Noise} = 0.45$ and back. Again, the precision and recall for each different value of window size were combined using the F_1 measure and the results are shown in figure 3.25. Here, we have more interesting results where the best performing nodes use a window size of $N = 1000$. Smaller windows sizes are more easily affected by noise, and it is difficult for them to remain close to the peak of the

MI surface. In this case, larger window sizes, such as $N = 5000$ perform badly also. The case of $N = 5000$ specifically, as it gathers 5000 samples then chooses a threshold based on these samples. However, the noise then alternates to a different state, so the thresholds will always be suboptimal. Adapting the thresholds every 1000 samples even out-performs the MI thresholding algorithm's thresholds computed using all values of both signals.

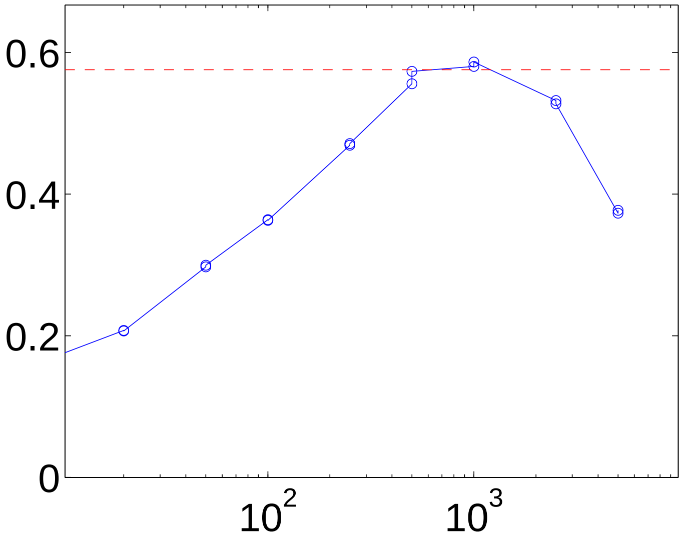


Figure 3.25: Performance of Online MI Thresholding on variable noise, plotted using the F_1 measure. Dashed red line indicates the performance of MI thresholding.

Discussion The online MI thresholding method described is essentially a *peak climbing* algorithm, and as such, it is important to initialise the procedure close to the slope of the peak. Using a small window size is susceptible to noise and can deviate the operating point away from the peak, hence requiring a number of adaptations to re-acquire the peak.

In our experiments, both nodes were set to update their thresholds simultaneously, but this might not be optimal for some MI surfaces. It might be better for nodes to alternate, with one node optimising while the other agrees to keep a fixed threshold. It is probably not optimal to perform *asynchronous threshold adaptation*, where neither node is aware that the other has adapted. In this case, one node (node A) will be gathering data on the other node (node B), but the properties of this data will change when node B alters threshold, and A will later attempt to adapt its threshold to maximise agreement with a mixture of old and new data.

One additional feature that might improve performance on real data would be to do selective updating of the thresholds. For example, if during the sample collection period no events occur, adapting the thresholds would dramatically reduce performance. A minimum quality score might be defined, such that no adaptation occurs unless this level of quality is reached. Alternatively, the adaptation might only occur if the current thresholds were deemed to be poor; such as if their quality score was deemed too low.

3.7 Summary and Discussion

In this chapter, mutual information thresholding was introduced and shown to explore a new paradigm in thresholding and fusion. The method was analysed theoretically using synthetic data and it exceeded the performance of the leading thresholding algorithms, along with providing a quality measure, indicating how well it performed. In the next chapter, the method of mutual information thresholding is further examined, looking at a variety of applications on real data.

In examining the assumptions underlying our method, it was shown that the method is quite flexible in some respects to the validity of these assumptions. Firstly, it was demonstrated that there is a significant disparity in the mutual information scores returned when there is common information present between sources, and when there is not. This indicates that failure of the method might be detected by examining the separation score. Secondly, the method has been shown to be quite robust to small mis-alignments when they are smaller than the typical object size. Finally, it was shown that correlated noise in the sources causes another peak to appear in the MI surface. In high noise conditions, this peak can be greater than the correct peak but the correct peaks can be found by examining the peak quality scores. An extended version of our algorithm, named EMIT, was described in section 3.4 to exploit this feature of the MI surface.

3.7.1 Fusion

A new paradigm There are generally two ways in which different data sources are combined. One approach is to create a new data representation, providing a better platform from which to perform analysis. Examples of this include linear combinations of the data, fusion using the max or min operator, or other non-linear combinations. The other common approach is that the analysis (such as thresholding) is performed separately on both sources of data and results are subsequently combined (using a binary operator, such as AND or OR, for example). Our novel method is a different paradigm for fusion. By performing the analysis on both sources of data simultaneously and using information from each source to assist the analysis of the other, we obtain results from two separate sources, but enhanced by each other.

Fusion After thresholding, one is left with two binary maps. If a single map is required, these results need to be fused in some way to obtain the final decision for each event.

One method is to use a binary operator, such as AND or OR, to combine the maps. An approach which is more robust against noise is to use the spatial information to determine the local support of each event. Support can be defined, for example, as the number of neighbouring events that have the same value as the central event. If the maps disagree on a detection result, the result with the greater support can be used. This is very effective at removing isolated noise. If the support values are equal, this could be an example of an object which is undetectable in one modality, such as a room-temperature bag using thermal infrared. Depending on the application, this disagreement could provide additional semantic knowledge. Using a 3×3 neighbourhood, this approach is similar to an OR fusion, followed by simple morphological erosion/dilation operations. Another approach, adopted in some early work [96] is to examine the support at a region level. Regions can be constructed by combining the maps using the binary OR operator. If a region has support in both modalities, it is retained, otherwise it is discarded as noise. Here *support* can be defined as a minimum fraction of its pixels belonging to foreground in each modality (*e.g.* 10%).

3.7.2 Relationship to other agreement measures

Besides mutual information, there are other measures to compute similarity and agreement between signals. One such measure is Kendall's τ [67][109]. The Kendall τ rank correlation coefficient (or simply the Kendall tau coefficient or Kendall's τ) is used to measure the degree of correspondence between two rankings and to assess the significance of this correspondence.

The integral-image based approach developed in this chapter allows the efficient counting of occurrence of pairs of binary values between two thresholded data sources (see the $C_{u,v}$ array in subsection 3.2.1). For a given pair of thresholds, the number of occurrences of each of the four binary pairings, $\{00, 01, 10, 11\}$, can be counted. While these counts were used, in this chapter, to compute the mutual information between the thresholded sources, they might equally be used to compute many other measures of similarity.

Some directions for future work would be in examining the benefits of using another binary-signal agreement-measure besides mutual information. Examples of potential methods to test include Spearman's rank correlation coefficient and Pearson product-moment correlation coefficient, as well as Kendall's τ.

In terms of the binary pairings counts and the total number of samples, N, Kendall's τ can be expressed as:

$$\tau = \frac{C_{x,y}(0,0)C_{x,y}(1,1) - C_{x,y}(0,1)C_{x,y}(1,0)}{\sqrt{C_x(0)C_y(0)C_x(1)C_y(1)}} \tag{3.40}$$

Another potentially useful measure is obtained by taking the numerator of Kendall's τ:

$$M_1 = \frac{1}{N}(C_{x,y}(0,0)C_{x,y}(1,1) - C_{x,y}(0,1)C_{x,y}(1,0)) \tag{3.41}$$

or simply

$$M_2 = \frac{1}{N}C_{x,y}(0,0)(C_{x,y}(1,1). \tag{3.42}$$

Some other combinations that stress agreement are

$$M_3 = \frac{C_{x,y}(0,0)C_{x,y}(1,1)}{\sqrt{C_x(0)C_y(0)C_x(1)C_y(1)}} \tag{3.43}$$

$$M_4 = \frac{C_{x,y}(0,0)C_{x,y}(1,1)}{1 + C_{x,y}(0,1)C_{x,y}(1,0)} \tag{3.44}$$

To briefly investigate the potential of these four measures, as well as Kendall's τ and to compare them to the use of mutual information, a preliminary experiment is described. In figure 3.26, examples of synthetically-generated difference images are shown. Using these two images as data sources, the corresponding surface for each of the six measures was computed, shown in figure 3.27, along with the corresponding thresholded images. In this test, measure M_1 and M_3 seem to perform acceptably, as well as the MI measure and Kendall's τ. Unfortunately, further tests on real data revealed that M_1 and M_3 were not robust measures of agreement.

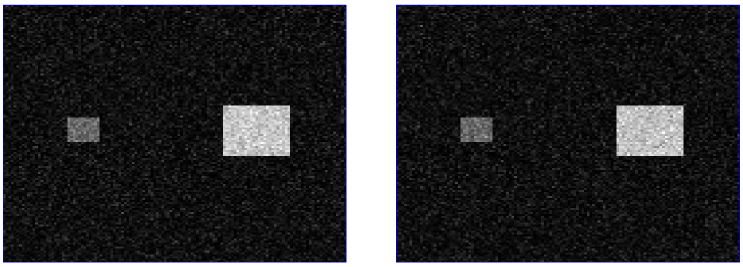


Figure 3.26: Synthetic difference images used to investigate the usefulness of other agreement measures.

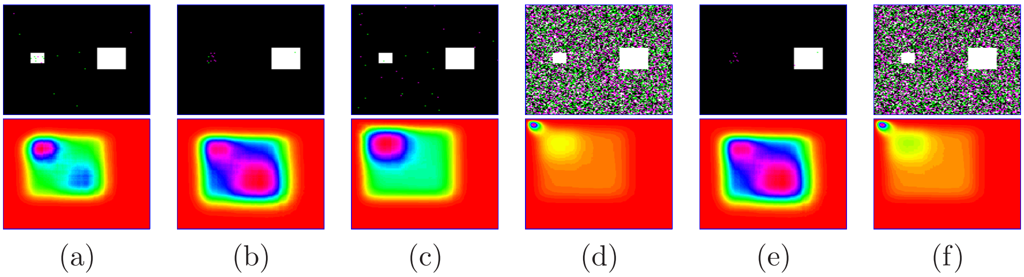


Figure 3.27: Measure surfaces for the difference images for all six measures and their corresponding thresholded images: (a)MI, (b)Kendall's τ, (c)M_1, (d)M_2, (e)M_3 and (f)M_4

3.7.3 Future work

Our thresholding method works on aligned data so can be used for local, as well as global thresholding. It can also be used to threshold space-time slices, such as groups of video frames. In these scenarios, the window size is an important parameter: too small and it may be sensitive to noise, too large and there is a chance the signal properties have changed and a global threshold would not be appropriate. This was shown clearly in the experiments on online MI thresholding. Investigating how the window sizes should be set automatically is an interesting area of further work.

Currently, the method does not consider spatial information or the proximity of pixels when choosing the thresholds. Incorporating this information into the method is another avenue of research to consider. For example, the two parameters (low and high thresholds) for hysteresis segmentation could be selected by maximising the MI between the resulting segmentation and another source of data. The experiment on selecting the optimal smoothing σ value touched on the use of spatial information to improve performance. Spatial information might also be included by counting binary pairings of larger neighbourhoods.

It is still unclear exactly what kinds of data sources are appropriate for use with MI thresholding. The gradient of a signal would seem to be strongly correlated with the original signal but performed excellently as a second data sources in the synthetic data experiments. In terms of real world visual data, the separate colour channels, such as $\{R,G,B\}$ or $\{H,S,V\}$ might be used as inputs to the proposed method, but whether they are too correlated to perform well, since they are derived from the same sensor, is another issue worth investigating and in fact is explored in the next chapter.

Finally, using this method on three or more sources of data is another area for future investigation. The quality measure developed gives a estimate of the reliability of the results and hence, this might be used to make a system more robust against the failure of one or more components, if it can quickly detect unreliability between the data sources. The combination of three or more sources provides many interesting challenges, such as whether they should all be combined simultaneously, or whether a pair-wise combination, using the quality values returned, provides better performance. To use the framework of [71], the sources could first be used pairwise and all thresholds for each source are saved where a peak of the MI surface occurs. Then to fuse three sources, triplets of thresholds are evaluated. All thresholds for each source that belong to a peak of the hypersurface are saved and the process continues, adding more sources. Another way would be to combine all sources simultaneously. An agreement measure, such as mutual information, might be defined for three or more signals and some optimisation procedure could be used to maximise this agreement measure between the multiple sources. For any multi-source method, care must be taken to robustify the approach, so that the addition of a single poor-quality source does not adversely affect the results.

In the next chapter, numerous applications that use MI thresholding and related techniques on real-world data are examined.

Chapter 4

Applications of Maximal Agreement Thresholding

4.1 Introduction

In the previous chapter, using synthetic data, it was shown that choosing thresholds to maximise agreement between two data sources outperforms traditional automatic threshold-selection algorithms. Mutual information was used to measure agreement between binary signals in the synthetic tests, but other agreement measures are also possible as was discussed.

In this chapter, real-data applications of agreement-based thresholding are investigated, exploring the use of both mutual information and the Kendall's τ agreement measure. These applications include the automatic learning of parameters for shadow detection, online dynamic skin model learning, foreground detection in thermo-visual data, feature selection for event detection in audio-visual surveillance scenarios and person detection in thermal imagery.

The applications are divided into two classes: Applications using weakly independent sources and applications using strongly independent sources. The distinction is the degree to which the data sources used can be considered independent. Specifically, if the sources come from separate sensors of different modalities, then they are considered strongly independent. In each class, 3 applications are presented.

In class A, using weakly independent sources, the detection of people in thermal imagery from the OTCBVS dataset [29] is examined. Using a silhouette-based and contour-based template in the context of MI thresholding, people are accurately detected. Next, extensive tests on foreground detection are performed using visible spectrum imagery from many publicly available datasets [1, 58, 108, 143]. The visible data is split into data sources, such as the individual components of R, G, B and H, S, V, and all pairs of sources are combined using MI thresholding and evaluated using ground-truthed foreground. Finally, the algorithm of MI thresholding is extended to handle *bounded ranges* instead of simple thresholds, and this allows the automatic selection of parameters for the detection of shadow pixels.

In class B, the benefits of MI thresholding for detecting foreground by combining thermal infrared and visible spectrum video are investigated. Next, visual information is combined with non-visual audio data to detect people in a surveillance context. It is shown that MI thresholding can be used to select the best features for detection by choosing those features that produce strong agreement between modalities. Finally, the bounded-range algorithm, introduced for shadow detection, is shown to efficiently choose parameters for skin detection in thermo-visual data and outperforms non-adaptive skin classifiers.

4.2 Applications using weakly independent sources

4.2.1 Foreground detection

The use of MI thresholding for foreground detection on real data is now investigated. The data used comes from publicly available datasets, as well as self-captured video sequences. Manually annotated ground truth is used to verify the accuracy of the resulting foreground. Since the goal is to evaluate the thresholding component of a foreground-detection system, the background model used is as simplistic as possible. Specifically, a background image is used as the model, which is computed by taking the median of a large number of frames. Frames are chosen with no moving objects present if possible, to attain the best possible background image.

As the method requires two data sources as input, the image data is split into multiple parts, and each pair of sources is evaluated. The initial data is an RGB *current image*, I_{curr}, and an RGB *background image*, I_{BG}. The first three sources are obtained by subtracting each colour band separately, giving a red-difference, green-difference and blue-difference image. Next, both I_{curr} and I_{BG} are transformed to the HSV colour-space and then subtracted. This gives 3 additional sources: difference images from H(hue), S(Saturation) and V(Value/Luminance). For all difference images, the absolute value is taken, since it is only the magnitude of change that is important, not its sign.

In the previous chapter, the gradient of a data source was shown to provide an excellent second source of data for the MI thresholding method. In order to use gradient here, the gradient magnitude of both I_{curr} and I_{BG} are computed, using the Sobel operator, and they are subtracted to obtain the gradient-difference for R, G and B. The same operation is performed on I_{curr}^{HSV} and I_{BG}^{HSV} to obtain the gradient-difference for H, S and V. This provides a total of 12 sources of data: absolute difference of pixel values of R,G,B,H,S and V, and the gradient difference of R,G,B,H,S and V. We will refer to the absolute difference images as $\{R, G, B, H, S, V\}$, and the gradient difference images as $\{r, g, b, h, s, v\}$.

The experiment was conducted as follows. A total of 492 images were gathered from 16 sequences. Each image was split into 12 data sources, as described above, and each of the sources was thresholded using the same four methods used for the synthetic data experiments in the previous chapter, namely: the Kapur, Otsu, Kittler and Rosin methods. This gives 48 binary masks for this image. Next, every pair of sources from the 12 are used as input to the MI thresholding algorithm. We differentiate between using data sources $\{R, h\}$ and using $\{h, R\}$, since the output of the former is the binary mask for R and the latter outputs a binary mask for h. The first source will be referred to as the *primary* source, as it is this image that will be thresholded and evaluated; the secondary source merely helps to find the threshold. This stage gives 132 binary masks, since the 12 pairs that have identical primary and secondary sources are not used, as this would simply use a median threshold. Additionally, the same tests are performed using the EMIT algorithm, to counter the potential effects of having multiple peaks in the MI surface. All binary masks were evaluated using the five evaluation measures described in the previous chapter. For each image, a total of 312 (132+132+48) binary masks are generated and evaluated. Images from the datasets, along with ground-truth examples, are shown in figures 4.1 and 4.2. Table 4.1 lists the datasets from which the data for these experiments was obtained. A σ value of 1.5 was used for the EMIT algorithm which was hand-tuned on a small number of images not used in testing.

EMIT vs. MI thresholding Since the data sources used in this experiment may have correlated noise, the EMIT algorithm's heuristic for peak selection is expected to give better performance than selecting the maximum peak in the MI surface. This was found to be the case. Table 4.2 compares the EMIT algorithm to the standard peak selection MI thresholding. In all five performance measures, the EMIT algorithm out-performs the standard algorithm.

In table 4.3, the top five MI-based thresholding methods are given for each measure.

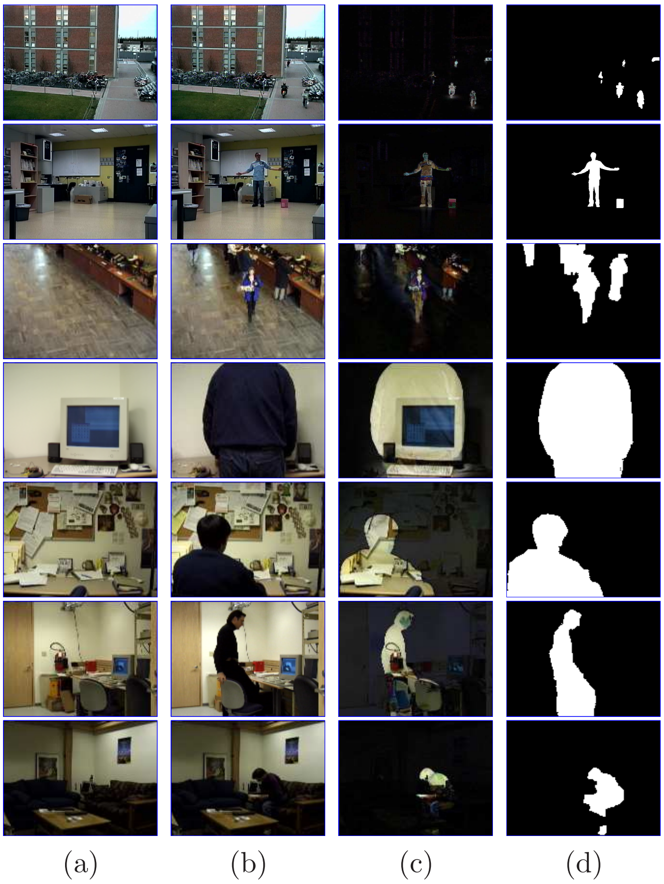


Figure 4.1: Sample images from the testing datasets: (a) Background image, (b) Current image, (c) RGB difference image and (d) Ground-truth.

Table 4.1: Datasets used in this experiment

Name	Source	Ground-truthed images
DCU	Own data	3
Terrascope	[58]	4
Wallflower	[143]	7
Laboratory	[108]	7
Highway I	[108]	6
Highway II	[108]	5
Campus	[108]	6
Intelligent Room	[108]	112
VSSN	[1]	342

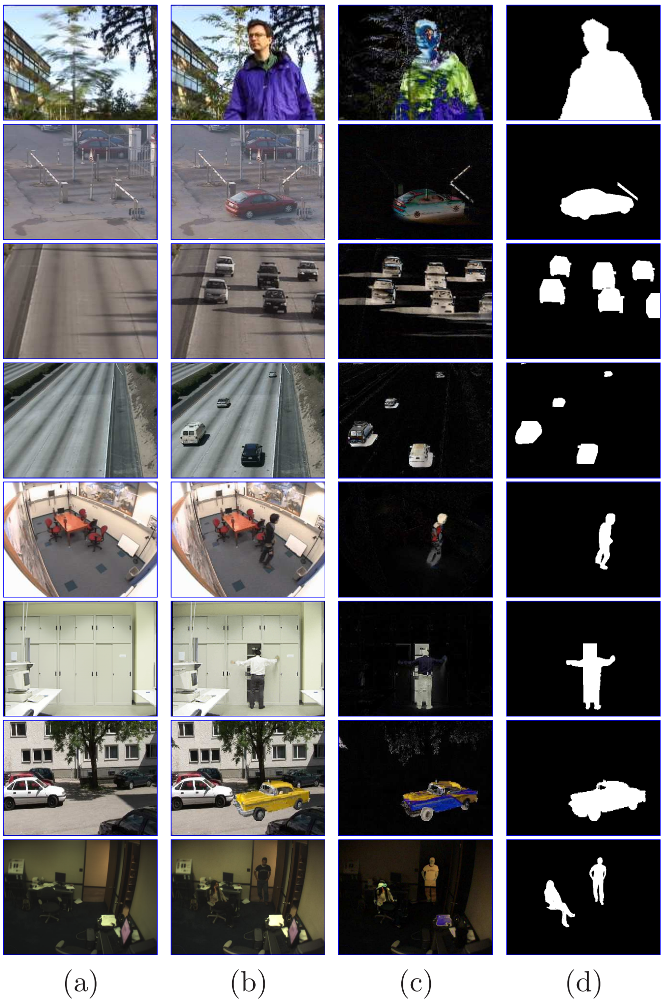


Figure 4.2: Sample images from the testing datasets: (a) Background image, (b) Current image, (c) RGB difference image and (d) Ground-truth.

Table 4.2: Comparing the EMIT algorithm to standard MI thresholding.

Measure	EMIT worse		Equal		EMIT better	
	frames	%	frames	%	frames	%
M_1	5799	8.93	8450	13.01	50695	78.06
M_2	14719	22.66	8439	12.99	41786	64.34
M_3	12602	19.40	8439	12.99	43903	67.60
M_4	24872	38.30	8439	12.99	31633	48.71
M_5	14719	22.66	8439	12.99	41786	64.34

For example, in column 1, row 1, $E_{V,H}(0.563)$ refers to the EMIT algorithm using the V and H difference images as the primary and secondard source, respectively. The value 0.563 is the average score of the M_1 (cost) measure over the entire series of images. Similarly, in column 1, row 3, $M_{G,H}(0.769)$ refers to the MI thresholding algorithm using the G and H difference images as the primary and secondard source, respectively, whilst the value 0.769 is the average score of the M_3 (equal error) measure over all tested images. The EMIT algorithm performs very well overall using four of the performance measures. Interestingly, the EMIT algorithm is not the top performer in measure M_3 (the equal error criterion), where standard MI thresholding does best. This suggests that the equal error measure weights precision and recall differently from the other measures, and this can be seen in the visualisations of the measures in the previous chapter. Also of interest, the secondary sources for the standard MI thresholding methods are all Hue-based, suggesting that the hue difference is the most independent of the sources, since if the noise was heavily correlated the standard MI thresholding method would select the noise peak.

Table 4.3: Top 5 performing MI-based thresholding methods according to each measure: E refers to the EMIT algorithm and M refers to standard MI thresholding. Average measure values are given for each method.

Measure	#1	#2	#3	#4	#5
M_1	$E_{V,H}(0.563)$	$E_{G,H}(0.560)$	$E_{R,H}(0.555)$	$E_{G,s}(0.555)$	$E_{B,H}(0.553)$
M_2	$E_{V,g}(0.509)$	$E_{G,b}(0.508)$	$E_{G,r}(0.507)$	$E_{G,g}(0.505)$	$E_{G,v}(0.505)$
M_3	$M_{G,H}(0.769)$	$M_{V,H}(0.762)$	$M_{R,H}(0.756)$	$M_{B,h}(0.751)$	$M_{g,H}(0.747)$
M_4	$E_{V,g}(0.771)$	$E_{G,b}(0.771)$	$E_{V,b}(0.768)$	$E_{G,g}(0.768)$	$E_{R,g}(0.767)$
M_5	$E_{G,b}(0.384)$	$E_{G,r}(0.383)$	$E_{G,v}(0.381)$	$E_{G,g}(0.379)$	$E_{V,g}(0.379)$

Best sources for foreground detection Using the EMIT algorithm for foreground detection, the choice of the best sources to use is investigated. Due to the large volume of tests in this section, it is unfeasible to give tables of figures relating to the scores of all pairs of sources for all evaluation measures. However, figure 4.3 attempts to visually illustrate the overall performance of all pairs, using a brighter grid square for a higher score. In figure 4.3(f), the results of all measures are combined by first computing the rank of each pair, according to each measure, and then taking the median rank from the 5 measures. A brightness is then assigned to each rank, from 1 (white) to 132 (black). On the y-axis of each subfigure is the *primary* source and the x-axis is the secondary source. As previously mentioned, *primary* refers to the fact that the scores comes from the result of applying the chosen threshold to this source, with the secondary source merely helping to find that threshold.

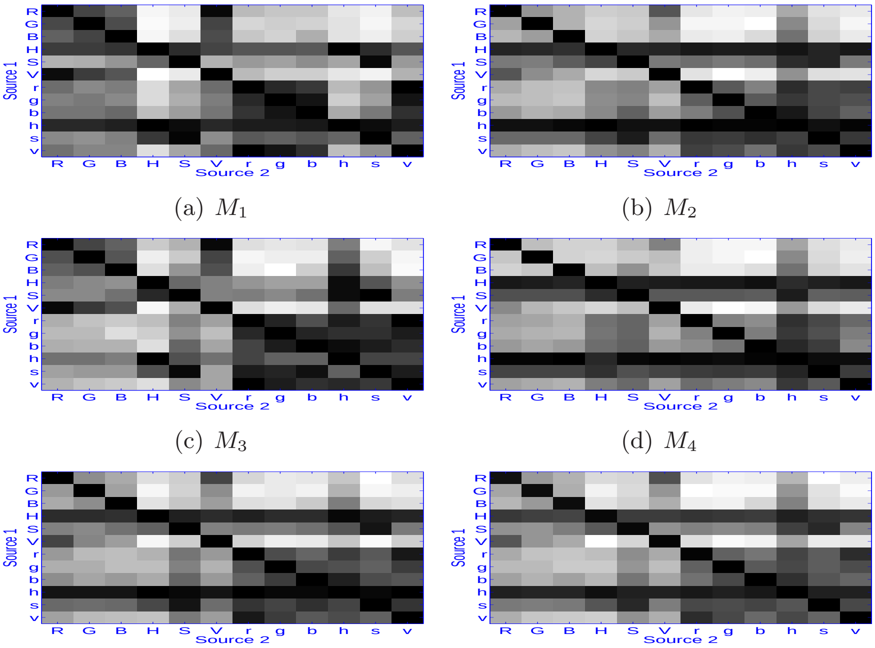
The top 10 performing pairs are shown in table 4.4. The sources are sorted by the median of their five ranks according to all five measures. As well as using the full set of 492 images, a smaller subset of 38 images was also used and the results shown in the three rightmost columns in the table. This subset was selected by removing the *Intelligent Room* and *VSSN* sequences, since they are the two largest sequences and heavily bias results due to their relatively large number of images.

It is clear from both sets of results that the most useful primary sources are $\{R, G, B, V\}$. Edge-based sources do not appear as good primary sources, nor do the hue or saturation differences. This is to be expected as gradient-based sources primarily emphasise the foreground on object boundaries only, and hue and saturation difference are prone to high noise. The gradient-based images provide excellent secondary sources, as evident in the results on the full set. In the partial set, sources H and s appear as the most useful secondary sources, but interestingly so do R, G and B. This suggests that the EMIT algorithm is able to avoid the strong correlation in the noise between these sources and find good thresholds.

Table 4.4: Top performing pairs of sources for the EMIT algorithm: *Full Set* refers to all 492 images and *Partial Set* refers to the full set with the VSSN and Intelligent Room sequences removed (38 images).

Full Set			Partial Set		
Primary source	Secondary source	Median rank	Primary source	Secondary source	Median rank
G	b	2	R	s	1
G	r	4	V	s	2
G	g	4	G	B	3
V	g	5	G	H	5
G	v	5	G	s	5
R	g	7	R	B	6
V	b	7	B	G	7
B	g	8	B	R	8
R	b	9	B	H	11
B	v	11	V	H	11

Results Table 4.5 shows the top 5 best performing methods overall, according to each measure. The method names are abbreviated for clarity. For example, Kp_R refers



(a) M_1

(b) M_2

(c) M_3

(d) M_4

(e) M_5

(f) Median rank

Figure 4.3: Evaluation scores of fusing all 12 pairs of sources: An illustration of the merits of fusing various combinations of sources, according to the 5 different measures. Brighter points represent better scores. Shown in (f) is a combination of all measures using each pair's median rank.

to Kapur's method using the R difference image. The values shown in brackets are the average value of each performance measure. Clearly, the Kapur method dominates the table, with the Otsu method following behind (e.g. Ot_V refers to Otsu's method using the V difference image). By combining the measures, taking the median rank of each method, the top 20 methods are shown in table 4.6. Again, the results of two sets are shown: the full set of 492 images, and the partial set of 38 images, obtained by excluding the Intelligent Room and VSSN sequences.

The MI-based algorithms are adversely affected by the noise in the Intelligent room sequence. Very low noise is present in the corners of the image, due to the pixels being overexposed, but the rest of the image has high noise. This corresponds to the synthetic data of the previous chapter where the noise peak has grown so large it swamps the true peak. Figure 4.4 shows a failure of the EMIT algorithm on an image from the Intelligent Room sequence.

On the other hand, table 4.7 shows some positive results in favour of the MI-based methods. For every image (of the 492) and each of the five performance measures, the best ranking method was established. The table indicates how frequently the best method was one of the 132 MI-based methods, instead of one of the other 48 standard methods. On average, the top performing method is based on MI in 75.57% of the images. Since there are more MI-based methods being evaluated, they have an advantage, but this is still an interesting result. When we further examine this finding, and compute how often each method is ranked first for a tested image, the $M_{V,H}$ method is the top performer more often than any other method, including the Kapur methods. Table 4.8 illustrates this fact. It shows how often a method was ranked first of all methods for an image/measure test. A total of 492 images were used, with 5 performance measures, giving a total of $2,460$ image/measure tests. The MI thresholding method using V as a primary source and H as a secondary source ($M_{V,H}$) was ranked first in 226 tests, more than any other method. However, due to the correlated noise in the Intelligent Room sequence, its average was severly affected. However, these results show that, given favourable conditions, the MI-based methods can sometimes outperform traditional approaches.

Quality-performance correlation In trying to automatically ascertain whether the EMIT algorithm was successful, the correlation between the quality measure returned and the performance measures was investigated. It was found that a weak correlation existed between the quality and most of the performance indicators.

Table 4.9 shows Pearson's correlation coefficient relating quality to the 5 perfor-

Table 4.5: Top 5 performing methods overall, according to each measure: *Kp* is Kapur and *Ot* is Otsu's method, average measure values shown.

Measure	#1	#2	#3	#4	#5
M_1	$Kp_R(0.595)$	$Kp_V(0.593)$	$Kp_G(0.590)$	$Kp_B(0.588)$	$Ot_V(0.566)$
M_2	$Kp_R(0.621)$	$Kp_V(0.617)$	$Kp_G(0.613)$	$Kp_B(0.609)$	$Ot_G(0.589)$
M_3	$Kp_B(0.836)$	$Kp_R(0.829)$	$Kp_V(0.826)$	$Kp_G(0.824)$	$Ot_B(0.786)$
M_4	$Kp_R(0.833)$	$Kp_V(0.830)$	$Kp_G(0.828)$	$Ot_V(0.821)$	$Ot_G(0.820)$
M_5	$Kp_R(0.487)$	$Kp_V(0.484)$	$Kp_G(0.479)$	$Kp_B(0.477)$	$Ot_B(0.450)$

Table 4.6: Top performing methods overall sorted by median rank of all 5 measures: Kapur (Kp), Otsu (Ot), Rosin (Rs), EMIT (E) and MI thresholding (M).

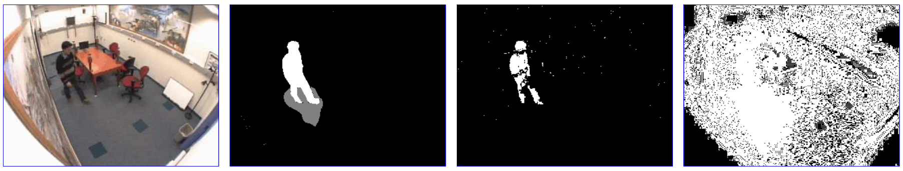
	Full Set		Partial Set	
#	Method source(s)	Median rank	Method and source(s)	Median rank
1	Kp_R	1	Kp_R	1
2	Kp_V	2	Ot_R	4
3	Kp_G	3	Ot_G	4
4	Kp_B	4	Kp_B	6
5	Ot_G	6	Ot_B	7
6	Ot_V	6	Ot_V	7
7	Ot_R	7	$E_{R,s}$	7
8	Ot_B	7	Rs_G	8
9	$E_{G,b}$	10	Rs_R	9
10	$E_{G,g}$	13	$M_{G,s}$	10
11	$E_{V,g}$	13	Rs_B	11
12	$E_{R,g}$	15	Kp_G	12
13	$E_{V,b}$	15	Rs_V	12
14	$E_{G,r}$	16	$M_{G,S}$	14
15	$E_{R,b}$	17	$E_{V,s}$	15
16	$E_{G,v}$	20	Kp_V	17
17	$E_{R,v}$	21	$M_{G,H}$	18
18	$E_{B,g}$	21	$E_{G,B}$	21
19	$E_{R,r}$	22	$M_{V,s}$	21
20	$E_{R,s}$	23	$M_{V,H}$	23

Table 4.7: Number of images in which a MI-based method out-performs other algorithms

Measure	# frames	% total
M_1	380	77.24%
M_2	365	74.19%
M_3	416	84.55%
M_4	333	67.68%
M_5	365	74.19%

Table 4.8: Ranked list of methods in order of the number of times they were ranked first for a frame/measure evaluation. Of a total of $2,460$ (492×5) frame/measure tests, the MI thresholding algorithm using the V and H sources was ranked first in 226 of them.

Method type	# Top scoring images	% Top scoring images
$M_{V,H}$	226	9.19
$M_{G,H}$	138	5.61
$M_{V,h}$	138	5.61
Kp_G	130	5.28
Kp_R	115	4.67
$M_{R,H}$	109	4.43
Kp_V	96	3.90
Kp_B	88	3.58
$M_{R,h}$	86	3.50
$E_{V,H}$	58	2.36
$E_{G,r}$	43	1.75
$E_{G,H}$	42	1.71
$E_{R,H}$	37	1.50
Ot_V	31	1.26
$E_{G,g}$	30	1.22
$M_{G,S}$	28	1.14
$M_{R,s}$	27	1.10
$E_{s,R}$	26	1.06
$M_{V,s}$	25	1.02
$M_{B,h}$	24	0.98



(a) Image (b) Ground truth (c) Kp_R (d) $E_{G,b}$

Figure 4.4: Example failure of the EMIT algorithm: due to the very low noise at the boundary, and the high noise in the centre, the sources have strongly correlated noise, causing the method to fail.

mance measures. The top five rows are the values for some of the highest performing sources using the median rank metric of table 4.4. The last line shows the values for all 132 pairs of sources over the 492 images.

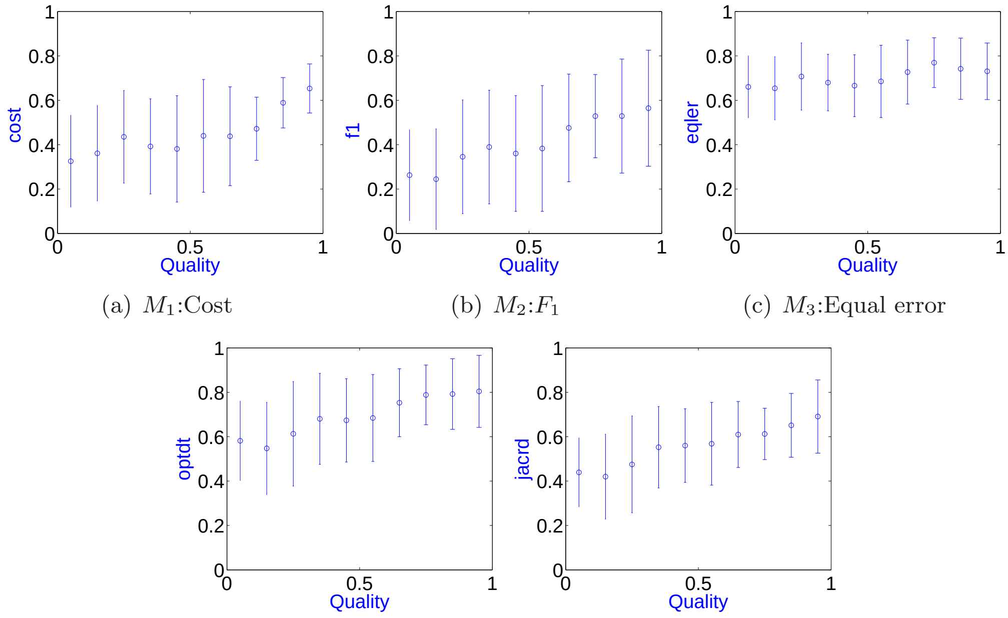
Figure 4.5 shows some graphs illustrating these weak correlations. The mean value of the performance measures tend upwards as the quality score increases. However, the variance is very large in comparison, so any kind of practical exploitation of this correlation would need to use a large number of tests in order to use quality as a reliable measure of performance.

Table 4.9: Correlation coefficients between quality and performance measures

Sources	M_1	M_2	M_3	M_4	M_5
G, b	0.494342	0.295082	0.415945	0.178785	0.205603
V, g	0.512387	0.295198	0.427084	0.183760	0.215294
G, g	0.519210	0.283399	0.425815	0.171710	0.204506
G, r	0.509556	0.264159	0.396021	0.136299	0.163691
R, g	0.514160	0.300826	0.421541	0.177638	0.205225
ALL	0.209598	0.262382	0.094052	0.256596	0.323423

Discussion From the experiments, it has been shown that the EMIT version of the algorithm performs much better than the standard maximum peak selection method, in the presence of correlated noise in the sources. The Kapur method of thresholding was shown to perform very well across different datasets. It consistently chose a higher threshold than most other methods and avoided noise, giving high precision. There were cases however when the EMIT algorithm performed better. Figure 4.6 shows a number of such examples.

An interesting result of these tests was the discovery that the $M_{V,H}$ method was ranked *first* more often than any other method for a given image from the test collection.



(a) M_1:Cost

(b) M_2:F_1

(c) M_3:Equal error

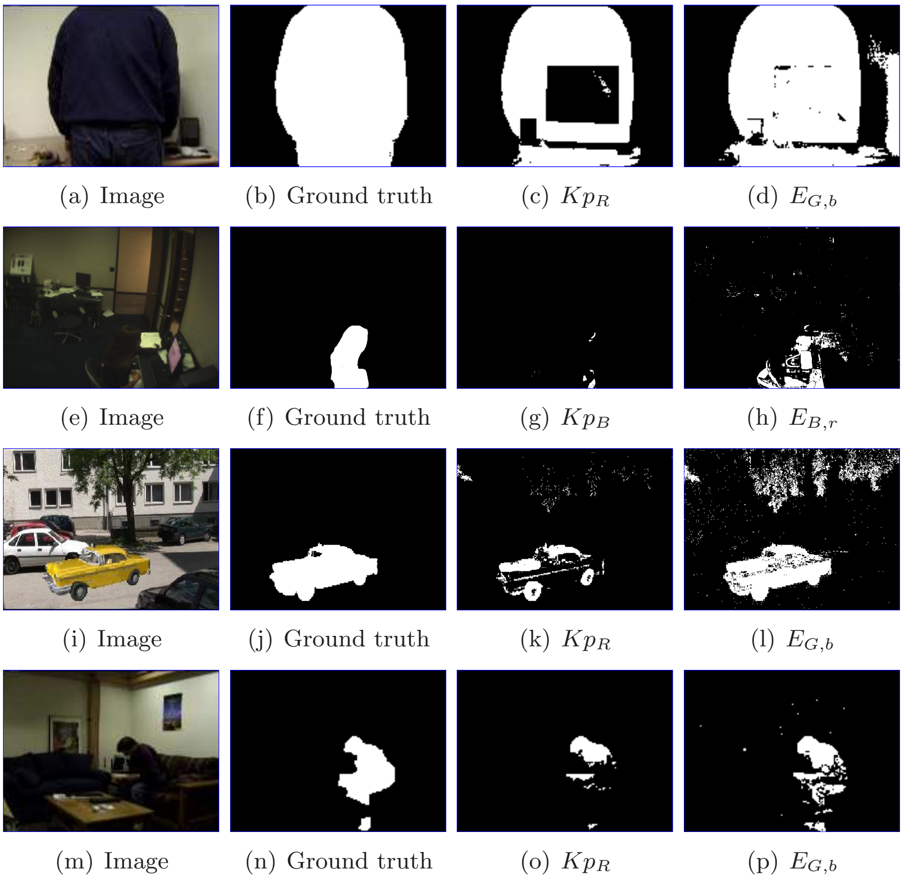
(d) M_4:Optimal Dist

(e) M_5:Jaccard

Figure 4.5: Graphs illustrating the weak correlation between quality and performance. Error bars indicate one standard deviation.

Its poor performance in the Intelligent-Room images, resulting from correlated noise due to overexposure of some pixels, can be corrected by rerunning the method on those pixels that were not classed as background by both sources. This approach gives results comparable to Kapur on these images, but it is left to future work to determine when to automatically exclude pixels in this way and thereby avoid this problem. The results in table 4.7, showing that MI-based methods are very often the top-performers, give hope to the use of these methods in foreground detection, despite the obvious correlated nature of the sources. The challenge is therefore to create automatic methods for choosing the best performing sources. For this goal, the correlation between quality and performance might be a profitable line of inquiry.

This experiment focussed only on using a *primary* source and selecting the threshold for this source by using its relationship with a *secondary* source. An interesing area of future work could examine the various ways in which the binary images produced for both sources could be fused for improved foreground detection. Some preliminary work in this direction is reported in a later section which focusses on skin pixel detection using thermal infrared and colour visual information.



(a) Image (b) Ground truth (c) Kp_R (d) $E_{G,b}$

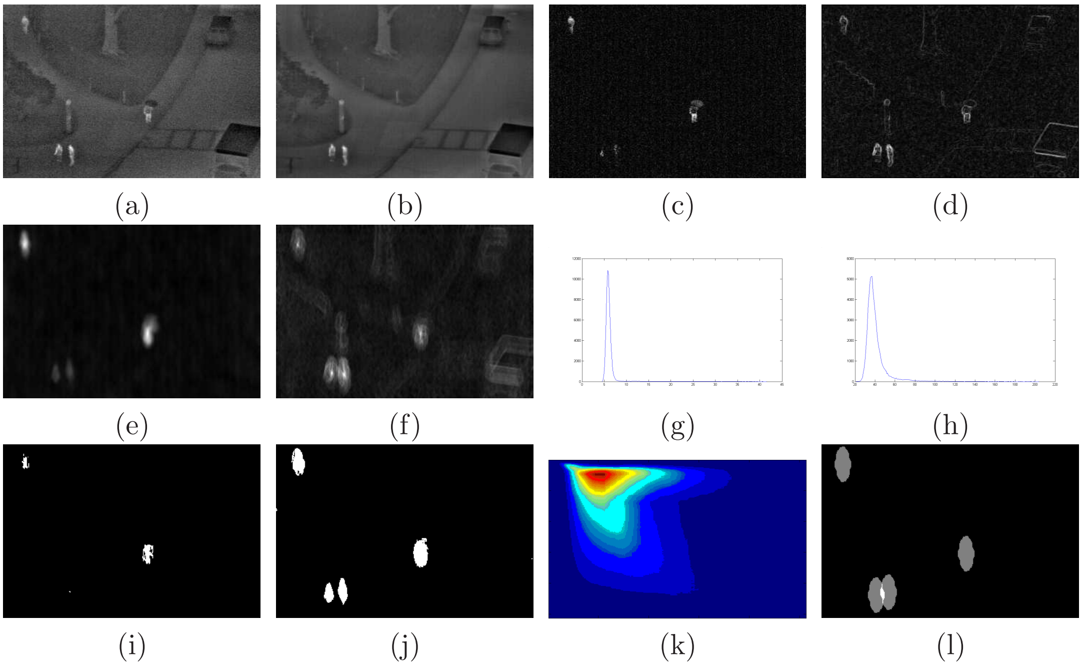
(e) Image (f) Ground truth (g) Kp_B (h) $E_{B,r}$

(i) Image (j) Ground truth (k) Kp_R (l) $E_{G,b}$

(m) Image (n) Ground truth (o) Kp_R (p) $E_{G,b}$

Figure 4.6: Example results of foreground detection

4.2.2 Person detection



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 4.7: Person-detection example: (a) Current image, (b) Background image, (c) Background difference, (d) Image edges, (e) Silhouette detection map, (f) Contour detection map, (g) Histogram of (e), (h) Histogram of (f), (i) Kapur thresholded result, (j) the proposed method, (k) Mutual information surface, (l) Detected People

To further test the MI thresholding algorithm, it was incorporated into a person detection system and used the OSU Thermal Pedestrian Database from the OTCBVS Benchmark Dataset [29] to evaluate performance. The database contains sequences of pedestrian images taken with a thermal infrared camera in different seasons and features wide variation in the appearance of pedestrians. Since the goal is to evaluate the thresholding component of the system, the other components are chosen to be as simplistic as possible.

The system worked as follows. First, the median background image was computed for each sequence. Then for each image, two detection signals were created: one based on pedestrian contour and the other based on pedestrian silhouette. The contour detection map was obtained by convolving a pedestrian contour template, shown in figure 4.9(a), with the Sobel edges of the image. The silhouette detection map was obtained by convolving a pedestrian silhouette template, shown in figure 4.9(b), with the absolute difference image between the current image and the background image.

The pedestrian silhouette template is simply a representative person shape taken from the data and binarised. The contour template is simply its boundary pixels. The assumption that all people appear at a similar scale can be made, since the camera is mounted at a high angle. Thresholds for these maps were then obtained using the EMIT algorithm with $\sigma = 1.5$ (the same σ value as used in the foreground detection application). Pedestrian regions were determined as all pixels that had above threshold values in both maps (binary AND fusion). Next, each local maxima in the contour detection map within these regions was paired with the closest local maxima in the silhouette detection map within these regions. Maxima in the silhouette detection map were then paired with the closest maxima in the contour detection map. Person candidates corresponded to each pair of maxima, from the two separate maps, that were both paired to each other (i.e. they were both closest to each other). Candidates were then evaluated according to the minimum description length principle, in respect to how much of the pedestrian regions they could explain. A pedestrian candidate template (shown in figure 4.10) is used to evaluate the fitness of each candidate by calculating the maximum number of pedestrian-region pixels it overlaps with, when centred on either maxima of the candidate. The best candidate is considered a 'true' person and the pedestrian region pixels it overlaps are removed. This process continues until there are no remaining candidates, or no candidate can explain more than a predefined number of pixels. This lower limit was set at 15% of the template size. The EMIT algorithm was modified slightly to handle the appearance of an extra (non-noise) peak in the MI surface; an example of which is shown in figure 4.8(g). Instead of using Rosin thresholding, only peaks that were greater in height than 25% of the highest peak were considered. In figure 4.8(f), a very small peak is seen in the MI surface at position $(154, 154)$ of height 0.0043. This peak, although it is very small, is not eliminated by the Rosin threshold in the first stage of the EMIT algorithm, and actually returns the highest quality score of 0.6025. Using its thresholds results in figure 4.8(h), missing most of the people. The modification of the first stage of the EMIT algorithm was necessary to remove peaks such as this. The correct peak has a MI value of 0.0412 and a quality of 0.2634.

The proposed algorithm was compared to the Kapur method, which has shown excellent performance in the foreground detection tests and also in previous work [118]. Since there are two sources of information (the silhouette and contour detection maps), either source could be used, or Kapur thresholding could be applied to both maps and their binary results fused in some way. It was found that the use of the silhouette map performed best, when compared to the contour map or to a binary AND/OR fusion of both maps. The results of running the system on the OTCBVS person database are shown in table 4.10. The rest of the system used was identical to the EMIT system, only the thresholding component had changed.

An example of person detection is shown in figure 4.7. The images are from sequence 8, where Kapur performs poorly, with a very low person recall. In this difficult example, the two people, in the bottom left of the image, have been standing in the same spot for the entire sequence, so have been included in the background image. However, the two people do not remain completely stationary and therefore their slight motion leaves an impression on the difference image and hence, on the silhouette based detector map. The MI thresholding method causes the silhouette threshold to drop so that it agrees with the strong detection in the contour-based detection map. Kapur, on the other hand, sets the two thresholds independently and therefore fails to detect all the people.

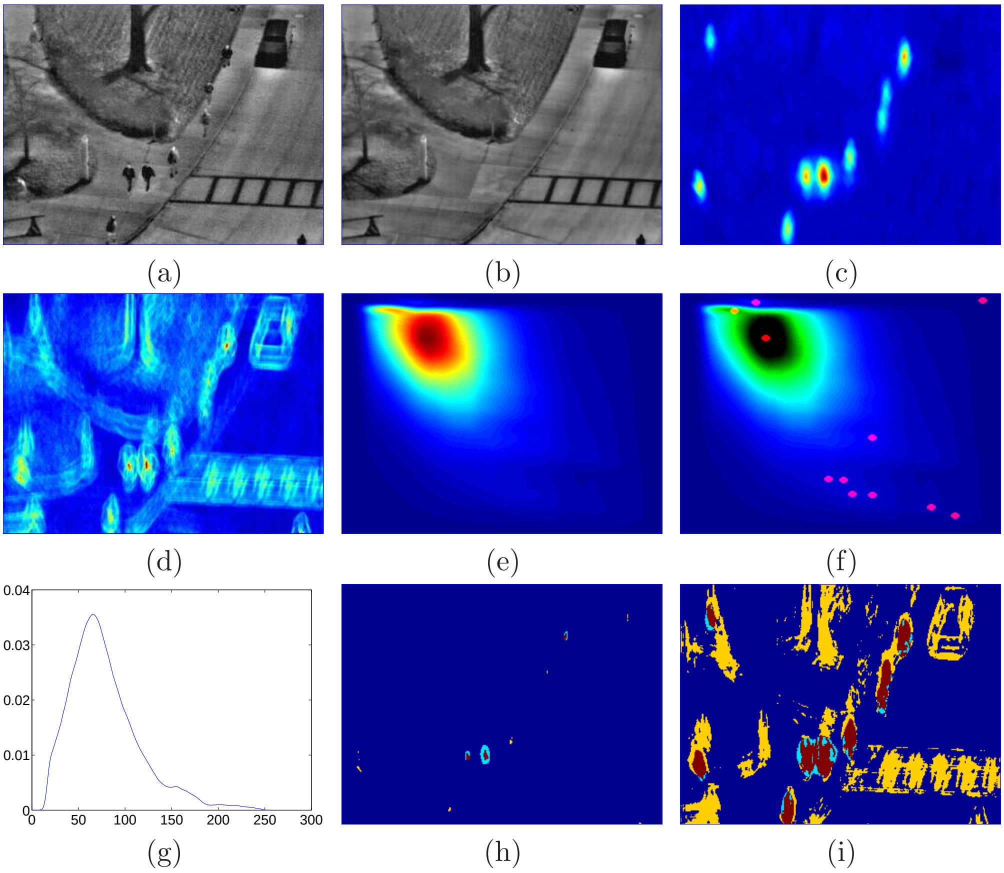
For the results shown in table 4.10 a value of 0.15 was used as the lower limit (minimum fraction) of the candidate mask that a valid person should take into account. To ensure that this ad-hoc figure did not bias results, the experiment was run multiple times with difference values of this limit. Figure 4.11 shows the relative overall performance of both systems, measured using the F_1 measure. Figure 4.12 shows the same results plotted on a precision-recall curve for both systems. The plots show that the EMIT system clearly outperforms the Kapur-based system.

Experiments in the previous chapter, using synthetic data, showed that one data source and its gradient could be used together as data sources for the MI thresholding algorithm, producing excellent results. Here, real-world data was used and it was shown that a spatial-gradient-based signal (the contour filtered image) can be combined with a temporal-gradient-based signal (the background difference) to produce good results, outperforming thresholding using individual signals alone.

4.2.3 Shadow detection

In this subsection, the detection of shadow pixels is targetted and we examine how the algorithm for MI thresholding can be adapted for this purpose. Specifically, the algorithm is extended to select, instead of a pair of thresholds, a pair of bounds that maximise agreement between sources.

Shadow detection is a useful component in background modelling algorithms, as it eliminates foreground pixel errors caused by colour changes due to shadows cast by moving objects. Other interesting work has been done by Finlayson et al. on removing shadows from still images [37], given some assumptions about the scene lighting. In



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.8: Example of additional high-quality-value peak in MI surface: (a) current image, (b) background image, (c) silhouette detection map (source 1), (d) contour detection map (source 2), (e) MI surface, (f) MI surface with peak locations superimposed, (g) plot of main diagonal of MI surface showing the peak at $(154, 154)$, (h) result of using the peak with highest quality, (i) result of using the peak with highest MI value. Results in (h) and (i) are colour-coded as follows: cyan indicates a detection in (c), yellow indicates a detection in (d) and red indicates a detection in both maps, navy blue indicates no detection in either source for that pixel.

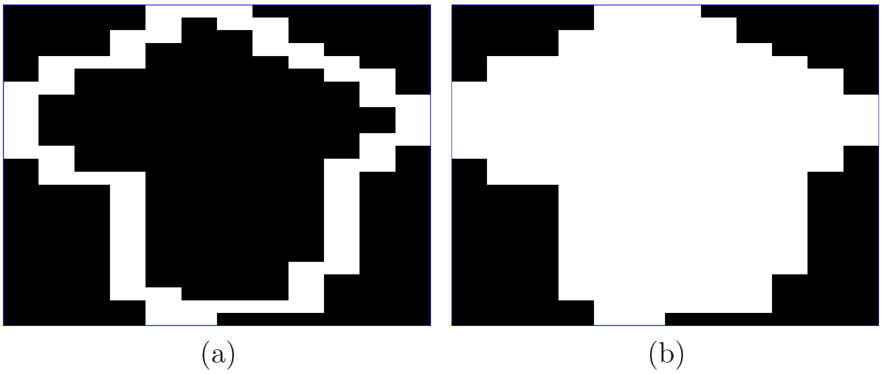


Figure 4.9: Pedestrian models used for experiments: (a) contour template and (b) silhouette template

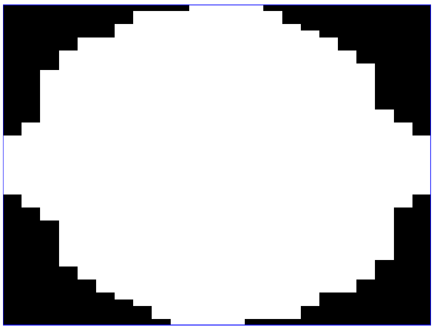
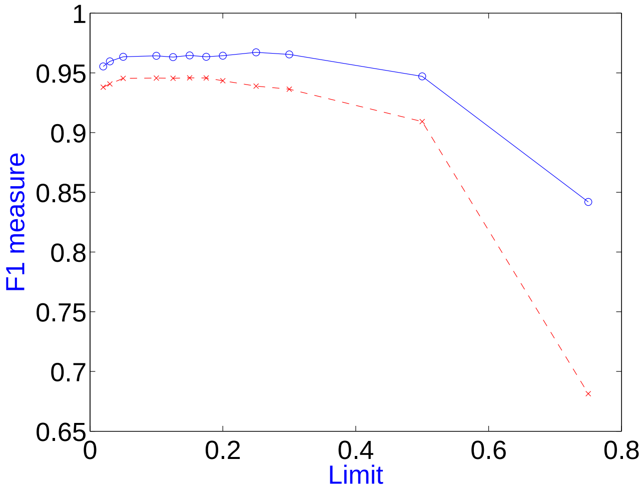
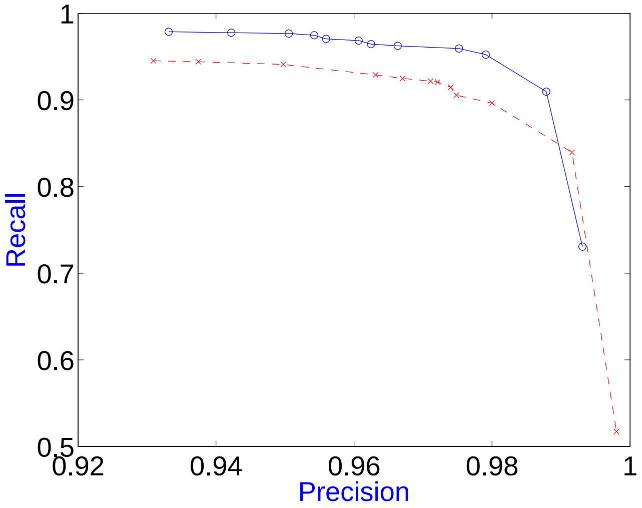


Figure 4.10: Pedestrian foreground model obtained by convolving the silhouette template with a unit impulse response and binarised using a threshold of zero.



(a) image

Figure 4.11: Performance of the EMIT system vs. the Kapur system using the F_1 measure to combine precision and recall. The x-axis is the lower limit for the fraction of pixels a person should account for (set to 0.15 in table 4.10): EMIT (Blue circles) and Kapur (Red crosses).



(a) image

Figure 4.12: Precision-recall curve for the EMIT system vs. the Kapur system as the limit value changes: EMIT (Blue circles) and Kapur (Red crosses).

Table 4.10: The results of using the EMIT algorithm and Kapur thesholding in the pedestrian detection system on the OTCBVS database are shown below

Sequence	People	EMIT		Kapur	
		Precision	Recall	Precision	Recall
1	91	0.977778	0.967033	1.000000	0.868132
2	100	0.969072	0.940000	0.989899	0.980000
3	101	0.989899	0.970297	0.952381	0.990099
4	109	0.981818	0.990826	0.981982	1.000000
5	101	0.950980	0.960396	0.941176	0.950495
6	97	1.000000	0.958763	0.989247	0.948454
7	94	0.978947	0.989362	0.948980	0.989362
8	99	0.988235	0.848485	0.965517	0.565657
9	95	1.000000	0.989474	1.000000	1.000000
10	97	0.978495	0.938144	0.946809	0.917526
Total	984	**0.981211**	**0.955285**	0.971092	0.921748

this application, shadow pixels can be modelled as a bounded decrease in brightness:

$$l_3 \leq \hat{V}_i \leq l_4 \tag{4.1}$$

where $\hat{V}_i$ is the relative change in luminance of pixel i compared to the background pixel, and is given by $\hat{V}_i = V_i / \max\{V_i^{BG}, 1\}$. V_i is the current luminance of pixel i and V_i^{BG} is the luminance of background pixel i. The selection of appropriate values for bounds l_3 and l_4 can be done empirically, or can be trained on pre-annotated data. However, if the *assumption* is made that shadows also cause a decrease in the pixel's colour saturation [108], then a second source of data is available that can assist in parameter selection. This assumption may not be true in general, but may be expected to be true on backgrounds with strong colour saturation. The shadow-pixel is modelled in saturation space as a bounded range given by

$$l_1 \leq \hat{S}_i \leq l_2 \tag{4.2}$$

where $\hat{S}_i$ is the relative change in saturation, and is given by $\hat{S}_i = S_i / \max\{S_i^{BG}, 1\}$. Given an image containing a cast shadow, applying equation (4.1) to the luminance change image produces a binary image. A binary image is similarly obtained by applying equation (4.2) to the associated saturation change image. If the parameters $\{l_1, l_2, l_3, l_4\}$ are selected correctly, then a strong *agreement* between the two binary masks is expected, which is the same assumption made for MI thresholding. Mutual information can be used as an agreement measure, as was used in the experiments of the previous chapter, but as examined in the discussion of the last chapter, there are other agreement measures that can be computed in a similar manner. We now briefly recap the basis for measuring binary agreement and then describe the proposed dynamic bounding algorithm that chooses a lower and upper bound for a single data source, such that the binary image it generates will be maximally in agreement with a second binary source.

Since only binary images are considered, a 4-value co-occurrence histogram is all that is needed to compute agreement. Given 2 binary images, X and Y, with N pixels each, we let u and v be binary-valued variables, with $C_{u,v}$ equal to the number of pixels whose classification is u in image X and v in image Y. The mutual information, μ_{XY}, between the pair of binary images, X and Y, is computed as follows:

$$p_{XY}(u,v) = \frac{C_{u,v}}{N} \tag{4.3}$$

$$p_X(u) = p_{XY}(u,0) + p_{XY}(u,1) \tag{4.4}$$

$$p_Y(v) = p_{XY}(0,v) + p_{XY}(1,v) \tag{4.5}$$

$$\mu_{XY} = \sum_{u \in \{0,1\}} \sum_{v \in \{0,1\}} p_{XY}(u,v) log \frac{p_{XY}(u,v)}{p_X(u) p_Y(v)} \tag{4.6}$$

As well as mutual information, another measure, mentioned in the last chapter, that has been frequently used to determine correlation between signals is Kendall's τ [67]. This measure can be computed using the same histogram counts:

$$\tau = \frac{p_{XY}(0,0)p_{XY}(1,1) - p_{XY}(0,1)p_{XY}(1,0)}{\sqrt{p_X(0)p_Y(0)p_X(1)p_Y(1)}}. \tag{4.7}$$

Alternative agreement measures, other than the two given here, are also possible, as discussed in the final section of the previous chapter, and are all functions of the four values of $C_{u,v}$. Regardless of the choice of agreement measure, maximising this measure requires finding the optimum parameters in high-dimensional space, 4-dimensions in the case of shadow detection. As with most complex high-dimensional problems, finding a global maximum cannot be guaranteed. However, the Simplex algorithm [92] or some other gradient ascent method could be used to find a good local maximum. It is proposed instead to use a dynamic programming-based solution, similar to the algorithm for choosing thresholds for maximising mutual information, to optimise two of the parameters at a time, iterating between data sources until converging on a solution.

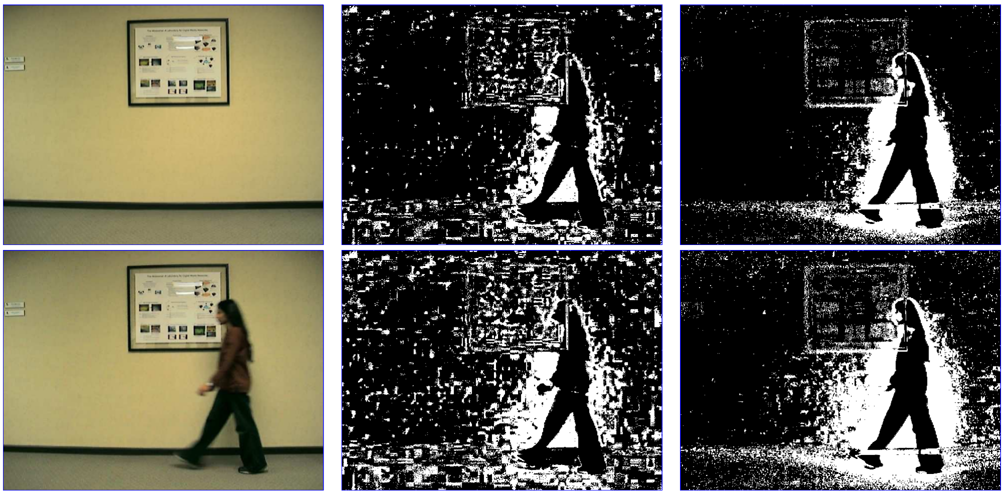
```
Input: Threshold list $\hat{A}$ and signals $X$ and $Y$
with $X = \{x_1, x_2, ..., x_L\}$, $Y = \{y_1, y_2, ..., y_L\}$
Initialise count maps to zero: $C_{*,*}(*, *) = 0$
$c_0 = \#\{k; x_k = 0\}$ // count zeros in binary signal
$c_1 = \#\{k; x_k = 1\}$ // count ones in binary signal
For all data points $(x_k,\, y_k)$
  Find largest $a_i \in \hat{A}$ such that $a_i \leq y_k$
  Find smallest $a_j \in \hat{A}$ such that $y_k \leq a_j$
  $C_{x_k,0}(1, 1) + +$
  if ($a_i$ and $a_j$ exist)
    $C_{x_k,0}(1, j) - -$
    $C_{x_k,0}(i + 1, j + 1) + +$
  end
$C_{0,0} = integralImage(C_{0,0})$ // integrate markers
$C_{1,0} = integralImage(C_{1,0})$ // integrate markers
$C_{0,1} = c_0 - C_{0,0}$
$C_{1,1} = c_1 - C_{1,0}$
```

Figure 4.13: Pseudocode for dynamic bounding algorithm

In the next section, the proposed dynamic bounding algorithm is explained in detail.

Dynamic bounding algorithm In order to choose the optimum pair of bounds that will maximise the agreement between the bounded image and the binary source, a brute-force search could be employed. Trying all pairs of thresholds from a discrete set of K elements has complexity in the order of $O(NK^2)$, where N is the number of pixels in the image. The dynamic programming algorithm described here is of order $O(K^2 + N)$ and evaluates all possible pairs of bounds in a discrete set.

The input to the algorithm is a discrete set of thresholds, $\hat{A} = \{a_1, a_2, ..., a_K\}$, a binary signal, X, and a real-valued signal, Y, of the same size as X. The goal is to select bounds for signal Y, such that when a binary signal, Y^*, is created using these bounds, its agreement with signal X is maximised. The output is a mapping array, $C_{p,q}(i, j)$, which gives the number of binary pairings of $x_k = p$ and $y_k^* = q$ when the bounds selected are a_i and a_j, with $i \leq j$. These counts can then be normalised and used in equation (4.6) or (4.7) to create an *agreement surface*, providing the agreement score for all possible bounding parameter selections. The bounds a_i and a_j that give the maximum agreement can then be selected. The pseudocode for the algorithm is given in figure 4.13. The *integralImage*() function refers to the standard dynamic programming method that efficiently replaces each pixel with the sum of all pixels in the rectangle whose opposite corners are this pixel and the pixel in $(1, 1)$ [149].



(a) background/image (b) bounded S (c) bounded V

Figure 4.14: Shadow parameter selection example 1: (a) the background image and the testing image, (b) detected shadow pixels in saturation-change images and (c) detected shadow pixels in brightness-change images. The top rows of (b) and (c) resulted from using Kendall's τ as an agreement measure. The bottom rows corresponds to the use of mutual information.

Results Figures 4.14 and 4.15 show the results of shadow detection on two images, with results shown from using Kendall's τ and MI as the agreement measure. The testing images came from the Terrascope dataset [58]. For the experiments, a median background image was used, and 256 equally spaced thresholds between 1/255 and 1. The four parameters are selected so as to maximise the agreement between the binary images obtained by bounding the saturation and luminance images, as in equations (4.1) and (4.2). For all tests, the initial parameters were set at $\{0.3, 0.97, 0.3, 0.97\}$, though other reasonable initialisations produced similar results. Parameters $\{l_1, l_2\}$ were optimised first, and then $\{l_3, l_4\}$. This continued until convergence.

Using mutual information as the agreement measure, image 1 (Gupta) converged to $\{0.3686, 0.9529, 0.4120, 0.9654\}$ in 4 iterations with a MI score of 0.0556. Image 2 (Crasto) converged to $\{0.4314, 0.9373, 0.5725, 0.9490\}$ in 6 iterations with a MI score of 0.0450. Using Kendall's τ as the agreement measure the method converges differently. Using the same initial parameters, Gupta converges in 7 iterations to $\{0.3686, 0.9294, 0.3889, 0.9500\}$ with $\tau = 0.3680$. The Crasto image converges in 5 iterations to $\{0.4549, 0.9333, 0.5725, 0.9490\}$ with $\tau = 0.3088$. Both agreement measures provide reasonable results for images in this dataset. Additionally, the proposed method is much more efficient than a Simplex search, which required over 150 iterations.

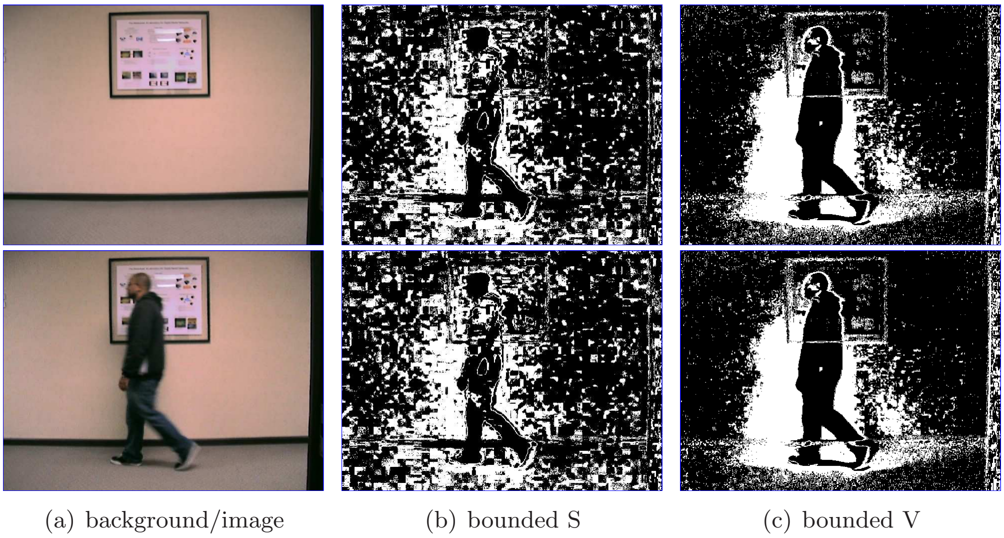

(a) background/image
(b) bounded S
(c) bounded V

Figure 4.15: Shadow parameter selection example 2: (a) the background image and the testing image, (b) detected shadow pixels in saturation-change images and (c) detected shadow pixels in brightness-change images. The top rows of (b) and (c) resulted from using Kendall's τ as an agreement measure. The bottom rows corresponds to the use of mutual information.

Overall, shadow detection using this method did not perform well on other data that was investigated, such as the ground-truthed shadow data provided by [108]. The assumption that saturation decreases is often not true as many backgrounds do not have strong colour content. Additionally, the two sources (luminance and saturation) cannot really be considered independent, as they come from the same sensor. In scenarios where the assumption is true, the method might be improved by first removing 'true foreground pixels', such as those whose hue has changed significantly. Later in this chapter, in section 4.3.3, a more practical application of the dynamic bounding algorithm is described, using thermo-visual information for adaptive skin detection.

4.3 Applications using strongly independent sources

4.3.1 Thermal and visible analysis

In this subsection, the use of thermal infrared and visual information is examined in a surveillance context, particularly in the detection of foreground objects of interest. Two experiments in this area are conducted.

In the previous chapter results were given on synthetic data, where it was shown that if common information is present, the MI values are much greater than those when no common information is present. Following on from this analysis, the first experiment in this subsection demonstrates how the MI value can be used to determine if common information is present in the scene, and this allows the detection of *empty frames* where no moving objects are present, and therefore no common information for the algorithm to exploit.

In the second experiment, the MI thresholding algorithm is used for detecting foreground in thermo-visual video data. This approach is compared to standard background modelling techniques, and also to more advanced models that contain shadow suppression modules. The proposed algorithm is shown to perform very well, using the common information shared by the thermal and visible signals to avoid the uncorrelted noise, but retain important object pixels.

Finally, a brief experiment is conducted, investigating the optimal smoothing procedure that was demonstrated synthetically in the last section. The distinct corner and peak of the MI and quality value graphs respectively are shown to be present in real data, as was the case with the synthetically generated data. However, the smoothing parameter selected at the appropriate scale is shown not to be optimal for the extraction of separate objects.

Empty-frame detection As the synthetic tests revealed in the last chapter, there is a good separation between the distributions of MI values when common information is present and when it is not. The experiment described here shows that it is possible to exploit the lack of common information to robustly detect when nothing is happening in the scene.

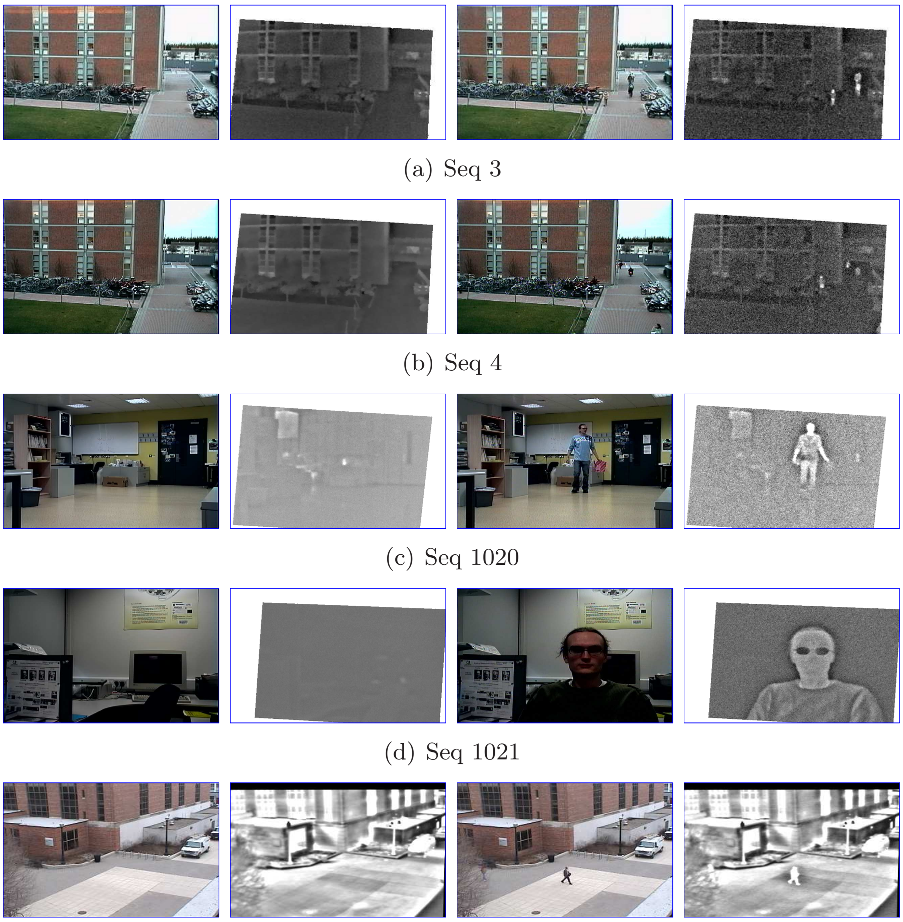
For 5 sequences, and a total of $22,225$ frames, those frames which contained no moving objects were manually annotated. Next, using a median background image for each sequence, in the visible and infrared spectra, the MI thresholding algorithm was used to detect appropriate thresholds. The MI value returned, m, was used to determine whether any common information was present. Specifically, the separation value, S, was computed using $S = (m - \mu)/\sigma$, with μ and σ given by equations 3.8 and 3.9, as before. This indicates how far outside the distribution of no-common-information the m value is placed. A threshold for the separation values was chosen as 3.5 to ensure that the vast majority of empty-frames were correctly detected. All frames whose separation value was below this threshold were determined to be empty frames. Since empty frames would occur together in periods, misclassified frames could be reduced using simple morphological operations. An erosion, followed by a dilation was carried out on the resulting binary results, using a 5×1 structuring element. Figure 4.16 shows examples of the images in the sequences used in this experiment. The last sequence is from the publicly available OTCBVS dataset, and the others were privately captured using the thermo-visual camera rig described in chapter 2. A graph of the separation values of sequence 2004 is shown in figure 4.17.

Using the ground-truthed annotation of the 5 sequences, the detection of empty frames was evaluated. An empty frame was declared if the separation value was less than 3.5. In the evaluations, a leeway of $+/-5$ frames was given at the boundaries of the annotated empty-frame periods to account for annotation error, and the uncertainty of determining exactly when an object has entered/left the scene. These boundary frames were ignored. Table 4.11 shows the results of the experiment on the 5 sequences. In sequence 3 for example, of the $1,409$ empty frames, all were detected except 4 (False negatives, FN). 146 false positives (declared as empty, but in fact contained a person) appeared but these were reduced to just 27 with the morphological postprocessing. The results appear quite positive, but some misclassifications occurred and they are now discussed.

In sequence 3, a thermally-weak person appears quite small in the scene, leading to the false positives detected. The empty frame detector believes there is nothing there because of the weak signal in both modalities, in terms of its size and its signal strength. More agressive morphological processing could remove these errors, since the separation value fluctuates around the threhold in this part of the sequence. In sequence 4, the false positives are caused by the presence of a man standing in the bike-rack, with a weak detection strength in both modalities. In the visible spectrum, he is mostly camouflagued by his black clothing and in the thermal infrared, his heavily insulated leather motorcycle clothing masks his temperature difference. In sequence 2004, the false positives are caused by a tree which heavily occluded the person entering the scene, giving them a weak signal in both modalities. On sequences 1020 and 1021 perfect classification is achieved, even without an post-processing. Over all sequences, it was found that when false negatives appeared, they occurred on the boundary of empty-frame periods, and would disappear if a leeway greater than 5 frames was used.

Instead of the simple morphological filtering that was used here, a better strategy might be to use a hysteresis approach, or the more complex Viterbi algorithm, to smooth the output and reduce misclassified frames. However, this simple post-processing operation achieves good results.

This *detection of nothing* is important to demonstrate, as it illustrates that the MI thresholding algorithm can, in a way, detect when it has failed, or when unfavourable conditions exist for it to operate optimally. The separation value itself can be interpreted as a confidence measure, indicating whether the algorithm may have produced *good* results.



(a) Seq 3

(b) Seq 4

(c) Seq 1020

(d) Seq 1021

(e) Seq 2004

Figure 4.16: Example backgrounds and sample images from the sequences used for empty-frame detection.

Sequence	Frames	Empty frames	Raw values		Post processed	
			FP	FN	FP	FN
3	11778	1409	146	4	27	4
4	5076	0	39	0	5	0
1020	1566	178	0	0	0	0
1021	792	111	0	0	0	0
2004	3013	1618	38	10	5	16

Table 4.11: Detecting empty frames by thresholding the MI separation value at 3.5: The FP and FN columns refer to false positives and false negatives respectively, where an epty frame was falsely detected or falsely missed.

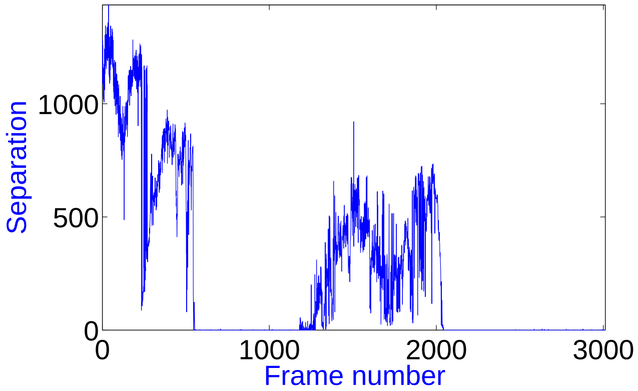


Figure 4.17: Separation values for sequence 2004: the two periods where no common information is present can be clearly seen between frames 550 and 1150, and between frames 2050 and 3000. Values below zero were clipped at zero for display purposes.

Thermo-visual foreground detection The background model used in this experiment is described in [63], which is an improved version of the seminal work of Stauffer and Grimson [137] on using mixtures of Gaussians to represent each pixel. The improvement over the traditional mixture of Gaussians approach was to introduce a faster initialisation method and to propose a single-parameter method of detecting shadows. In addition, the updating mechanism for the background model uses a truncated Gaussian, which causes the variance to slowly shrink. A minimum variance limit, not discussed in either paper, was needed to stop the variance of the models shrinking to zero. The parameters used for the background model are given in table 4.12. Where possible, the values given in the original paper [63] were used. Some values, such as σ_{init} were not specified so the values used were taken from a turorial paper on the mixture of Gaussians approach [107].

The algorithm outputs binary values for foreground and background, as well as marking some foreground pixels as shadows, when the shadow detection module is used. In order to make the background model suitable for MI thresholding, it was modified to output distance values as follows. Each new pixel is compared to all Gaussian models that formed its background model. Its absolute difference from the model mean was divided by the model standard deviation to obtain a distance value. Ordinarily, if this value is less than 2.5, a *match* is declared and the pixel is assumed to be a background pixel. Here, the minimum distance from the pixel to any of the background Gaussian models is used as the distance value for that pixel. Therefore, a distance map is obtained for the EMIT algorithm.

For this experiment two background models were used: One for RGB colour pixels and one for thermal infrared brightness. The visual model uses a 3-dimensional Gaussian per-pixel, whereas the thermal model uses one dimension. The video sequence used here was captured using the thermo-visual rig described in chapter 2 and contains $5,076$ frames. The scene is a busy pedestrian intersection in Dublin City University. A bike-rack is in the centre of the scene and a roadway is also included so cars can been seen passing by. To evaluate the algorithms, 29 images were selected from the sequence, approximately spaced over the entire duration. These images were manually annotated to mark foreground object pixels.

The difference images from both background models were fed into the MI thresholding algorithm, which produces two foreground results. The models also individually produce foreground results using the standard approach of thresholding at 2.5σ. A third improved foreground mask is created by taking the output of the visible spectrum model and removing pixels detected as shadows [63]. This shadow suppression module uses chrominance information not available to the MI thresholding algorithm. Since both the MI thresholding approach and the background model are pixel-based algorithms, no region-based filtering or morphological operations were performed on the resulting foreground images of any method.

The performance of each method of foreground pixel detection was determined using a manually annotated ground truth and observing the pixel classifications of each method. Table 4.13 gives the total precision and recall figures for all algorithms on the 29 ground-truthed foreground images. The F_1 measure, computed from the precision and recall values, shows that the MI thresholding algorithm out-performs the other algorithms in both the visible and infrared imagery. For a small decrease in recall, the precision of the foreground results are dramatically improved. The MI thresholding

Var	Val	Comment
λ	2.5	Distance (in σs) from mean value
		Standard foreground threshold value
K	5	Number of Gaussians
		value used in [63]
σ_{init}	30.0	initial Gaussian variance
		value used in [107]
w_{init}	0.05	initial Gaussian weighting
		value used in [107]
σ_{min}	3.0	Minimum standard deviation
L	500	Update rate parameter
		value used in [63]
T	0.6	Fraction of observed samples to model as background
		value used in [63]
τ	0.7	shadow threshold
		value used in [63]

Table 4.12: Parameters used in the background model.

Method	Precision	Recall	F_1
Stauffer Vis	0.035	0.887	0.068
Stauffer Vis (shadow removal)	0.168	0.793	0.278
MI Vis	0.653	0.743	0.695
Stauffer IR	0.242	0.502	0.326
MI IR	0.674	0.415	0.514

Table 4.13: Results of foreground pixel detection using standard background modelling vs. Mutual information thresholding.

algorithm exploits the fact that the objects appear in both modalities but false positives such as noise, infrared halos and visible shadows are uncorrelated in the sources, and therefore they are minimised in the foreground output. In figure 4.18 a number of example results are given of foreground returned by the tested algorithms. The shadow suppression does well to remove incorrect foreground pixels but still contains high noise. The proposed method clearly avoids noise by exploiting common information between modalities.

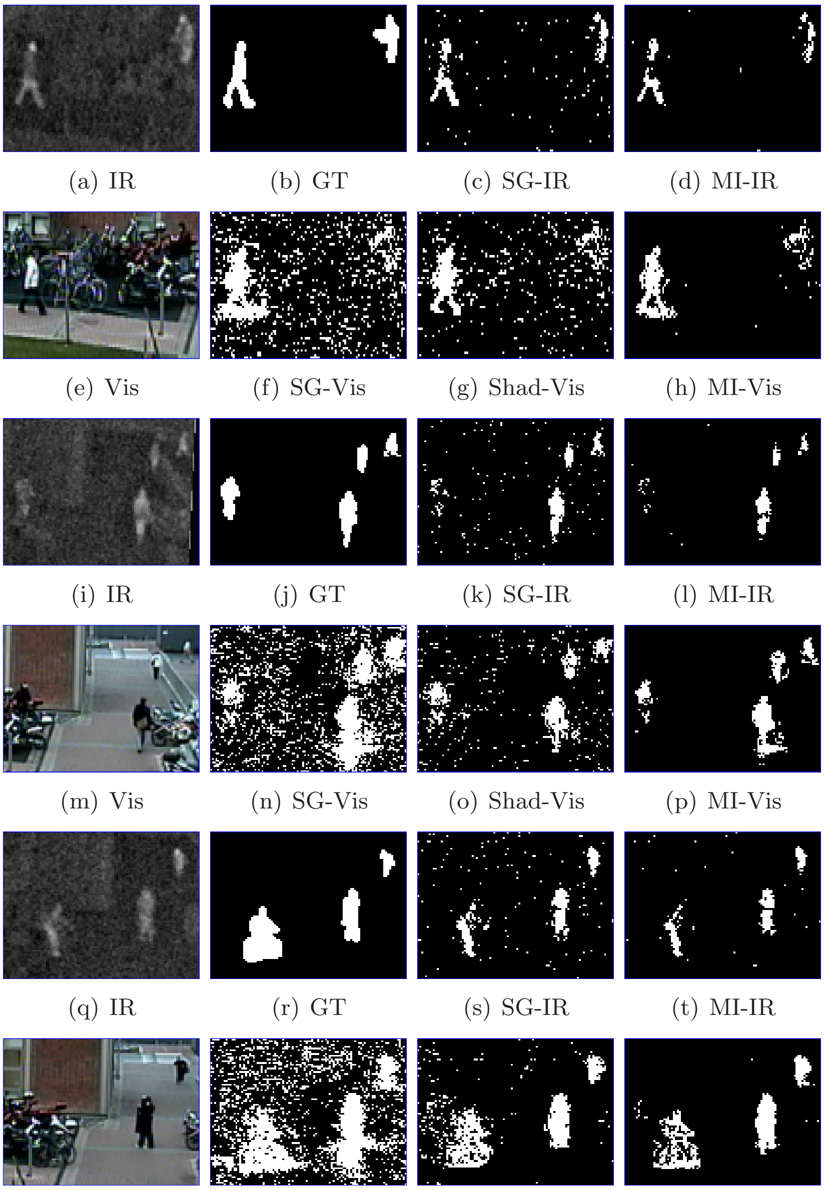
Optimal smoothing In the previous chapter, it was shown that spatial information could be exploited by smoothing the difference images before using the MI thresholding algorithm. Using synthetic data, the optimal scale for smoothing was found to produce a peak in the quality plot and a corner in the MI graph. This approach is now briefly investigated for real data.

A median background image was computed in the visible and infrared domains. The difference image in the visible spectrum was computed using the Euclidian distance in RGB space from the image pixel to the background image pixel. The infrared difference image was simply the absolute difference between the current image and the background image. The images and difference images are shown in figures 4.19(a), (b), (e) and (f).

By smoothing both difference images with a Gaussian kernel of width σ and using the smoothened signals as input to the MI thresholding algorithm, an MI value and a quality value are returned. Plotting these values on two graphs as σ is varied results in the plots of figure 4.19(c) and (g), for MI and quality respectively. The peak quality value occurs at $\sigma = 13.6$. A corner appears at a similar location on the MI graph. Using the value of σ, the resulting thresholded images are shown in figure 4.19(h). Here the thresholded visible and infrared images are shown together, fused colourwise to show foreground of visible only (blue), infrared only (yellow) and both (white). This can be compared to the unsmoothed case in figure 4.19(d). It is evident that although the graphs match the synthetic results and clearly indicate a value for σ, this value causes objects to be merged together due to their close proximity. In the next section, we revisit optimal smoothing, but in the context of event detection using audio and visual features.

4.3.2 Audio and video surveillance

In most of the preceding discussion, the word *pixels* was used to refer to the samples of data from the two detection signals. Although this word suggests that the input must be some form of visual data, any type of detection signal can be used. In this



(a) IR (b) GT (c) SG-IR (d) MI-IR

(e) Vis (f) SG-Vis (g) Shad-Vis (h) MI-Vis

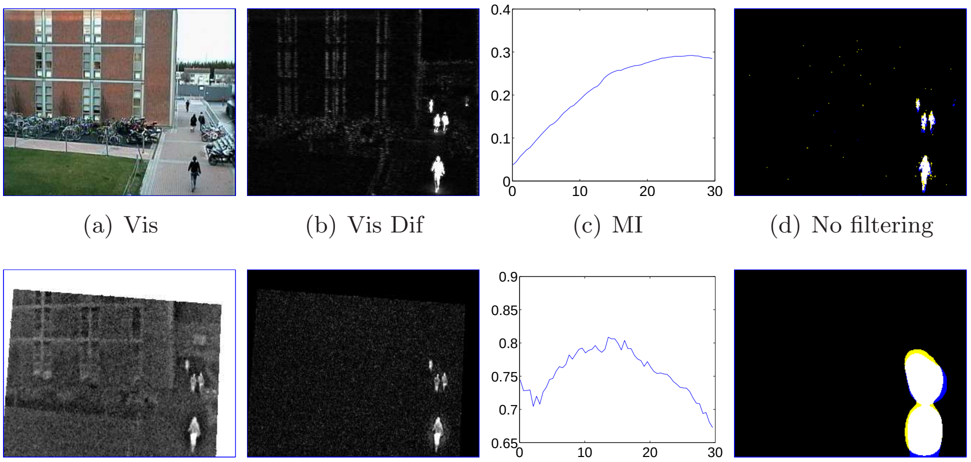
(i) IR (j) GT (k) SG-IR (l) MI-IR

(m) Vis (n) SG-Vis (o) Shad-Vis (p) MI-Vis

(q) IR (r) GT (s) SG-IR (t) MI-IR

(u) Vis (v) SG-Vis (w) Shad-Vis (x) MI-Vis

Figure 4.18: Example results comparing Stauffer and Grimson background modelling to MI thresholding: Images show the visible (Vis) and infrared (IR) frames, the annotated ground-truth (GT), infrared foreground detected using Stauffer and Grimson method (SG-IR) and MI thresholding (MI-IR), and visible foreground detected using Stauffer and Grimson method (SG-Vis), shadow-suppressed Stauffer and Grimson (Shad-Vis) and MI thresholding (MI-Vis). No filtering or morphological processing was performed on any of the images.



(a) Vis (b) Vis Dif (c) MI (d) No filtering

(e) IR (f) IR Dif (g) Quality (h) Filtered at $\sigma = 13.6$

Figure 4.19: Optimal smoothing of thermo-visual images. The MI and quality value plots in (c) and (g) indicate optimal smoothing at $\sigma = 13.6$, but this results in objects losing their separate identities, seen by comparing the unsmoothed (d) and smoothed results (h).

section, another application of using mutual information thresholding is given, this time to robustly detect surveillance events, namely people passing through a corridor, using noisy visual and audio detection signals.

Figure 4.20 shows the view of the corridor from a surveillance camera. The corridor has windows along one side, which cause frequent changes in the corridor light levels. An omni-directional audio microphone was placed directly beside the camera. Since it is not directional, it picks up sounds from behind the camera, as well as outside noise through the windows. The alignment of the detection sources is not perfect, since the camera has a narrow, but long field of view, looking down the corridor, whereas the microphone detects sound in a spherical area around it. However, since their area of detection overlap is in the near-field, the location where people are to be detected, sounds or visual detections outside this area should appear as uncorrelated false positives to the other modality.

Video and audio were recorded for a period of 1 hour. The frame-rate was 1 frame per second and the audio was sampled as 44kHz. From the visual data, 9 different detection features were extracted. We now examine how MI thresholding can be used to determine which features are best suited for the detection task, by exploiting the redundant information shared by the audio detection signal. The 9 features are as follows:

(a) Brightness difference Consecutive frames are subtracted and the sum of absolute differences between each pixel is computed. This feature can be affected by sudden brightness changes due to sunlight.

(b) Thresholded Brightness difference using the Kapur method This feature is the same as (a) above but the absolute difference of each pixel is thresholded using the Kapur method before summing them. Therefore the result is a sum of binary values.

(c) Thresholded Brightness difference using the Rosin method This feature is the same as (b), but uses the Rosin method instead of Kapur.

(d) Gradient Difference The gradient of each image is computed and consecutive gradient images are subtracted. The minimum of the gradient and the gradient difference is computed for each pixel as follows:

$$F_d = \sqrt{(\min((\nabla_x I_t)^2 + (\nabla_y I_t)^2, (\nabla_x I_{t-1} - \nabla_x I_t)^2 + (\nabla_y I_{t-1} - \nabla_y I_t)^2))}; \qquad (4.8)$$

The sum of F_d values of all pixels is the image gradient difference. This feature should be more robust to lighting changes than the previous three which are based on brightness.

(e) Gradient difference using the Kapur method This feature is the same as (d) above, but the absolute difference of each pixel is thresholded using the Kapur method before summing them. Therefore the result is a sum of binary *edge-change* values.

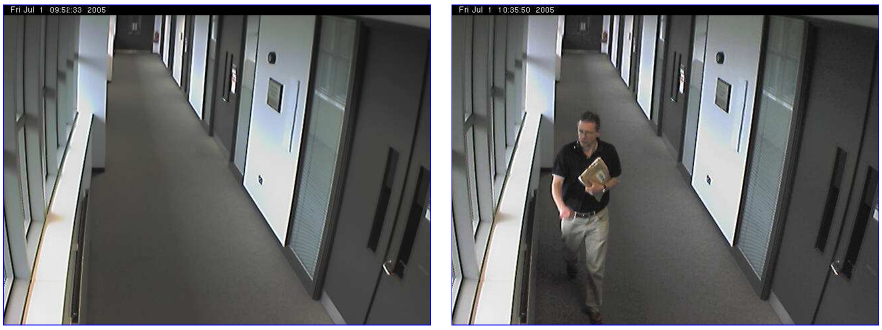
(f) Gradient difference using the Rosin method This feature is the same as (e), but uses the Rosin method instead of Kapur.

(g) Normalised Cross-correlation between consecutive images A block-based normalised cross-correlation (NCC) is computed between consecutive images. A non-overlapping block size of 64 × 64 pixels is used. This size block was chosen so that no block contained only an untextured region, which could cause false jumps in the value due to noise. The NCC value for all blocks are added, and then this total is subtracted from the maximum possible sum. This gives a value of zero for two identical images and larger values for different images.

(h) Normalised Cross-correlation with background image This feature is the same as (g), except that a median background image is used instead of the previous frame.

(i) Background brightness difference This feature is the same as (a), except that a median background image is used instead of the previous frame. The background image does not change, so this feature is not robust to gradual changes in the scene brightness.

An *audio-detection signal* was created by computing the audio RMS value over 5 second windows, giving a total of 899 samples. Each of the visual feature was resized to this length by taking its maximum value over a 5-sample sliding window. The windows for the audio and visual feature filtering were partially overlapping, shifting by 4-samples each time. Figure 4.21 shows plots of these 9 features, as well as a plot of the audio detection signal. MI thresholding was performed using each of the visual features with the audio detection signal.

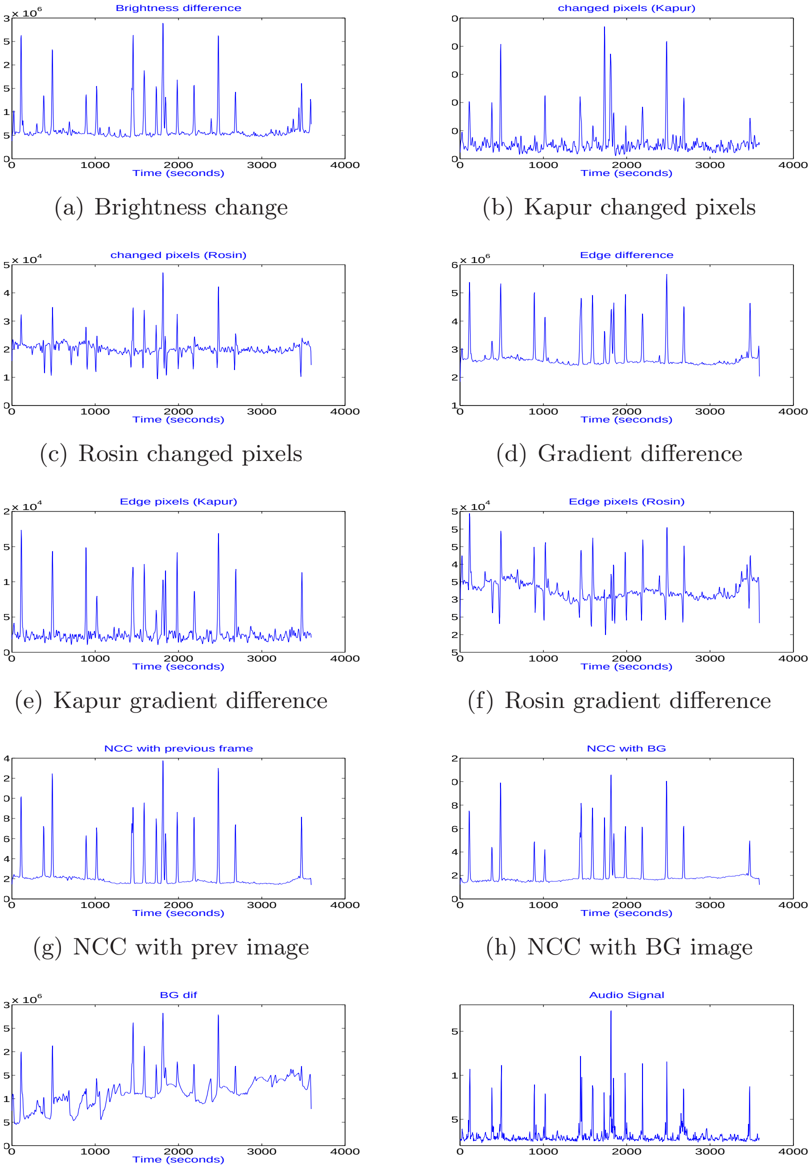


(a) Background image

(b) Sample image

Figure 4.20: The camera's view of the corridor.

Optimal Smoothing Figure 4.22 shows the changes in peak MI score and quality that occur as both sources are smoothed using a Gaussian filter of increasing scale. Four of the nine features are shown as representative examples. Three of these graphs parallel very strongly the results of using synthetic data to determine the optimal filtering scale, as shown in the previous chapter in figure 3.16. There is a sharp, linear rise in the MI value up to the optimal scale, then the MI values rise more slowly. Using the corner-finding method on which the Rosin thresholding method [116] is based can help automatically determine the optimal scale from the MI plot. Similarly, in the quality value plot, the maximum quality value occurs around the optimal scale. As



(a) Brightness change

(b) Kapur changed pixels

(c) Rosin changed pixels

(d) Gradient difference

(e) Kapur gradient difference

(f) Rosin gradient difference

(g) NCC with prev image

(h) NCC with BG image

(i) Background difference

(j) Audio detection signal

Figure 4.21: Plots of the 9 visual feature and the audio volume feature

seen from the graphs, the peak quality for feature (g) occurs around 1.7 and around 1.8 for feature (h). A similar optimal smoothing value occurs for most of the other features also. Table 4.14 shows the selected σ values for each feature, using both the corner of the MI graph and the peak quality value. The notable exceptions are the Rosin-thresholding based features, and the background brightness difference (feature (i)). While table 4.14 indicates a peak quality for the brightness difference feature (feature (a)) at $\sigma = 3.9$, the plot in figure 4.22(b) shows that it also has a peak at around $\sigma = 1.8$. Therefore, this feature does agree with the majority of features that the optimal smoothing parameter lies close to $\sigma = 1.8$.

Feature	Corner	Peak Quality
a	2.0000	3.9000
b	1.8000	1.8000
c	5.8000	2.2000
d	2.7000	1.7000
e	1.7000	1.8000
f	5.9000	0.7000
g	1.8000	1.8000
h	1.8000	1.8000
i	1.5000	4.4000

Table 4.14: Selected σ values for each feature, using the corner of the MI curve and the peak quality value.

The Rosin-thresholding based features, as shown in figure 4.21(c) and 4.21(f), have distinctive peaks and troughs. The peak/troughs in the Rosin-thresholding based features are caused by the camera setup and how people approach the camera. The peak is caused by the foreground object. At the far end of the corridor is a dark-gray coloured door, against which people are not very distinct. In the histogram, this appears as if the background distribution has widened, therefore increasing the threshold and lowering the number of changed pixels detected, causing a visible ‘dip’ in the graph. It is in fact possible to determine whether a person is walking away-from or towards the camera by whether the dip occurs before or after the trough. This property of the Rosin-based features makes them unsuitable for smoothing, since Gaussian smoothing would excessively reduce the peaks’ heights due to their proximity to the troughs. It was found that the occurance of peaks/troughs was not a general feature of Rosin’s method, but related directly to the experimental setup, as explained.

The background brightness difference, feature (i), is a weak feature for event detection in this application. Although the background image is largely free of noise, as it was computed using a median of many images, it is not robust to the gradual change in lighting level, therefore its value has gradually increased as seen in figure 4.21(i), making it unsuitable for thresholding with a single threshold. Its MI value graph does not increase monotonically, as most others do, but has a peak, close to $\sigma = 1.8$ interestingly enough.

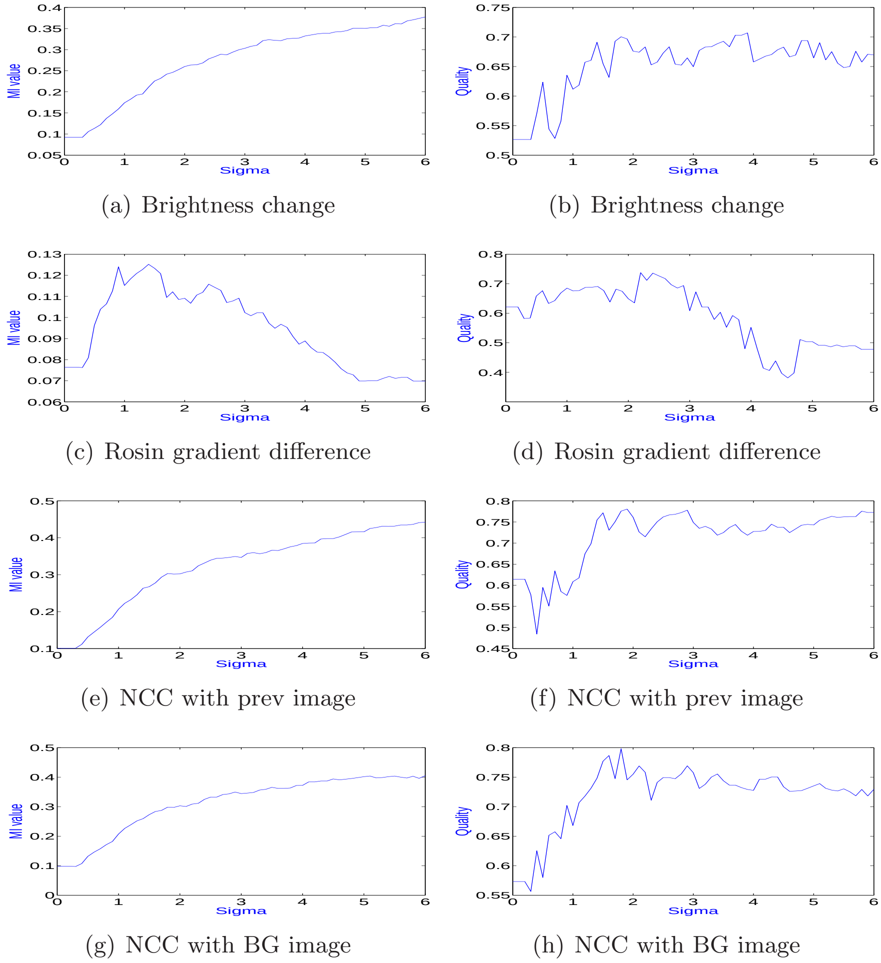
The rest of features, excluding (c), (f) and (i), indicate $\sigma = 1.8$ as a good smoothing parameter. Even feature (a), although 3.9 is selected, the graph in figure 4.22(a) shows a similar peak at 1.8. The corner finding method did not always select 1.8, but visual inspection of the graph showed that a range of σ values around this point seemed suitable. Therefore, the smoothing value was set at $\sigma = 1.8$ and filtering was done on all signals using a Gaussian filter of this width. The affects of smoothing on performance are now examined.

Feature Selection In trying to determine which visual features are most useful for this event detection task, each one can be plotted as a point on the quality-MI plane. Figure 4.23(a) shows this plot. It seems that there is a divide between good features (a,d,e,g,h) and poorer features (b,c,f,i). However, it is difficult to distinguish between (d) and (e), for example, since (d) has a higher mutual information value and (e) has a higher quality.

By smoothing the feature signals, using $\sigma = 1.8$, our ability to discriminate between the features can be improved. Previously, it was discovered that $\sigma = 1.8$ appeared as the optimal scale at which to perform smoothing of the data. Figure 4.23(b) shows how the smoothing affects the positions of each of the visual features on the plane. The plot appears far more linear, and therefore easier to distinguish how well features should perform. Now features (d,g,h) appear to be the top performers.

This is verified in figure 4.24, where the performance of each feature is measured using a manually annotated ground truth (GT). The Haussdorff distance is used to measure performance, with smaller values indicating a better result. The GT annotation was done by noting whether a person is present in each of the 899 data samples. This ground truth is shown in figure 4.25, marked as GT. The ground truth, along with the thresholded visual signals are shown in figure 4.25. Signals are shown for the smoothed and unsmoothed cases.

Comparing figures 4.23(b) and 4.24(b), a very strong connection is seen between signal agreement (mutual information and quality) and signal performance in event detection. Features (d,g,h) show the highest agreement with the audio signal, and also



(a) Brightness change

(b) Brightness change

(c) Rosin gradient difference

(d) Rosin gradient difference

(e) NCC with prev image

(f) NCC with prev image

(g) NCC with BG image

(h) NCC with BG image

Figure 4.22: Plots of the MI peak value (a)(c)(e)(g) and the quality score (b)(d)(f)(h) as smoothing increases. The rows correspond, from top to bottom, to the visual features in figure 4.21(a), (c), (g) and (h). The normalised cross-correlation (NCC) feature plots comply strongly with the results of the synthetic data smoothing tests.

perform best. These are followed by features (e) and (a), then (b) and finally, the worst performers, (c), (i) and (f). The smoothing is necessary to take the spatial relations of the samples into account. The MI-quality plot of figure 4.23(a) does not clearly indicate which feature is better. After smoothing, feature (a) performs well and (c) performs very poorly, and the MI-quality plot is clear on which provides the best signal agreement.

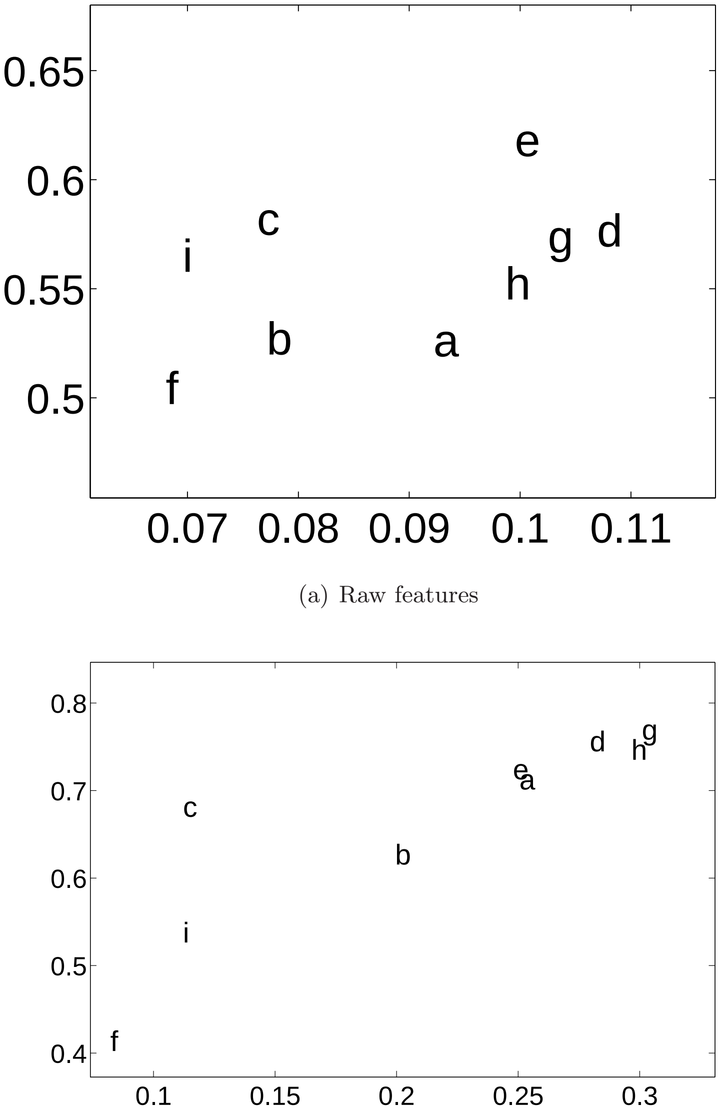
Interestingly, without the smoothing, feature (a) is the worst performer, with many false positives and feature (c) performs very well. After smoothing, feature (a) performs well and (c) performs very poorly.

It is clear from this application that MI thresholding can not only assist in multimodal systems with disparate data sources, but can also be used to guide the selection of features for object/event detection, in order to optimise system accuracy. In this section, the selection of visual features was examined, using an audio detection signal as a reference. It was shown that it is possible to determine which features are most useful, without using any ground-truth or manual annotation, but simply by observing the position of the features on the quality-MI plane. Observing the changes in MI value and quality can also assist in feature selection. Features whose MI values do not generally increase are usually poorly performing features. Also, the procedure for the selection of an appropriate scale for signal smoothing, justified synthetically in the last chapter, has been shown to perform well on real data here.

4.3.3 Skin-colour model learning

In this section, the dynamic bounding algorithm, introduced for shadow detection in section 4.2.3 to extend the MI thresholding algorithm, is used to dynamically choose colour and thermal bounds for automatic skin detection in thermo-visual imagery. Using two separate skin detectors, one based on colour, the other on infrared, the detectors negotiate to find the best parameters to model skin. This proposed approach has some conceptual similarities to the *co-training* method of Blum and Mitchell [12] where two independent classifiers are used to train each other.

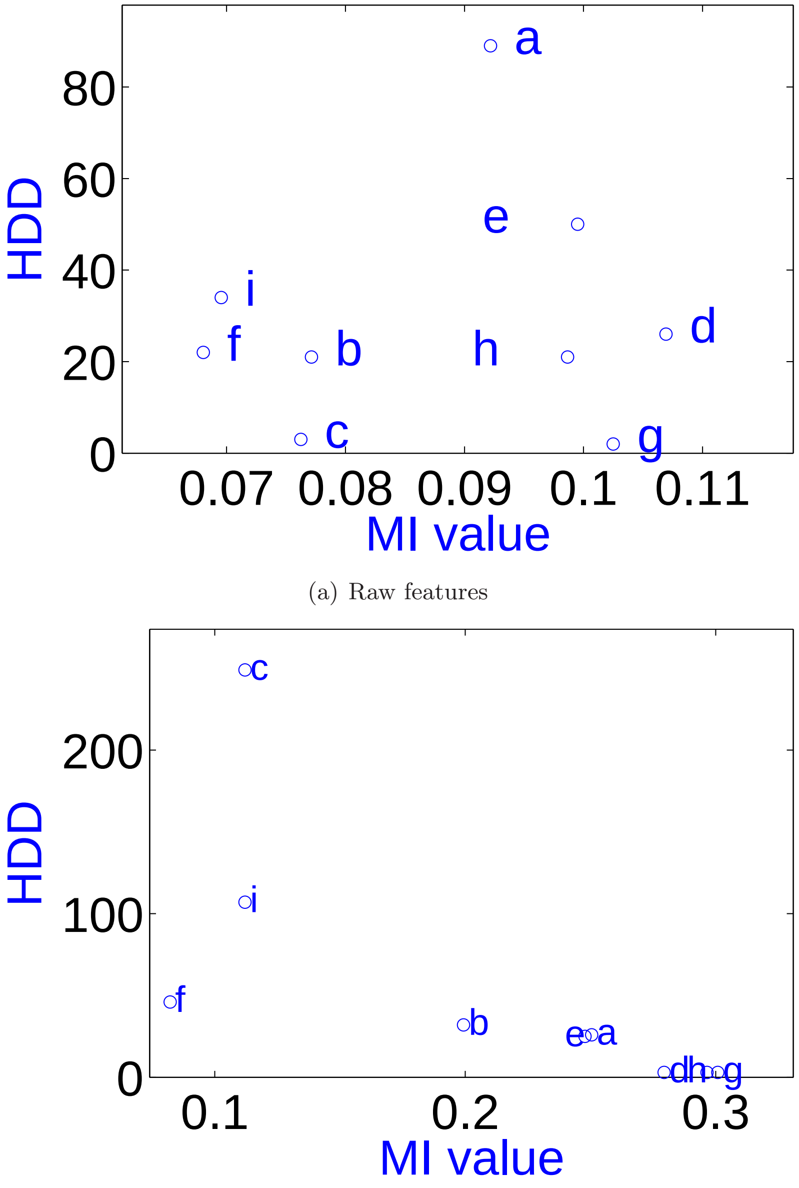
Skin pixel modelling Figure 4.26 shows a colour image and its corresponding thermal infrared image. Skin pixels lie in a particular subspace in both the thermal and visible domains. Similar to the shadow detection application in section 4.2.3, simple bounds are used to model skin in both the colour and infrared domains, and the shared information between the modalities can be exploited to compute the parameters for both these subspaces. In the visible domain, a certain bounded subspace of the HSV



(a) Raw features

(b) Smoothed with $\sigma = 1.8$

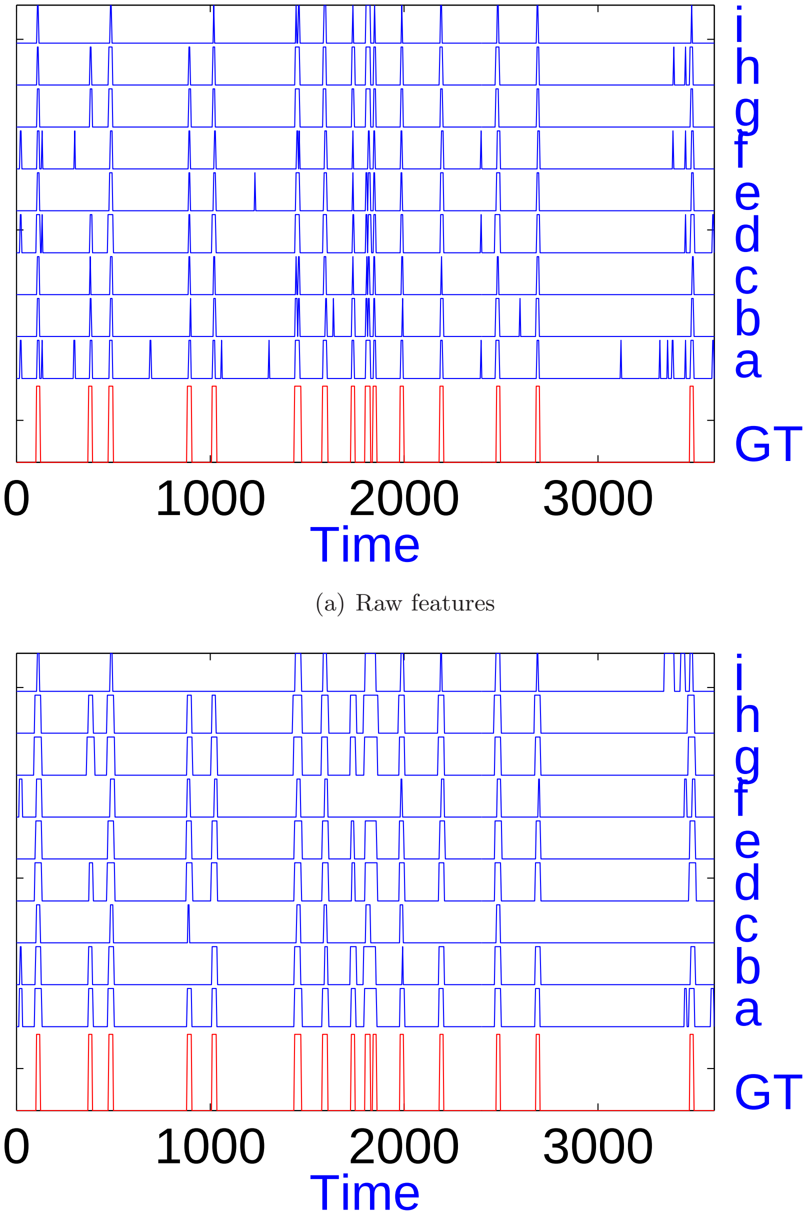
Figure 4.23: Effects of optimal smoothing on signal agreement: MI-quality plot of features before (a) and after (b) smoothing with $\sigma = 1.8$.



(a) Raw features

(b) Smoothed with $\sigma = 1.8$

Figure 4.24: Effects of optimal smoothing on performance: Haussdorff distance between MI thresholded feature and the ground-truth, before (a) and after (b) smoothing with $\sigma = 1.8$.



(a) Raw features

(b) Smoothed with $\sigma = 1.8$

Figure 4.25: Comparison of thresholding raw data and smoothed data. The ground truth (GT) is shown at the bottom of each figure and the nine visual features are shown above it, labeled on the right (a) to (i). These feature labels correspond to the visual features in figure 4.21

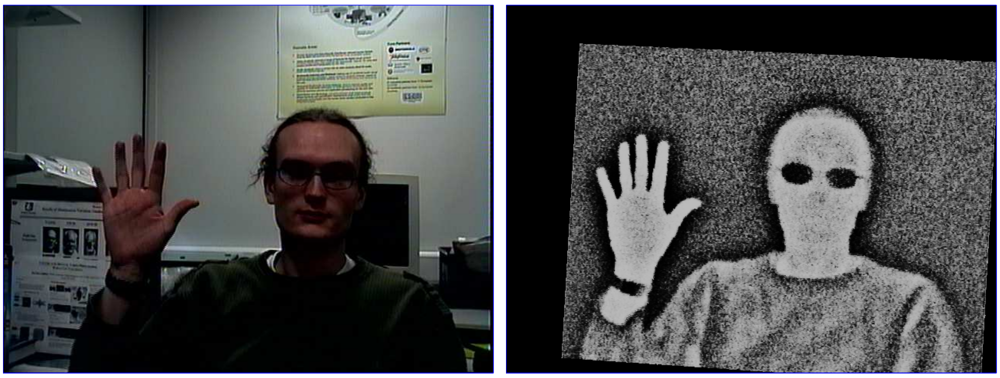


Figure 4.26: Examples of (a)visible and (b)infrared input images

space is selected to indicate a possible skin pixel. Using $\{l_1, l_2, l_3, l_4, l_5, l_6\}$ as the boundaries of the subspace, a pixel i belongs to this subspace if its colour components in HSV space, (H_i, S_i, V_i) conform to:

$$l_1 \leq H_i \leq l_2 \tag{4.9}$$

$$l_3 \leq S_i \leq l_4 \tag{4.10}$$

$$l_5 \leq V_i \leq l_6. \tag{4.11}$$

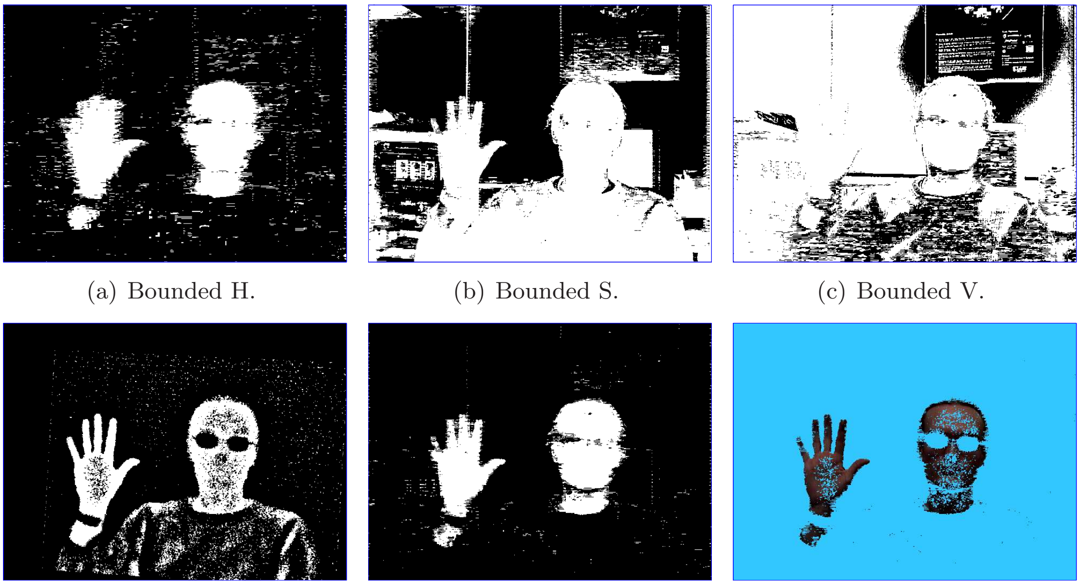
Since the hue component can be considered circular, the hue wheel is rotated using $H_i \leftarrow (H_i + 128)\ mod\ 256$, so that red, the dominant hue in skin pixels, is in the centre of the band. In the thermal infrared images, a similar model is used for the appearance of skin pixels, with pixel I_i being a potential skin pixel if

$$l_7 \leq I_i \leq l_8 \tag{4.12}$$

where $\{l_7, l_8\}$ are the thermal brightness boundaries. Therefore, the parameters for the models are fully represented by $L = \{L_{VIS}, L_{IR}\} = \{\{l_1, l_2, .., l_6\}, \{l_7, l_8\}\}$.

In figure 4.27, examples of the use of these models are shown in relation to figure 4.26. Setting $L = \{\{78, 159, 60, 255, 3, 139\}, \{67, 137\}\}$ maximises the Kendall's τ agreement measure. Pixels within the hue, saturation and value boundaries are shown in figure 4.27(a)-(c). Figure 4.27(e) combines (a)-(c), showing pixels that are within all the colour boundaries, and are considered possible skin pixels. Figure 4.27(d) shows infrared pixels that fall within the thermal boundary, and are therefore considered possible skin pixels.

Ideally, if there are skin regions present in the scene, and there are not many skin-like distractors present in visible or infrared, then there should be a high level of agreement between the binary images in figure 4.27(d) and (e). By selecting pixels that appear as skin in both modalities (binary AND fusion), figure 4.27(f) is produced.


(a) Bounded H.

(b) Bounded S.

(c) Bounded V.

(d) Bounded IR.

(e) Bounded HSV.

(f) Likely skin pixels.

Figure 4.27: Examples of bounded (a)hue, (b)saturation, (c)value, (d)infrared and (e)HSV. Binary AND fusion of (d) and (e) produce the skin pixels in (f).

System overview The input to the system is a colour image, the corresponding thermal infrared image and an initialisation method. The initialisation method provides a binary image either from the colour or thermal image. The other modality's bounds will be optimised to maximise agreement. Bounds are then alternatively optimised iteratively until convergence.

Figure 4.29 shows the two initialisation methods used in this work. The first method applies a dynamic threshold to the IR image using Rosin's method [116]. The second method uses predefined colour bounds to provide the initial binary image. In the experiments, the M parameter is set as $M = 255/5$, which sets quite a broad range, so almost all skin-like pixels will be included. After initialisation, the system will iteratively optimise all the parameters until it converges, as illustrated in figure 4.28. The *flag* variable indicates whether the IR bounds should be optimised first. When optimising pairs of colour bounds, such as hue bounds $\{l_1, l_2\}$, some pixels may already be excluded since they are outside the other colour bounds. For example, if the saturation bounds are set at $[l_3 = 10, l_4 = 100]$, then HSV pixel $(50, 150, 100)$ is already classified as non-skin, regardless of any changes to the hue bounds. These pixels are catered for by excluding them from processing and adding them on to the appropriate counts at the end (either to $C_{0,0}$ or $C_{1,0}$). The final outputs are (i) the set of 8 parameters, L, (ii) 2 binary maps (one for each modality) and (iii) an agreement value score. While mutual information performs well overall, it produces incorrect skin in some instances, since the MI value does not change if one of the binary images is inverted. When using bounds, and not simply a threshold, this can cause ambiguity over the correct parameters to choose. For example, using bounds $\{0, 128\}$ could give the same amount of agreement as using $\{129, 255\}$. As this is not a desirable property, Kendall's τ is chosen as the agreement measure.

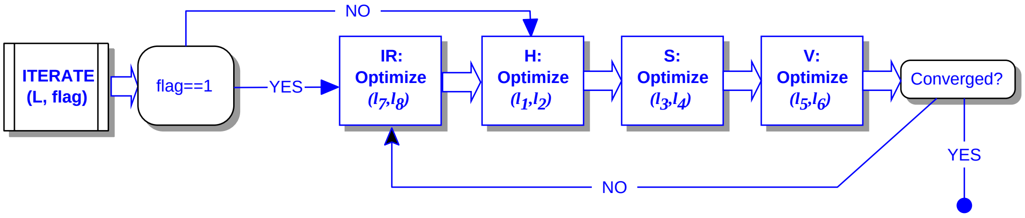
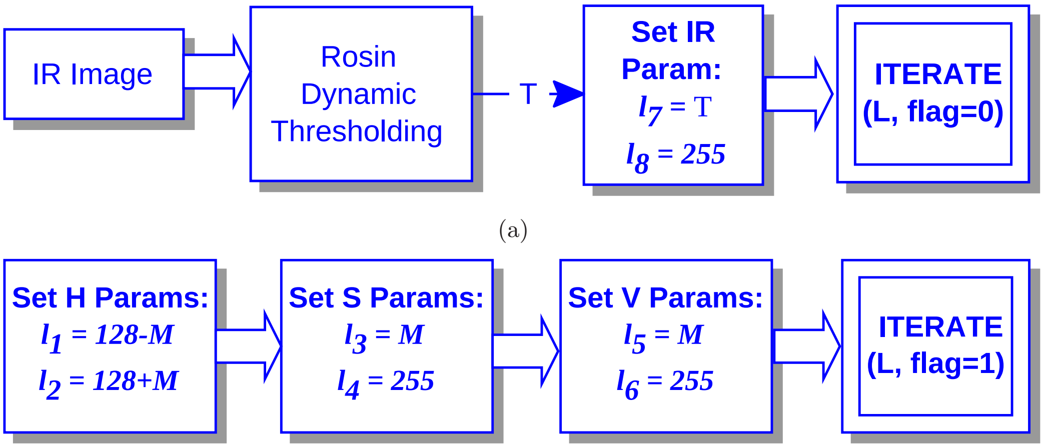

ITERATE (L, flag)
flag==1
YES
NO
IR: Optimize (l_7,l_8)
H: Optimize (l_1,l_2)
S: Optimize (l_3,l_4)
V: Optimize (l_5,l_6)
Converged?
YES
NO

Figure 4.28: Iteration function


IR Image
Rosin Dynamic Thresholding
T
Set IR Param: l_7 = T l_8 = 255
ITERATE (L, flag=0)

(a)

Set H Params: l_1 = 128-M l_2 = 128+M
Set S Params: l_3 = M l_4 = 255
Set V Params: l_5 = M l_6 = 255
ITERATE (L, flag=1)

(b)

Figure 4.29: Initialisation Methods: (a)Infrared-based and (b)Colour-based initialisation

Initialisation evaluation In order to compare the initialisation methods of figure 4.29, the proposed algorithm was run on $6,697$ images from 7 thermo-visual video sequences. The experiment investigated which method would cause convergence to the highest agreement value. The results are given in table 5.1. Both methods converged in a similar number of iterations on average, as shown in columns 3 and 5. Neither method showed superiority, with both methods having roughly equal performance on average, and converging to the same parameters about one-third of the time. Sequence D contains a lot of skin-like pixels, due to the colour of the floor, causing the colour-based initialisation to perform poorly in this sequence. An example frame from this sequence is shown in the top row of figure 4.31. On the other hand, sequence E contains many people and therefore a lot of 'hot' pixels, causing the infrared-based initialisation to perform poorly in this sequence. Similarly, an example frame from this sequence is shown in the third row of figure 4.31.

Using 16 ground-truthed images of skin pixels, an objective evaluation of the initialisation methods was conducted. Figure 4.30 shows how a change in agreement value (Kendall's τ) affects the detector performance, measured using the F_1 measure. The x-axis shows the increase in agreement when initialisation method 1 (figure 4.29(a)) is used instead of initialisation method 2 (figure 4.29(b)). The y-axis shows the change in performance. Blue circles indicate the detected colour skin images and red crosses indicate the infrared detected skin images. Six circles and six crosses are plotted on the origin, since the two methods converged to the same set of parameters in 6 of the 16 images. Because all points lie in the upper-right or lower-left quadrants, this shows that an increase in agreement is strongly correlated with an increase in performance. By running the algorithm twice, once with each method, and selecting the set of parameters with greater agreement, high quality skin detection is obtained. Examples of detected skin are shown in figure 4.31.

Bound adaptation To investigate the adaptivity of the bound selection, changes in the appearance of skin were introduced in the video data. As skin usually appears brighter than other objects in infrared, a kettle full of hot water is introduced to the scene to act as a significant distractor. This causes the thermal camera to gradually adjust its contrast to adapt to the new high-temperature object. The proposed method dynamically adapts to this, as shown in figure 4.32, by decreasing the IR bounds for skin accordingly, as the contrast decreased.

For the colour bounds, figure 4.33 demonstrates a similar case of adaptation. At the beginning, the scene contains two people with similar skin tones. When a third

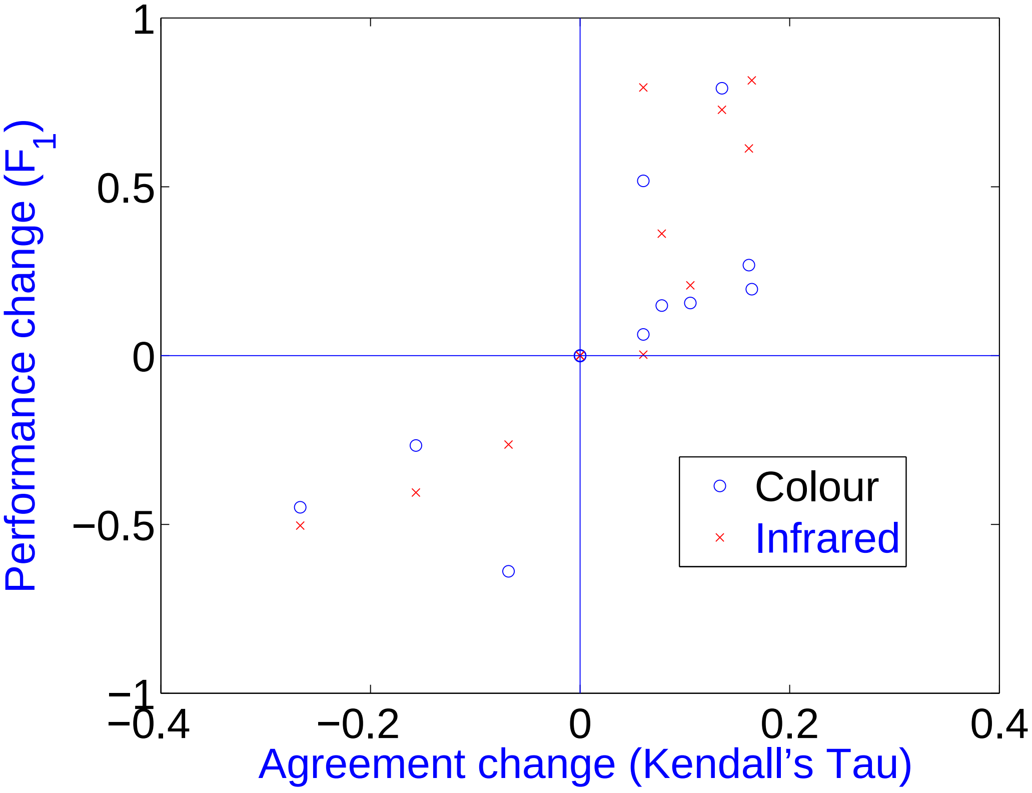
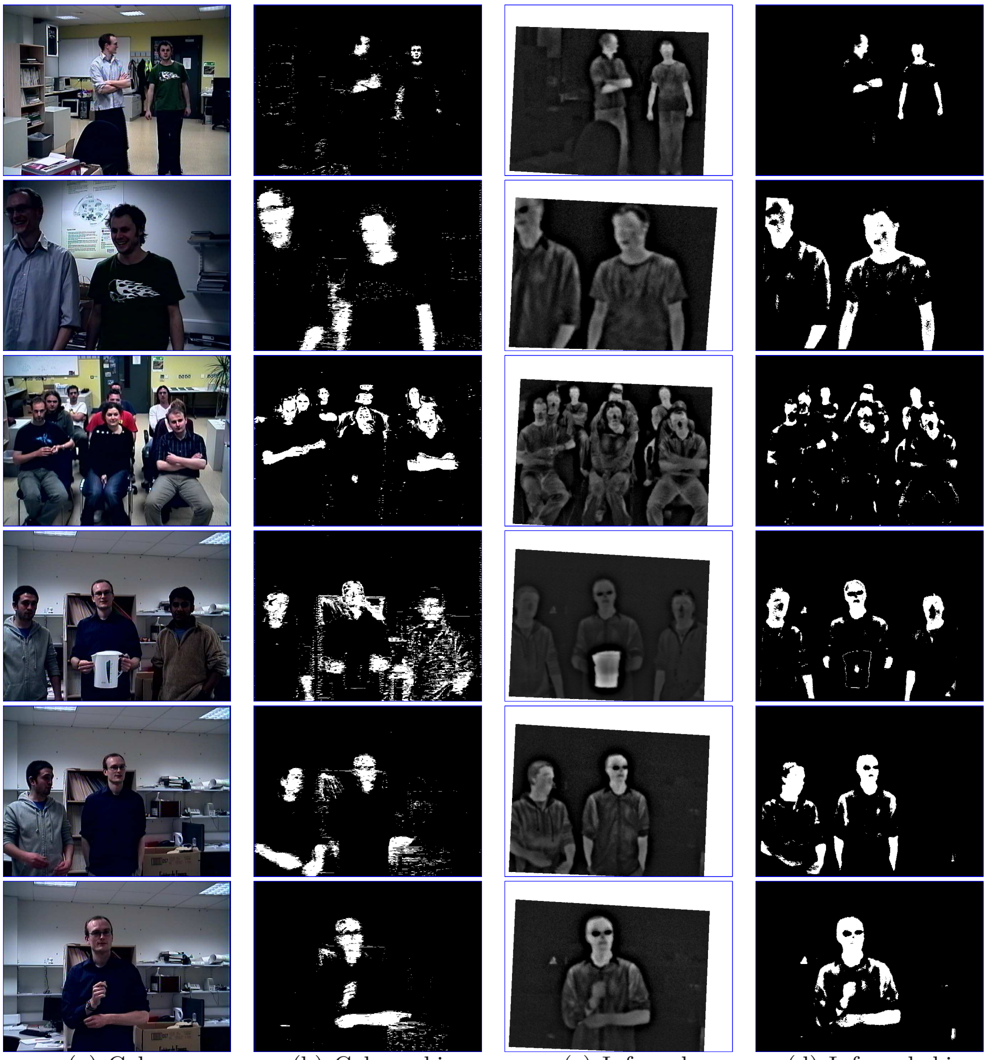


Figure 4.30: Objective comparison of initialisation methods: The x-axis shows the increase in agreement from using init method (a) instead of init method (b) from figure 4.29. The y-axis shows the increase in performance, measure with the F_1 measure. The plot clearly shows that increased agreement leads to increased performance and shows that both init methods have approximately equal performance, with init method (a), based on infrared initialisation, having a slight advantage on this small testing set.



(a) Colour (b) Colour skin (c) Infrared (d) Infrared skin

Figure 4.31: Examples of detected skin in each modality, using both initialisation methods and selecting the parameters with higher agreement.

	Frame	Method 1		Method 2		Both
Seq	count	Iter.	%	Iter.	%	%
A	235	3.42	14.89	3.91	16.60	68.51
B	406	3.63	19.46	4.43	3.20	77.34
C	615	4.05	9.27	4.16	79.67	11.06
D	2984	4.33	83.88	3.86	16.09	0.03
E	306	3.39	0.00	3.61	100.00	0.00
F	997	4.10	47.14	4.09	29.19	23.67
G	1154	3.91	24.78	4.10	12.65	62.57
ALL	6697	3.83	28.49	4.02	36.77	34.74

Table 4.15: Table above indicates the percentage of frames for which each initialisation method converged to the highest agreement score, for all seven sequences tested. The rightmost column indicates that both methods converged to very similar configurations, within a small tolerance.

person, with different skin tone, enters the scene, the lower bound for Hue decreases to adapt the skin model to cater for this. The luminance lower bound is also shown to decrease.

Fusion evaluation After selecting appropriate parameters for the skin models, the resulting output is a pair of binary images, from visible and from infrared, as sources of evidence as to whether or not a pixel is a skin-pixel. These binary masks can be fused for a final classification decision. Five simple fusion schemes were evaluated on 16 ground-truthed skin-detection images. The fusion schemes were (i) binary AND, (ii) binary OR, (iii) Visible only, (iv) IR only and (v) region-based fusion. The region-based scheme examined all the connected-component regions in the binary OR image. If a region had 10% or more of its pixels also belonging to the binary AND image, then it was included. Otherwise, only the pixels in that region from the AND image were used. Although the threshold of 10% is ad-hoc, a range of thresholds were found to perform similarly. The results are given in table 4.16. As expected, the AND fusion achieves very high precision and the OR fusion achieves high recall. Using IR only performs well, compared to visible only, as there were fewer distractors at a similar brightness to skin in the dataset, compared to skin-colour-like distractors in the visual domain. Using the F_1 measure [147] to combine precision and recall, the region based fusion performed best overall.

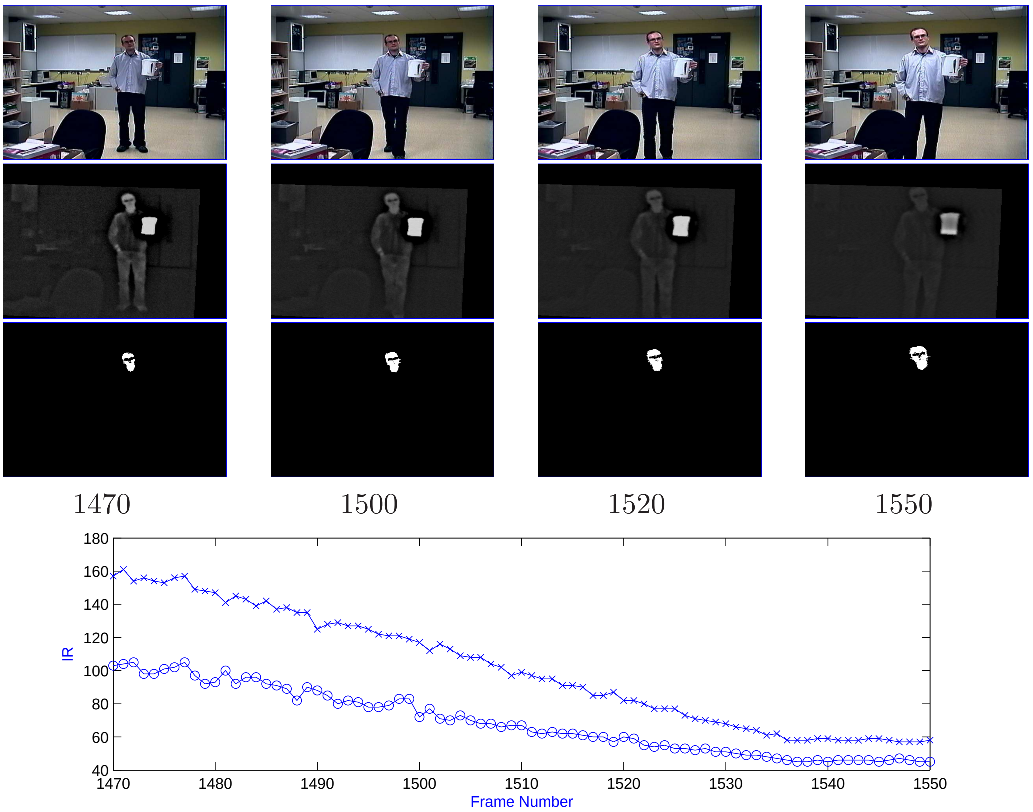


Figure 4.32: Adaptation of the IR bounds to a hot kettle entering the scene: Visible images (top row), Infrared images (2^{nd} row), Detected Skin using Region-based fusion (3^{rd} row) and a plot of the IR bounds $\{l_7,l_8\}$ adapting to the camera's contrast change. Frame numbers shown beneath images.

	AND	OR	VIS	IR	REG
Precision	0.976	0.605	0.641	0.776	0.849
Recall	0.516	0.878	0.664	0.731	0.838
F_1	0.675	0.717	0.652	0.753	0.843

Table 4.16: Binary fusion methods evaluation.

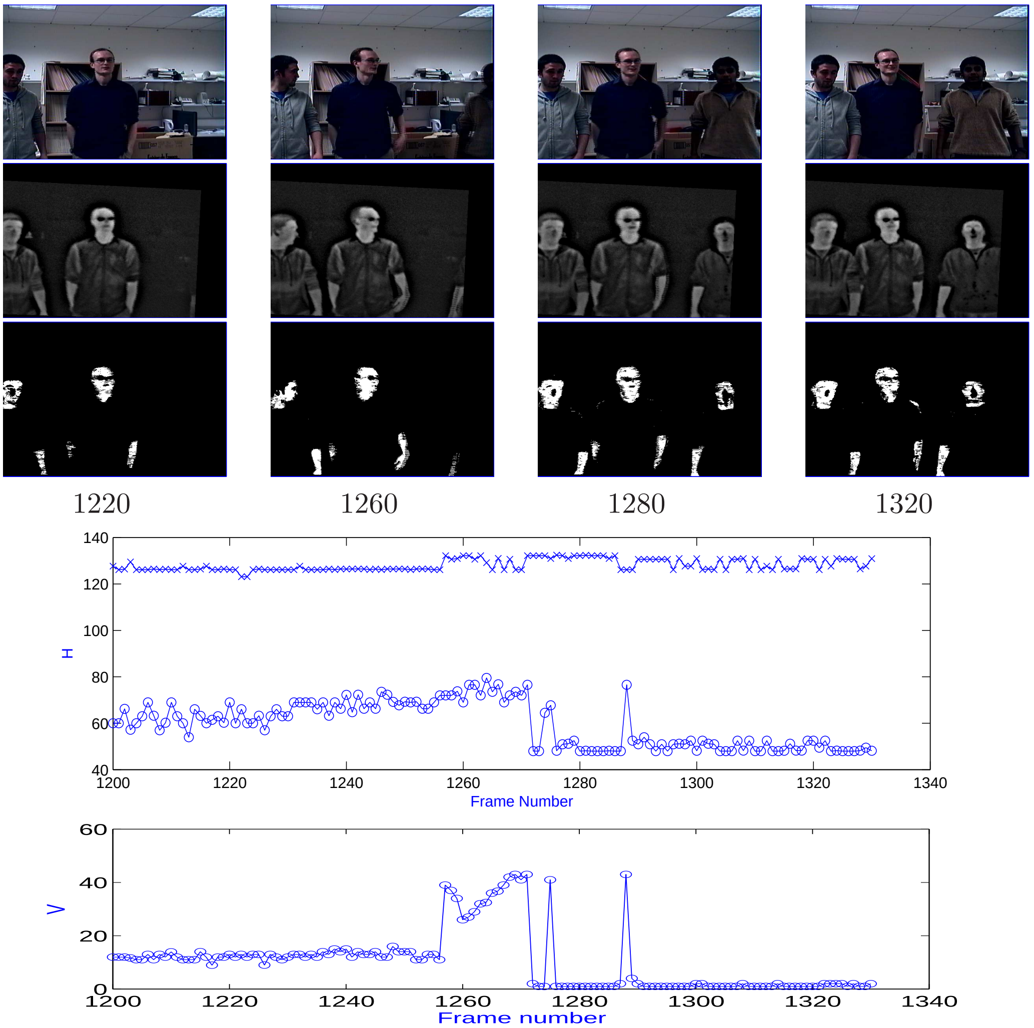
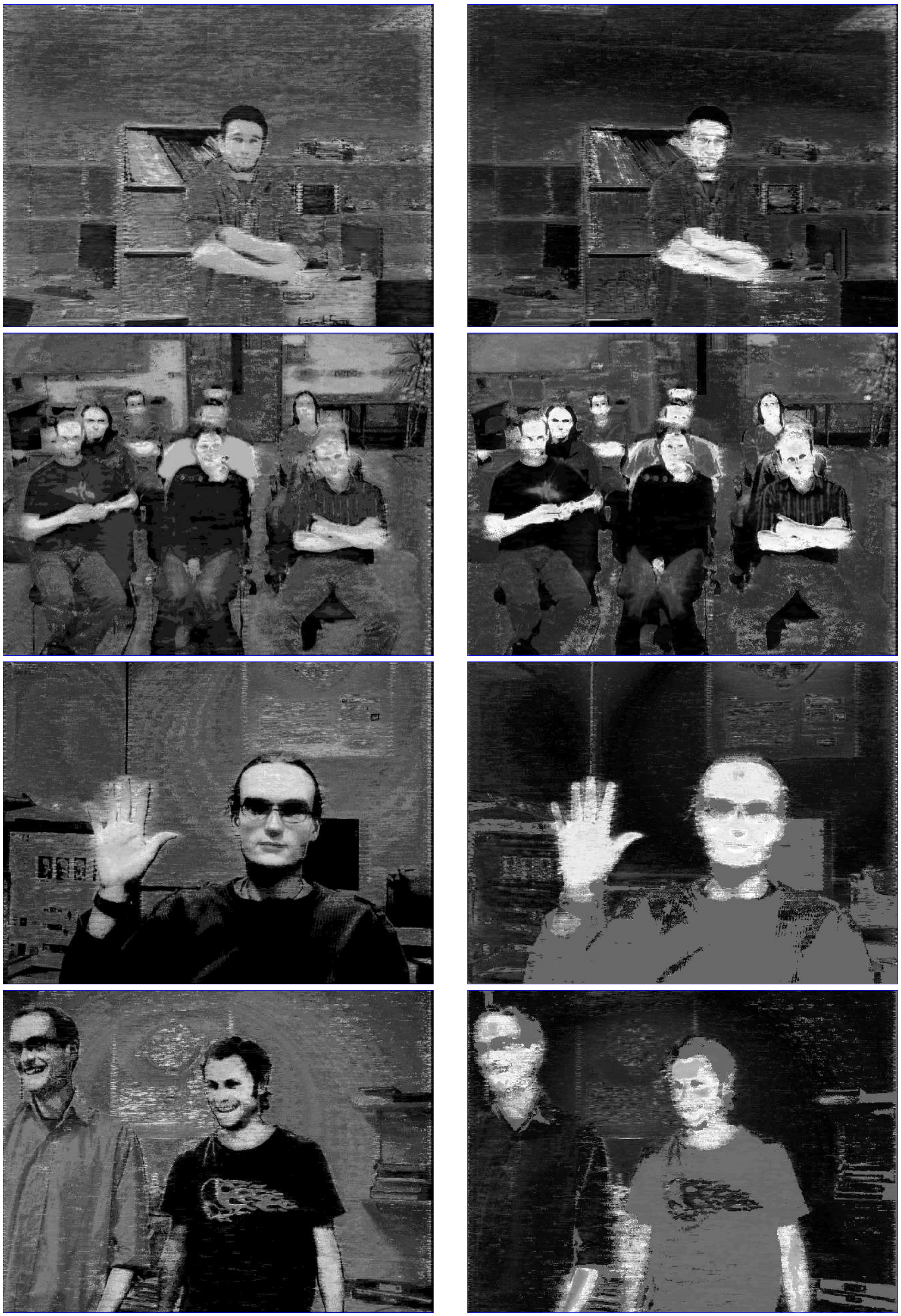


Figure 4.33: Adaptation of the colour bounds to a person with darker skin entering the scene: Visible images (top row), Infrared images (2^{nd} row), Detected Skin using binary AND fusion (3^{rd} row) and a plot of the changing Hue bounds $\{l_1,l_2\}$ and luminance lower bound $\{l_6\}$. Frame numbers shown beneath images.

Adaptive probabilistic model The method described does not exploit any temporal information available in video sequences, but works on individual frames. We now examine how the proposed method can be used to automatically create probabilistic models of skin and background colour appearance and this is compared to a pre-learned human-annotated colour model. Manually annotated skin and background images are available online as part of Sigal et al.'s work on skin segmentation [127]. Using a similar approach to the original work, these samples were used to create $32 \times 32 \times 32$ RGB colour histograms for both skin and background appearance, and these histograms were normalised and used as probabilistic models of the skin and background. For a given colour image, Bayes' rule can be applied and these models create a log-likelihood image, giving each pixel a skin-likelihood value. The pre-trained model was created using 723 images which contained $8,929,954$ skin pixel samples and $129,642,003$ background pixel samples.

The proposed skin and background models were created in a similar fashion but the samples they are trained with were all automatically selected by the adaptive bounding algorithm. For each image in the video sequence, skin pixels are detected by maximising agreement and then performing binary AND fusion to achieve high precision. All these pixels are inserted into the skin model. All pixels which are classified as background by both IR and visible are inserted into the background model (NAND fusion). All other pixels are ambiguous, so are ignored. For each video image tested, 100 of the previous frames are used for training the skin and background models. Figure 4.34 shows examples of the log-likelihood image created by the proposed method versus the pre-trained model. Figure 4.35 shows the ROC curve that indicates the improvement of using adaptive skin modelling over a pre-trained model.

Detection on OTCBVS data Figure 4.36 illustrates the results of using the proposed method on data from the OTCBVS dataset. An approximate manual alignment of the visible and thermal images was performed using a planar homography, but since the cameras are so close to the subject, this model is not always a good fit. The method performs well here. The thermal image shows a few of the disadvantages of thermal images from a BST (ferroelectric) camera, such as the ears appearing colder than the background and hence not being detected.



(a) Pre-trained model (b) Proposed model

Figure 4.34: Examples of log-likelihood images created by (a) the Pre-Trained model and (b) the proposed method

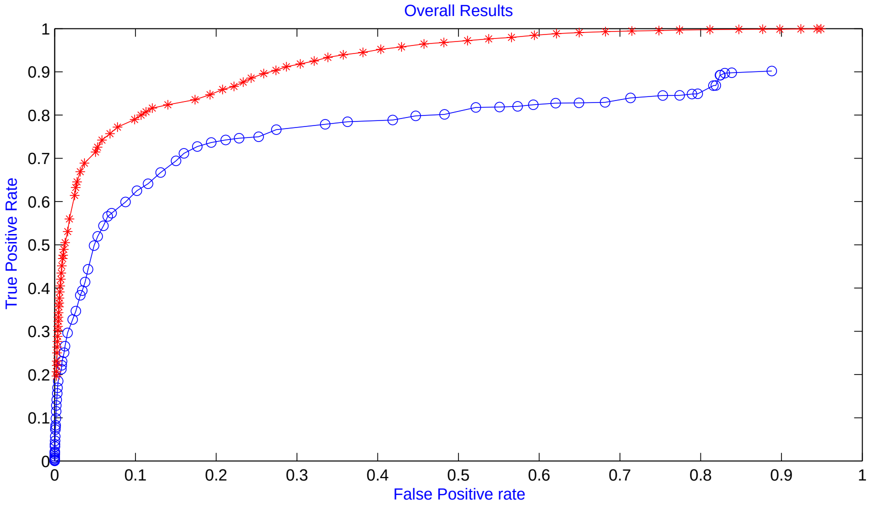
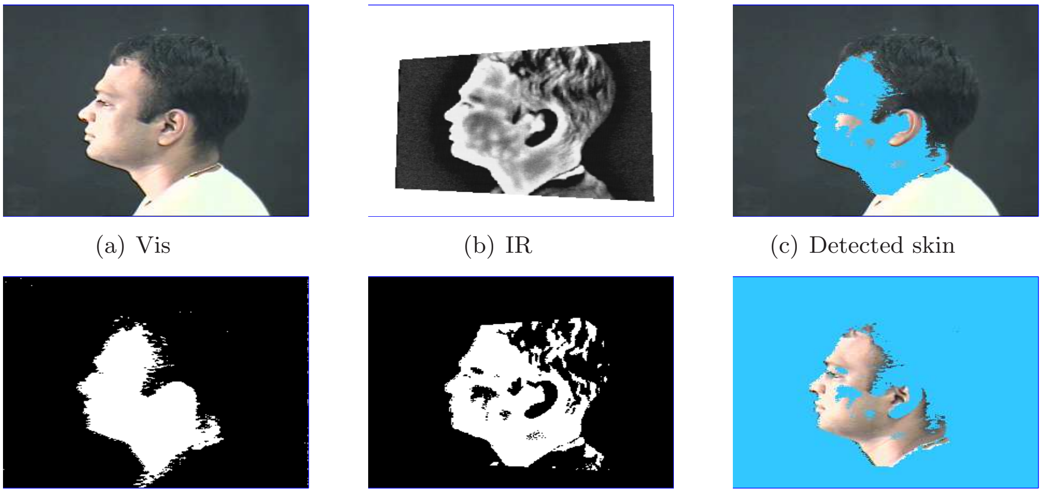


Figure 4.35: ROC curve for the automatically learned probabilistic model (red stars) vs. the pre-trained model (blue circles)


(a) Vis

(b) IR

(c) Detected skin

(d) Bounded HSV.

(e) Bounded IR.

(f) Detected skin

Figure 4.36: Sample results on aligned data from the OTCBVS dataset.

4.4 Discussion

In this chapter, a wide range of applications of the MI thresholding method were demonstrated, using both weakly and strongly independent sources.

4.4.1 Weakly independent sources

In the first three applications that were examined, the assumption of source independence may not have been true in the data. The extended version of the algorithm (EMIT) was tested to see if it could cater for cases where correlated noise caused multiple peaks to appear in the MI surface.

In detecting foreground, it was shown that the EMIT algorithm fared much better than the standard peak selection approach. The Kapur method was the top performing thresholding algorithm overall, but there were cases when the Kapur method would perform poorly, selecting very high thresholds and missing important foreground objects.

In detecting shadow pixels, not only was the assumption made that the sources were independent, but also that shadows caused a decrease in saturation in the background pixels. Despite these ambitious assumptions, the shadow parameters converged to reasonable bounds on images from the Terrascope dataset. In other shadow datasets, the assumption did not hold and there was not enough common information to exploit in the saturation and brightness difference images.

In detecting people in the OTU database of thermal imagery, an additional peak would often appear in the MI surface and the EMIT algorithm performed well in selecting the correct peak. A further modification was required to the EMIT heuristic for peak selection: thresholding the MI surface at 25% of the peak, instead of using Rosin's method. This was required in order to avoid selecting thresholds that were too high. Given this modification, the person detection system based on EMIT outperformed Kapur's method. The proposed algorithm was able to lower its threshold to adapt to people who left only a slight impression on the silhouette map, by exploiting the common information in the contour detection map.

Overall, the use of MI thresholding and related techniques to weakly independent data appeared moderately successful. In shadow detection, successful operation was only demonstrated on one dataset. In other datasets, the assumption of background saturation did not always hold.

The selection of the correct peaks in the MI surface, when multiple peaks occur, is a difficult task. Figure 4.37 illustrates a synthetic example of data sources that have an MI surface with multiple peaks. The data is made up of a series of objects (squares) of progressively greater signal-to-noise ratio (SNR). Selecting the peak in the MI surface results in figure 4.37(e). On the other hand, by selecting the peak with highest quality, as the EMIT algorithm does, the lowest SNR object is dropped, producing figure 4.37(f).

Using maximal mutual information as the threshold selection criterion encourages high *recall*, since it is based on entropy. Using the quality value, on the other hand, encourages high *precision*. The Kapur method, as shown in the extensive tests of this chapter, produces high thresholds, and therefore has a high precision. The EMIT algorithm, by using the quality of the peak, usually sets a high threshold, achieving higher precision than the standard MI thresholding algorithm, so is more like the Kapur algorithm.

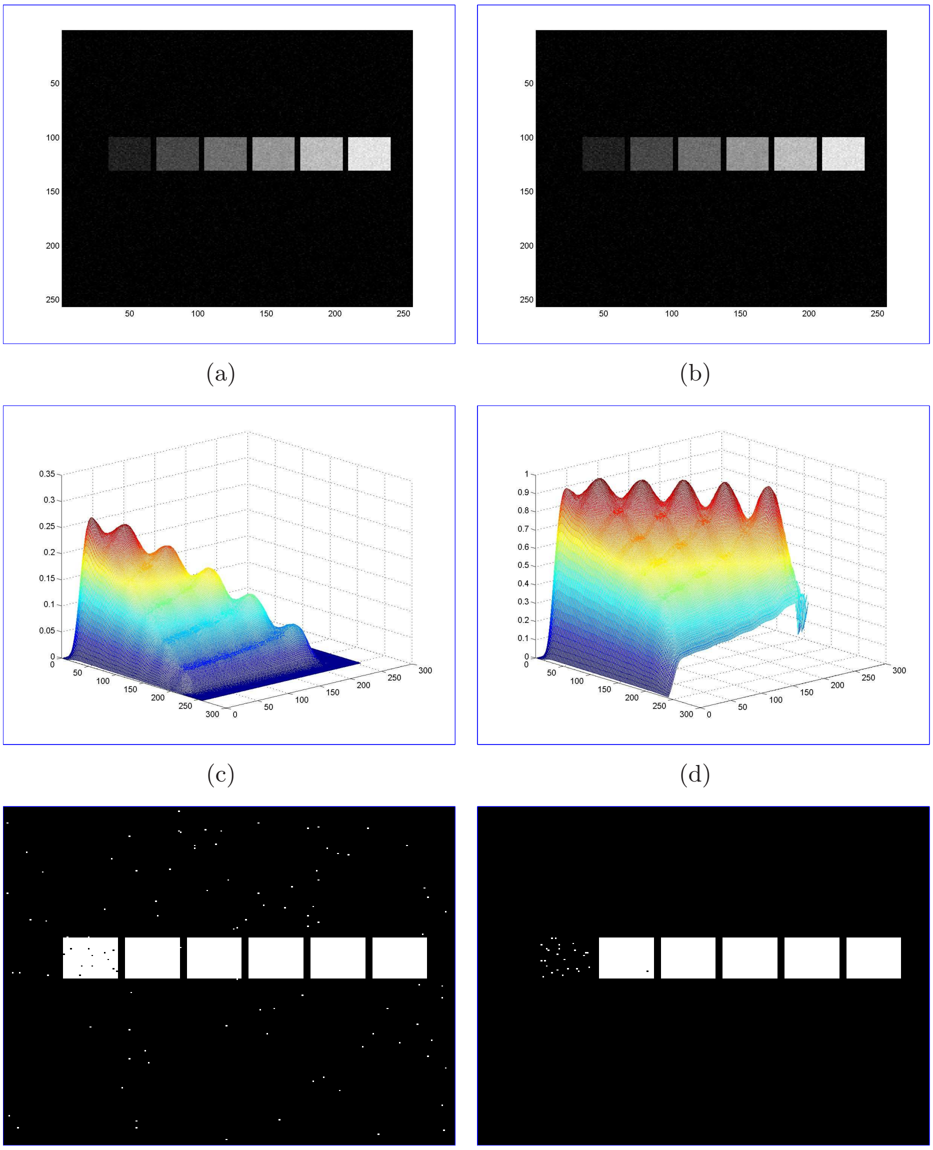
4.4.2 Strongly independent sources

In the second set of applications, the sources used came from disparate sensors, and therefore could be considered strongly independent. The various applications examined how visual information could be fused with audio data and with thermal infrared data, targetting event detection, foreground object detection and skin pixel detection.

Using thermal infrared and visible spectrum background models, it was shown that detection of empty frames was possible using only the MI value between the detected foreground images of both sources. This followed on from the synthetic test of the previous chapter that showed a clear separation between cases of no common information and that of common information being present.

In order to detect foreground pixels, the background models in the thermal infrared and visible spectrum domains were modelled using standard background modelling techniques. The use of MI thresholding in this context was shown to outperform the standard approaches, even when a shadow suppression module was used to counter false positives due to decreases in lighting.

In the penultimate application, nine different visual features were combined with an audio detection signal to detect surveillance events in a corridor. Using the method derived for synthetic data in the previous chapter, the optimal smoothing parameter was determined to exploit the close proximity of correct detections. Before optimal smoothing, the ranking of the nine features from the MI-quality plot was unclear, as the relative ordering of some features was ambiguous. However, after smoothing using the optimal σ value, the MI thresholding method provided a very clear ranking



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.37: Sythetic data with multiple peaks in the MI surface: (a) source 1, (b) source 2, (c) MI surface showing multiple peaks, (d) Quality surface, (e) thresholded result using maximum MI, (f) thresholded result using maximum quality peak.

of the useful visual features for event detection. This was confirmed using a manually annotated ground-truth, indicating a clear correlation between detection usefulness and feature agreement.

Thermal and visual modalities appear well suited to the task of skin detection, as ideal complentary sources of data for this task. A dynamic bounding algorithm was described that allows the efficient search of a high dimensional space to find the optimal agreement parameters for both modalities. By examining two methods of initialisation, it was shown that both methods perform well under different circumstances and by performing the optimisation twice, once with each method, the best parameters can be selected using the set with maximal agreement. A number of simple fusion schemes were evaluated and it was shown that a region-based fusion out-performs either the visible or infrared modality alone. It was further shown that building a skin-colour model adaptively significantly out-performed a pre-trained skin model.

4.4.3 Future work

While this chapter covered numerous applications of MI thresholding, there are many other areas of investigation that could provide worthwhile topics for future research. A number of practical applications where MI thresholding could be of use are mentioned here, as well as some more theoretical aspects that may provide possible extensions to the algorithm.

Edge detection was one area where Rosin's method of thresholding was found to provide good edges by thresholding the gradient magnitude. Another paper by Rosin [117] investigated the use of other edge saliency measures besides the magnitude, such as the edge's lifetime through scale-space, its *wiggliness* and its local contrast. An interesting area of future work lies in using these saliency measures as input to the MI thresholding algorithm and investigating how well the use of these sources helps in edge detection.

Shot boundary detection in digital video is another area where the use of MI thresholding might result in gains in performance. Multiple features, such as colour histograms, MPEG-1 motion vectors and edge-based descriptors have been shown to be useful indicators of a shot change. The adaptation of thresholds and the combination of multiple feature have been shown to be beneficial [13], so the use of the proposed method could be an interesting area of future research.

A single threshold may be too blunt an output for some applications. For example, in [33], using mouth-motion cues to determine if an actor is speaking, high and low thresholds are used. Values above the high threshold and below the low threshold indicate that the actor is speaking and not speaking, respectively, but values between these thresholds are determined to be ambiguous and the detector refuses to give a positive or negative answer. If a pair of thresholds were chosen for each source (4 thresholds in total), so that each 'thresholded' image had three values, $\{yes, no, unknown\}$, then the thresholds could be chosen to maximise MI between the sources. This would be a higher dimensional problem, but might have a dynamic-programming solution that would make the search more efficient.

One may go further, and instead of mapping the sources to 3-value images, they could be mapped to *confidence* values using a 2-parameter sigmoid model. This model is given by:

$$S(x) = \frac{1}{1 + e^{\frac{\mu - x}{\sigma}}} \tag{4.13}$$

The use of a sigmoid is similar to thresholding with a threshold of μ, but it includes a slope parameter, σ, and assigns values between 0 and 1 to samples that are close to the threshold. The parameters of the sigmoid could be selected automatically to maximise agreement between the signals, using the correlation coefficient or rank correlation.

In combining visual and infrared data for skin detection, Kendall's τ was used as an agreement measure instead of mutual information. The similarities and differences between MI surfaces and the Kendall's τ surfaces has not been extensively investigated in this work and could provide insights into which measure is best for different applications. Additionally, a method of combining the two measures could add robustness to system performance, as the correct threshold would be expected to be present as nearby peaks in both surfaces.

In selecting an appropriate scale for signal smoothing, the method proposed in this work used the same smoothing parameter σ for both sources. The value for this parameter could be deduced from the MI and quality value graphs. If a separate σ value were to be used for each source, this would result in a surface, relating a pair of smoothing parameters to the resulting MI and quality values. A question for future research is how appropriate smoothing parameters should be selected by using these surfaces.

While the use of data source smoothing did, to some extent, exploit the spatial relations between samples, there is more work to be done in this regard. The smoothing of thermal and visual images in section 4.3.1 lost the individual identities of the objects and this is not desirable. By creating a parametric model for each source that incorporates spatial and brightness information, the parameters for this model could be set by maximising agreement: either mutual information or another agreement measure. A simple approach might be to use hysteresis thresholding, as mentioned in the conclusion of the previous chapter, where the low and high thresholds are selected by maximising the agreement between the resulting segmentations. While the proposed method examines single pixels, using their values in both sources, spatial information might also be included by looking at pairs of neighbouring pixels and using their values in both sources. There might easily be a dynamic-programming solution for this paradigm that would efficiently select thresholds to maximise this spatial agreement between the data sources.

Having extensively investigated the use of agreement-based threshold selection for combining multiple data sources, the remainder of this thesis now examines a different but related issue, namely that of visual tracking using multiple sources of information. Similarly to the work described in the previous and current chapters, a general framework for fusion is proposed and extensively evaluated. While the proposed framework is general in nature, the main focus of the described work is on fusing visual and thermal infrared data. Unlike the current chapter where a series of varied application for the MI thresholding method were studied, the next two chapters exclusively target the application of visual tracking. The next chapter introduces the proposed tracking framework, which uses a bank of spatiograms, and compares its performance to traditional tracking approaches. The following chapter builds on this work and extends the tracker, allowing it to adapt its tracking approach in difficult scenarios.

Chapter 5

Spatiogram Fusion

5.1 Introduction

In the two previous chapters, we examined how measures of agreement between sources could be used to aid in detecting events and objects that were common to both sources, by adaptively adjusting their parameters to maximise agreement. Traditionally, the object detection phase of a vision system is complemented by a tracking phase. Once an object has been detected, it is passed to the tracker to update the object's position from frame-to-frame. While the object detection step could be run in each frame, it may be computationally expensive and therefore may only be executed sporadically. In fact, the object detection step may not be guaranteed to find the object every time, so would generate a fragmented track. The object detector is typically tuned to detect a wide range of objects. The tracker, on the other hand, maintains a model of one specific object's appearance, so should rapidly and accurately locate that specific object in the next frame.

5.1.1 Overview

In this chapter, the contribution of this thesis to object tracking using spatiograms to fuse information from multimodal sources is described. Firstly, a brief review of object descriptors that have commonly been used for tracking is conducted. Next, histogram-based tracking is discussed, in order to introduce the concept of spatiograms and their use in tracking. The limitations of histogram- and spatiogram-based tracking are discussed and a new framework for tracking, termed a *Spatiogram Bank Tracker*, is proposed to overcome these limitations. In this framework, the features used for tracking are split over multiple Spatiograms. The validity of the assumptions that underpin this approach are discussed. We derive a mean-shift algorithm for this framework, allowing efficient object localisation. Finally, a number of experiments illustrating the advantages of this tracking framework are described.

5.2 Related research

Many descriptors have been proposed for object modelling in order to provide robust tracking in video sequences. Histograms [19, 22, 105, 106, 163] have commonly been used for tracking, as they discard spatial information and are therefore insensitive to changes in pose of deformable objects. At the other extreme, image templates [86], which impose rigid spatial constraints on feature layouts, have also been used. In [31], Elgammal et al. provide a parametric feature-spatial distribution model for object modelling. A spatial kernel bandwidth parameter must be supplied, to tradeoff between rigid spatial constraints and no spatial constraints. Another object descriptor that has been proposed is the Spatiogram [11], which generalises the histogram to include spatial information by allowing higher-order spatial moments to be part of the descriptor. The spatiogram can be thought of as lying somewhere along the axis connecting histograms and templates. This axis also contains descriptors such as the aforementioned feature-spatial distribution model [31] and SIFT models [81]. The spatiogram's geometric model bridges the gap between histograms, which allow for arbitrary transformations, and more specific models of feature position deformation, such as affine, projective and B-splines. Like histograms, spatiograms enable comparison between image patches without specifically computing a geometric transformation between them, but like the more specific models, spatiograms retain some information about the geometry of the patches.

The work of [31] is most similar to the work described in this chapter but differs in a number of ways. Firstly, to compare two spatial-feature distributions as they propose, all pairs of samples must be compared which will considerably reduce tracking speed. The use of the fast Gauss transform has been proposed to address this but it is expected that the Spatiogram-bank tracker, proposed in this work, will still have a computational advantage. Secondly, a parameter that controls the importance of spatial information must be tuned. Finally, there is no obvious extension to the work described in [31] that would allow the dynamic weighting of features to adapt to different tracking scenarios. Adaptive tracking is shown in the next chapter to be beneficial for robust tracking.

5.3 Review of spatiograms

In this section, the concept of a spatiogram is explained intuitively and mathematically, and its usefulness in object tracking is discussed. Spatiograms were first introduced in 2005 in [11] as a generalisation of the common histogram, which has frequently been used in tracking applications [22, 105, 106]. We first examine the use of histograms in object tracking and then proceed to discuss spatiograms.

5.3.1 Histograms

A histogram is a normalised count of the number of times a feature falls into a specified range of values. The normalised count of bin b for the target object can be computed as follows:

$$n_b^{'} = C \sum_{i=1}^{N} k(||x_i||^2)\delta_{ib} \tag{5.1}$$

where N is the number of pixels, $\delta_{ib} = 1$ if the i^{th} pixel falls in the b^{th} bin and $\delta_{ib} = 0$ otherwise, C is a normalising constant that ensures the n_b values sum to one, $x_i = [\mathtt{x},\ \mathtt{y}]^T$ is the spatial position of the i^{th} pixel, k is a smoothing kernel, which weights pixels that are closer to the centre, reducing the effect of background pixels. It also has the effect of smoothing the similarity surface. Epanechnikov or Gaussian kernels are commonly used [23].

To evaluate a matching candidate of size h, containing N_h pixels, at location y, its histogram is computed as follows, with C_h performing a similar normalisation function to C:

$$n_b(y) = C_h \sum_{i=1}^{N_h} k(||(x_i - y)/h||^2)\delta_{ib} \tag{5.2}$$

A target and candidate histogram with B bins each can be compared using the Bhattacharyya coefficient [23], which is the most commonly used measure in histogram-based object tracking:

$$\rho(y) = \sum_{b=1}^{B} \sqrt{n_b(y)n_b^{'}} \tag{5.3}$$

5.3.2 Spatiograms

Spatiograms [11] are a generalisation of the common histogram, capturing not only the number of occurances of particular feature values, but also additional spatial moment information. Formally, if $f(x)$ describes the feature value at position x, where $x \in \chi$, the range of positions being evaluated, we let

$$h_f^{(i)}(v) = \sum_{x \in \chi} x^i \delta(x, v) \tag{5.4}$$

where $\delta(x, v) = 1$ if the feature value, $f(x)$, falls into the v^{th} bin, and $\delta(x, v) = 0$ otherwise. Notice that when $i = 0$, the values of $h_f^{(0)}(v)$ are simple histogram bin-counts. A k^{th}-order spatiogram is defined as a tuple of all the moments up to order k: $< h_f^{(0)}(v), h_f^{(1)}(v), ..., h_f^{(k)}(v) >$. In the work considered in this thesis, 2^{nd}-order spatiograms are used to model object feature distributions. This is equivalent to storing a spatial mean and variance with each histogram bin. Investigating the use of higher-order spatiogram is left for future work. The spatial mean and covariance of each bin for a 2^{nd}-order spatiogram are computed as follows:

$$\mu_b(y) = \frac{1}{\sum_{j=1}^{N_h} \delta_{jb}} \sum_{i=1}^{N_h} (x_i - y)\delta_{ib} \tag{5.5}$$

$$\Sigma_b(y) = \frac{1}{\sum_{j=1}^{N_h} \delta_{jb}} \sum_{i=1}^{N_h} (x_i - \mu_b(y))^T (x_i - \mu_b(y))\delta_{ib} \tag{5.6}$$

where, as before, N_h is the number of pixels in the region, y is the position of the region centre and x_i is the spatial position of the i^{th} pixel. The spatial coordinates are scaled to lie between -1 and $+1$ to normalise spatiograms to handle different region sizes. The spatial distribution of each bin b is modelled as a Gaussian with the mean and covariance given above. Figure 5.1 demonstrates the additional information that spatiograms contain when compared to histograms.

To compare two spatiograms, the following *Bhattacharyya-like* similarity measure is the one used in the original Spatiogram work [11]:

$$\rho(y) = \sum_{b=1}^{B} \psi_b(y) \sqrt{n_b(y) n_b'} \tag{5.7}$$

where $\psi_b(y)$ is the spatial similarity measure, given by:

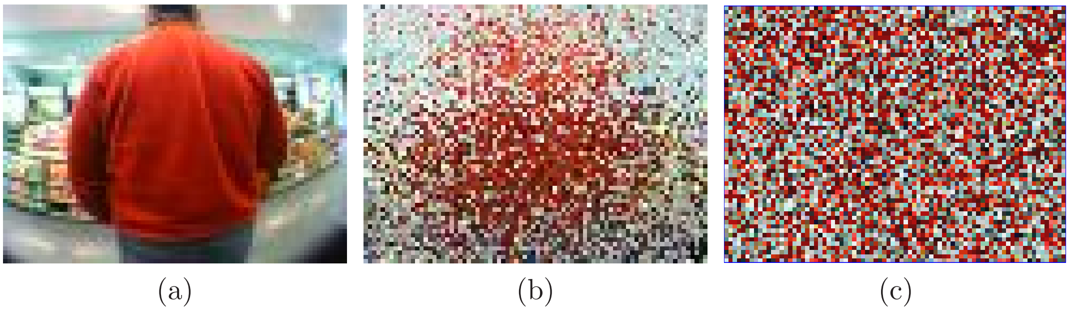

(a) (b) (c)

Figure 5.1: Illustration of the information contained in a spatiogram versus a histogram: Original image shown in (a), an approximation of (a) generated by randomly selecting pixels from the probability distribution of its 8×8×8 bin spatiogram (b) and histogram (c)

$$\psi_b(y) = \eta exp\left\{-\frac{1}{2}(\mu_b(y) - \mu_b^{'})^T \hat{\Sigma}_b^{-1}(y)(\mu_b(y) - \mu_b^{'})\right\} \tag{5.8}$$

where $\hat{\Sigma}_b^{-1}(y) = (\Sigma_b^{-1}(y) + (\Sigma_b^{'})^{-1})$, so that the distance between the spatial means is normalised to the average of the two Mahalanobis distances and η is the Gaussian normalisation constant. In order to ensure that each Σ_b is invertible, they are assumed to be diagonal, and a minimum variance value is set to one pixel. This measure gives high similarity scores to spatiograms whose histogram bins counts are similar and whose spatial means are aligned.

5.4 Spatiogram-Bank Tracker

5.4.1 Limitations of spatiograms

In the context of combining object features for tracking, there are two main drawbacks to using histograms or spatiograms. Firstly, their memory requirements (and hence their computational load) increase exponentially as more features are added and secondly, they do not scale well to higher dimensions due to the *curse of dimensionality* [8].

As an example of the first drawback, an RGB colour histogram with 32 bins per channel requires a total of $32^3 = 32768$ bins. If an extra channel, such as thermal infrared, is added, this increases to $32^4 = 1048576$, which increases the memory requirements and decreases the tracking speed due to increased computation. The second drawback concerns the *curse of dimensionality* [9] which states that it is more difficult to accurately estimate feature distributions for higher dimensional spaces, since exponentially more samples are required. It has also been shown that the Bhattacharyya coefficient, often used in tracking to measure similarity between histogram distributions, is not very discriminative in higher dimensions [157]. To overcome these difficulties, tracking can be achieved by splitting the feature-set over several histogram trackers and combining their outputs. For example, instead of using a K dimensional histogram, K one-dimensional histograms could be used and their outputs combined, which is equivalent to using K separate trackers. This substantially reduces the memory and computational requirements, and also allows the use of parallel processing to further speed up tracking. Unfortunately in the case of histograms, it is not theoretically justifiable to separate features in this way, as the assumption of independence for marginal histograms does not hold, as now illustrated.

The images in figure 5.2(a) are significantly different and can clearly be distinguished by their joint red-green histograms, shown in (b). The drawback of joint-distributions, as mentioned earlier, is that they require exponentially more memory as more features are added, as well as suffering from the *curse of dimensionality.* Using marginal histograms as an approximation for the joint-histogram is not a valid solution, since both images have identical marginal distributions in both the red and green bands, therefore cannot be distinguished if the features are separated (see figure 5.2(c)). In the next subsection, instead of using marginal histograms for tracking, it is argued that the use of marginal spatiograms is *more valid* for this propose.

5.4.2 Proposed framework

In [11], the spatiogram is proposed as a more accurate model for object tracking than histograms. The work in this thesis proposes to make spatiogram tracking more efficient and suitable for multimodal data fusion by splitting the features over multiple separate spatiograms. The tracking framework proposed is illustrated in figure 5.3, where the pixel-based features used to track the target object are split over N spatiogram model trackers. All trackers evaluate a series of potential object position hypotheses and return a similarity score for each one. The combined score for each hypothesis is computed by multiplying the similarity scores from each tracker. Formally, the combined score is written:

$$\rho(y) = \prod_{k=1}^{K} \rho^{(k)}(y) \tag{5.9}$$

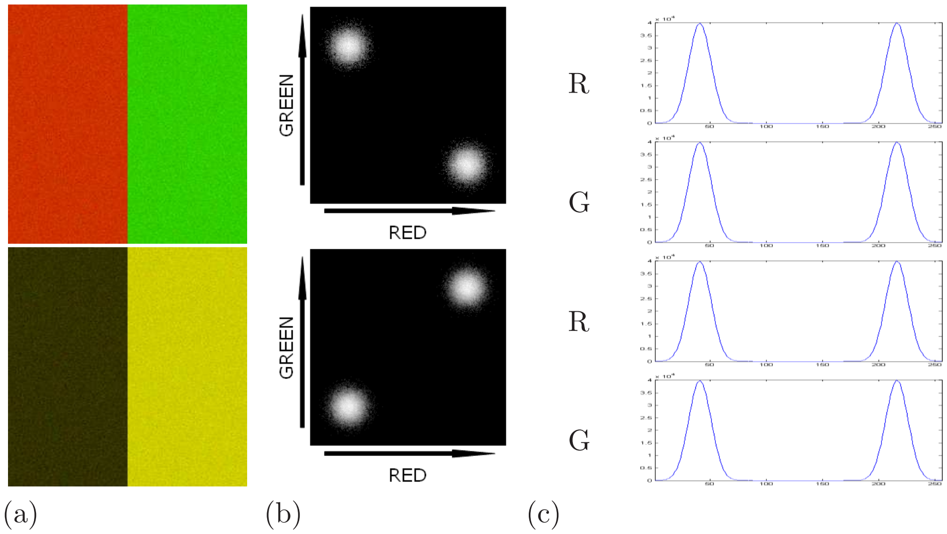


Figure 5.2: Illustrating the invalidity of marginal histograms: (a) Synthetic images, (b) their associated joint histograms shown in log scale that clearly allow the images to be distinguished, and (c) the marginal histograms of their red and green bands (both images have identical marginal histograms therefore cannot be distinguished if the features are assumed to be independent)

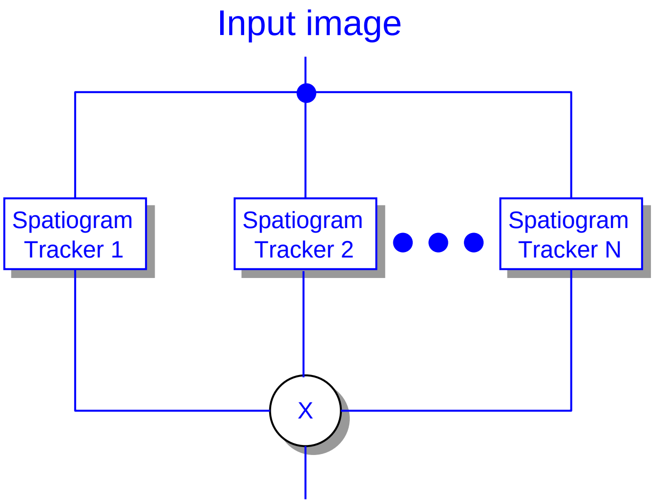


Figure 5.3: Bank of spatiograms framework

where $\rho^{(k)}(y)$ is the similarity score, returned by the k^{th} spatiogram tracker, between the model and the candidate at position y. This tracking framework is adopted for object feature fusion for a number of reasons.

Firstly, the increase in memory and processing requirements is linear with respect to the number of features used (unlike the exponential increase associated with typical histograms and spatiograms) and it does not suffer from the *curse of dimensionality* in accurately estimating feature distributions. The framework allows features to be arbitrarily divided between the K trackers. In our experiments, one tracker is used per feature, but one could combine an RGB spatiogram tracker with an infrared brightness tracker, for example. Unlike template matching, spatiograms trackers do not impose rigid spatial constraints. Instead, the small amount of stored spatial information allows more general object deformations. Also, the tracking framework used can incorporate a mean-shift approach to object localisation, allowing rapid object tracking (described in section 5.5).

Secondly, this framework draws on previously reported work in evaluating various fusion schemes for object tracking [98], where it was found that multiplying similarity scores outperformed simple addition, weighted sums and non-linear score fusion schemes, such as *min* and *max*. If one considers the similarity metric as a probability, multiplying scores is equivalent to assuming the features used by the trackers are independent. The metric used to compare spatiograms is very similar to the Bhattacharyya coefficient, which itself is closely related to the probability of *Bayes error* [7]. In [75], Leichter et al. propose a general framework for tracker fusion by computing a combined probability density function (PDF) by multiplying the PDFs of all trackers (assuming a uniform prior) and our framework can be interpreted as conforming to this general framework. If all spatiogram trackers in our framework perform an exhaustive search in a local search window by computing similarity scores for each location, these scores can be multiplied by a constant without affecting the final combined tracking result. If the constant is chosen so that the scores are normalised to sum to one, then the similarity scores essentially form a PDF which is then multiplied to produce the final combined PDF, hence the similarity to Leichter's framework.

Thirdly, by separating the features, instead of integrating them into one tracker, we provide a flexible architecture for feature addition, removal or weighting, allowing the combined tracker to adapt under different circumstances. This has been shown to benefit tracking in changing environments [8, 20] and is explored in the next chapter on adaptive tracking. In terms of the limited spatial information stored, modelling of each feature bin as a Gaussian may seem restrictive, but in fact captures some useful general spatial distribution properties, as discussed below.

5.4.3 Modelling marginal spatiograms

A model is now derived for the assumed feature distribution when marginal spatiograms are used instead of the full spatiogram.

Spatial distribution probabilities Histograms and spatiograms imply probability distributions of feature values. In the case of the histogram, there is no spatial dependency, so $p(\mathbf{x}, b) = p(b) = n_b$. For spatiograms, we have $p(\mathbf{x}, b) = p(b)p(\mathbf{x}|b) = n_b\phi_b(\mathbf{x})$, where $\phi_b(\mathbf{x})$ denotes the spatial Gaussian model of bin b. Given a particular location, x, the probability of occurrence for each feature bin is computed as:

$$\begin{aligned} p(b|\mathbf{x}) &= \frac{p(\mathbf{x}|b)p(b)}{p(\mathbf{x})} \\ &= \frac{n_b\phi_b(\mathbf{x})}{\sum_{i=1}^{B} n_i\phi_i(\mathbf{x})} \end{aligned}$$

Since $\phi_b(x)$ and each $\phi_i(x)$ are all Gaussians, this shows that the actual spatial distribution of feature value v is a Gaussian divided by a sum of Gaussians. This distribution can be multimodal and is therefore more flexible than the simple Gaussian distribution model would imply.

Fusion of multiple spatiogram models When marginal histograms are used to approximate the full joint-histogram distribution, in the two-band case we obtain $n_{a,b} = p(a, b) = p_1(a)p_2(b) = n_a^{(1)} n_b^{(2)}$. The approximation that is used when marginal spatiograms are used instead of the full joint-spatiogram is now derived. The case where each pixel has two features (hue and saturation, for example) is examined and it can then be generalised to multiple features. For a particular location, x, the spatial distribution of pixels that belong to bin a of feature z_1 and b of feature z_2 is given by:

$$\begin{aligned} p(a, b|\mathbf{x}) &= p(a|\mathbf{x})p(b|\mathbf{x}) && (5.10) \\ &= \left(\frac{p_1(\mathbf{x}|a)p_1(a)}{p_1(\mathbf{x})}\right)\left(\frac{p_2(\mathbf{x}|b)p_2(b)}{p_2(\mathbf{x})}\right) && (5.11) \\ &= \frac{p_1(a)p_2(b)p_1(\mathbf{x}|a)p_2(\mathbf{x}|b)}{\left[\int_{B_1} p_1(\mathbf{x}|i)p_1(i)\right]\left[\int_{B_2} p_2(\mathbf{x}|j)p_2(j)\right]} && (5.12) \end{aligned}$$

$$= \frac{n_a^{(1)} n_b^{(2)} \phi_a^{(1)}(\mathbf{x}) \phi_b^{(2)}(\mathbf{x})}{\left[\sum_{i=1}^{B_1} \phi_i^{(1)}(\mathbf{x}) n_i^{(1)}\right] \left[\sum_{j=1}^{B_2} \phi_j^{(2)}(\mathbf{x}) n_j^{(2)}\right]} \tag{5.13}$$

where p_1 and p_2 refer to the probabilities obtained from the spatiogram model of the first and second feature (z_1 and z_2). This expression can be simplified by noting that the product of two normalised Gaussians, with means q and r, and covariances, Q and R, is a normalised Gaussian multiplied by a constant term:

$$N(x; q, Q) N(x; r, R) = z N(x; c, C) \tag{5.14}$$

with

$$C = (Q^{-1} + R^{-1})^{-1} \tag{5.15}$$

$$c = C(Q^{-1} q + R^{-1} r) \tag{5.16}$$

and the constant term, z, given by:

$$z = N(q; r, Q + R) = \frac{1}{(2\pi)^{m/2} |Q + R|^{1/2}} \; e^{\left(-\frac{1}{2}(q-r)^T (Q+R)^{-1} (q-r)\right)} \tag{5.17}$$

where m is the number of dimensions (2 in this case). Now equation (5.13) can be rewritten as a Gaussian divided by a weighted sum of Gaussians, since all the ϕ terms are Gaussians. If we write:

$$\phi_a^{(1)}(\mathbf{x}) \phi_b^{(2)}(\mathbf{x}) = z_{a,b} \phi_{a,b}(\mathbf{x}) \tag{5.18}$$

And let

$$z'_{a,b} = \frac{z_{a,b} n_i^{(1)} n_j^{(2)}}{\sum_{i=1}^{B_1} \sum_{j=1}^{B_2} z_{i,j} n_i^{(1)} n_j^{(2)}} \tag{5.19}$$

Then we can rewrite equation (5.13) as

$$p(a, b|\mathbf{x}) = \frac{n_a^{(1)} n_b^{(2)} z_{a,b} \phi_{a,b}(\mathbf{x})}{\sum_{i=1}^{B_1} \left[\sum_{j=1}^{B_2} n_i^{(1)} n_j^{(2)} z_{i,j} \phi_{i,j}(\mathbf{x})\right]}$$

$$= \frac{z'_{a,b} \phi_{a,b}(\mathbf{x})}{\sum_{i=1}^{B_1} \left[\sum_{j=1}^{B_2} z'_{i,j} \phi_{i,j}(\mathbf{x})\right]}$$

Firstly, this shows that the spatial distribution of features, in the case of fusion multiple spatiograms, is a Gaussian divided by a weighted sum of B Gaussians, $B = B_1 B_2$, as is the case for a single spatiogram. Therefore, the approximation of the joint-distribution obtained by using marginal spatiograms is itself a spatiogram. Secondly, this spatiogram is given by $\bar{n}_{a,b} = z'_{a,b}$, with $\bar{\mu}_{a,b}$ and $\bar{\Sigma}_{a,b}$ given by equations (5.16) and (5.15). It is similar to the histogram approximation, but adds more weight to joint-feature bins whose marginals have significant overlap in their spatial layout.

5.4.4 Validity of separate spatiograms

In figure 5.2, it was clearly shown that the use of marginal histograms as an approximation for the full joint-histogram provides a poor object model, since the assumption of independence does not hold. Using marginal spatiograms instead, a model of the resulting joint histogram approximation was derived and it was found that it differs from the marginal histogram approximation since it adds more weight to joint-feature bins whose marginals have significant overlap in their spatial layout. Whether this leads to a more valid model of a tracked object is now investigated using illustrative examples.

Using a spatiogram model as a probability distribution, object image approximations can be generated by sampling from the distribution. This illustrates the information contained in these object model descriptors. Figure 5.4 shows examples comparing the image approximations generated by histograms, marginal histograms, spatiograms and marginal spatiograms. As can be seen by comparing rows (b) and (c) in the figure, the approximation images generated from marginal histograms are significantly different from those generated from the full histogram, creating many false pixel colours that did not exist in the original. Due to the added spatial information, the marginal spatiograms provide a good approximation to the full spatiograms, as evidenced by comparing rows (d) and (e) of figure 5.4. Using a decorrelated colour-space, such as YUV, the approximation is improved further, as shown in row (f).

The validity of this approach is not true in general of all objects, as a significant amount of information is lost by using marginal spatiograms instead of the full joint-spatiogram. As seen in figure 5.5, it is possible in some cases for marginal spatiograms to produce approximations of the joint distribution that are just as bad as marginal histograms. In both examples, the feature bins have very large (non-Gaussian) spatial variances and therefore the marginal spatiograms cannot exploit any spatial correlation between the bins. The second example might be better approximated if the full covariance matrix was used, instead of making it diagonal. Clearly, marginal spatiogram independence is not always well justified. However, it is argued in this thesis that the approximation of the tracked object by marginal spatiograms is a more valid assumption than using marginal histograms, as the generated examples of figure 5.4 illustrate. Although there are cases when using marginal spatiograms performs just as poorly as marginal histograms, the most important consideration in using marginal spatiograms for tracking is not whether they provide a good approximation of the full joint-spatiogram, but whether this approximation of the model allows robust discrimination of the tracked object from the background clutter and other objects.

5.5 Spatiogram Bank Mean-shift derivation

Mean-shift [21] is an iterative kernel-based procedure to locate the local mode in a distribution. It has been successfully used in many tracking applications [20, 22, 163] to efficiently locate objects in subsequent frames under the assumption that the object overlaps itself in consecutive frames. For fast-moving objects and low frame-rate video, where this assumption may not be valid, multiple kernels can be used [106].

The mean-shift derivation in this section is motivated by and follows the general procedure presented in [11] where the mean-shift procedure was derived for a single (possibly high-dimensional) spatiogram. The novel aspect of the work in this section is to derive the procedure for a bank of (low-dimensional) spatiograms, thereby avoiding the curse of dimensionality, lowering the computational cost and providing a convenient framework for adaptive feature weighting.

To initiate the iterative mean-shift scheme using the proposed tracking framework, each tracker is first given an object position hypothesis, which is generally equal to its position in the previous frame or a prediction of its current location, based on a velocity estimate, for example. Using the similarity measures returned by each tracker, along with the pixel features and spatiogram models, mean-shift performs gradient ascent on the similarity surface and computes a new object position hypothesis. This procedure is iterated until convergence.

The combined similarity measure, $\rho(y)$, is expressed as the product of all K individual tracker similarities, first examining a simple two tracker system where $\rho(y) = \rho^{(1)}(y)\rho^{(2)}(y)$ and generalising later to handle K trackers. So assuming $K = 2$, we perform a Taylor series expansion around $\rho(y)$ at y_0, and obtain

$$\begin{aligned}\rho(y) \approx \rho(y_0) \quad & +[n^{(1)}(y) - n^{(1)}(y_0)]^T \tfrac{\partial\rho}{\partial n^{(1)}}(y_0) \\ & +[n^{(2)}(y) - n^{(2)}(y_0)]^T \tfrac{\partial\rho}{\partial n^{(2)}}(y_0)\end{aligned}$$

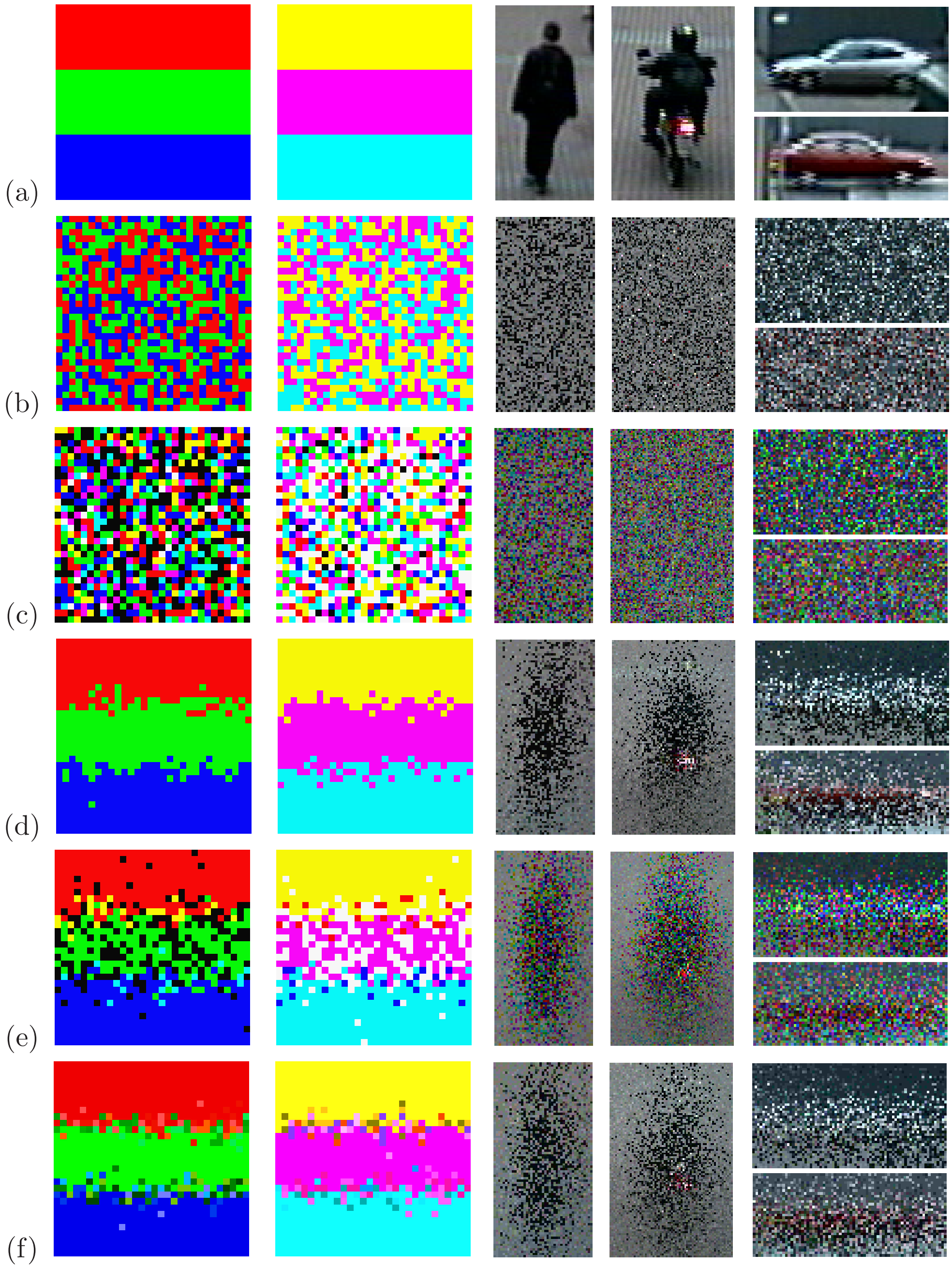
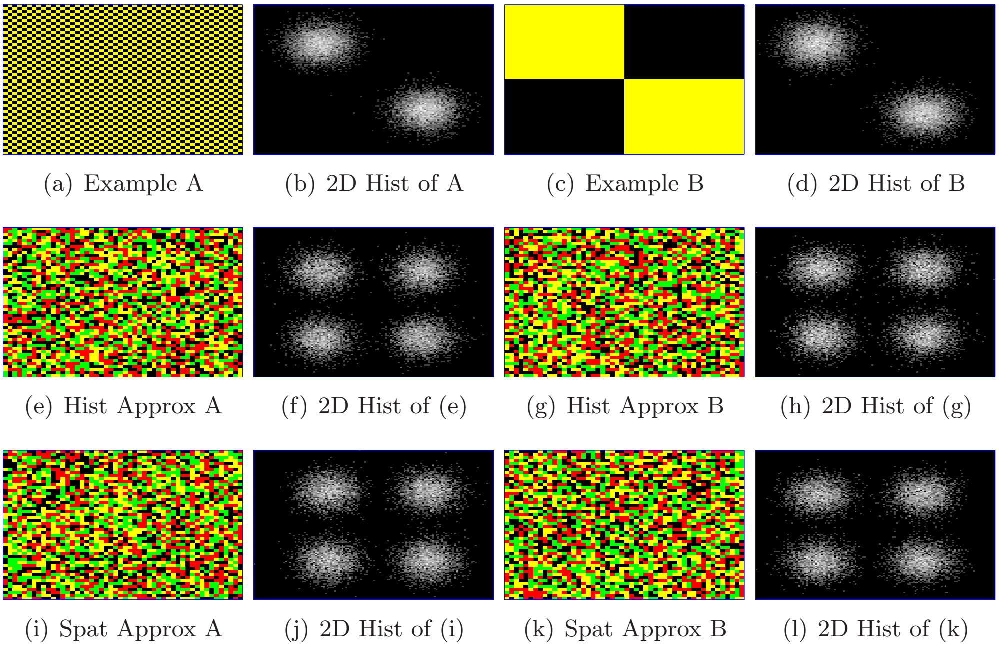


Figure 5.4: Example object model images generated by sampling the distributions of the histogram and spatiogram models: (a) Original image, object model images generated by (b) Full RGB histogram, (c) Marginal RGB histograms, (d) Full RGB Spatiogram, (e) Marginal RGB Spatiograms, (f) Marginal YUV Spatiograms.


(a) Example A

(b) 2D Hist of A

(c) Example B

(d) 2D Hist of B

(e) Hist Approx A

(f) 2D Hist of (e)

(g) Hist Approx B

(h) 2D Hist of (g)

(i) Spat Approx A

(j) 2D Hist of (i)

(k) Spat Approx B

(l) 2D Hist of (k)

Figure 5.5: Marginal spatiograms providing a poor object model: figures (a) and (c) show two examples of images that are poorly approximated by marginal spatiograms. The lack of distinguishing spatial information makes the approximations generated by marginal spatiograms (third row) as bad as those produced by marginal histograms (second row).

$$
\begin{aligned}
&+[\mu^{(1)}(y) - \mu^{(1)}(y_0)]^T \tfrac{\partial \rho}{\partial \mu^{(1)}}(y_0) \\
&+[\mu^{(2)}(y) - \mu^{(2)}(y_0)]^T \tfrac{\partial \rho}{\partial \mu^{(2)}}(y_0)
\end{aligned}
$$

where the superscript notation refers to the tracker number (for example, $\mu^{(2)}$ refers to the bin spatial means of the features used by tracker 2). Using the fact that the tracker scores are independent with respect to the parameters of other trackers, we obtain

$$
\begin{array}{ccc}
\frac{\partial \rho}{\partial n^{(1)}} = \rho^{(2)}(y)\frac{\partial \rho^{(1)}}{\partial n^{(1)}} & , & \frac{\partial \rho}{\partial \mu^{(1)}} = \rho^{(2)}(y)\frac{\partial \rho^{(1)}}{\partial \mu^{(1)}} \\
\frac{\partial \rho}{\partial n^{(2)}} = \rho^{(1)}(y)\frac{\partial \rho^{(2)}}{\partial n^{(2)}} & , & \frac{\partial \rho}{\partial \mu^{(2)}} = \rho^{(1)}(y)\frac{\partial \rho^{(2)}}{\partial \mu^{(2)}}
\end{array}
$$

Inserting into the previous equation for $\rho(y)$

$$
\begin{aligned}
\rho(y) \approx \quad & \rho(y_0) + \\
& \rho^{(2)}(y_0)\{([n^{(1)}(y) - n^{(1)}(y_0)]^T \tfrac{\partial \rho^{(1)}}{\partial n^{(1)}}(y_0) + \\
& [\mu^{(1)}(y) - \mu^{(1)}(y_0)]^T \tfrac{\partial \rho^{(1)}}{\partial \mu^{(1)}}(y_0)\} + \\
& \rho^{(1)}(y_0)\{([n^{(2)}(y) - n^{(2)}(y_0)]^T \tfrac{\partial \rho^{(2)}}{\partial n^{(2)}}(y_0) + \\
& [\mu^{(2)}(y) - \mu^{(2)}(y_0)]^T \tfrac{\partial \rho^{(2)}}{\partial \mu^{(2)}}(y_0)\}
\end{aligned}
$$

Simplifying, and generalising to K trackers, we obtain

$$
\begin{aligned}
\rho(y) \approx \quad & \rho(y_0) + \\
& \textstyle\sum_{k=1}^{K} \tfrac{\rho(y_0)}{\rho^{(k)}(y_0)}\{([n^{(k)}(y) - n^{(k)}(y_0)]^T \tfrac{\partial \rho^{(k)}}{\partial n^{(k)}}(y_0) + \\
& [\mu^{(k)}(y) - \mu^{(k)}(y_0)]^T \tfrac{\partial \rho^{(k)}}{\partial \mu^{(k)}}(y_0)\}
\end{aligned}
$$

We can simplify this expression by defining two new variables:

$$
\begin{aligned}
\Gamma_n^{(k)} &= [n^{(k)}(y) - n^{(k)}(y_0)]^T \frac{\partial \rho^{(k)}}{\partial n^{(k)}}(y_0) \\
\Gamma_\mu^{(k)} &= [\mu^{(k)}(y) - \mu^{(k)}(y_0)]^T \frac{\partial \rho^{(k)}}{\partial \mu^{(k)}}(y_0)
\end{aligned}
$$

Inserting them into the previous expression:

$$\rho(y) \approx \rho(y_0) + \sum_{k=1}^{K} \frac{\rho(y_0)}{\rho^{(k)}(y_0)} \left\{ \Gamma_n^{(k)} + \Gamma_\mu^{(k)} \right\} \tag{5.20}$$

Taking the derivative of (5.20) with respect to y and setting this equal to zero, yields:

$$\sum_{k=1}^{K} \frac{\rho(y_0)}{\rho^{(k)}(y_0)} \frac{\partial \Gamma_n^{(k)}}{\partial y} = -\sum_{k=1}^{K} \frac{\rho(y_0)}{\rho^{(k)}(y_0)} \frac{\partial \Gamma_\mu^{(k)}}{\partial y}$$

where

$$\begin{aligned} \frac{\partial \Gamma_n^{(k)}}{\partial y} &= -\sum_{i=1}^{N} \alpha_i^{(k)} g\left(\| \frac{y_0 - x_i}{h} \|^2 \right) (y - x_i) \\ \frac{\partial \Gamma_\mu^{(k)}}{\partial y} &= -\sum_{b=1}^{B} v_b^{(k)} \end{aligned}$$

where $g(x) = -dk(x)/dx$ is the negative derivative of the kernel profile, which is constant if the Epanechnikov kernel (referred to in section 5.3.1) is used. $\alpha_i^{(k)}$ and $v_b^{(k)}$ are given by:

$$\alpha_i^{(k)} = \frac{C_h}{h^2} \sum_{b=1}^{B} \psi_b^{(k)}(y_0) \sqrt{\frac{n_b^{'(k)}}{n_b^{(k)}(y_0)}} \delta_{ib} \tag{5.21}$$

$$\nu_b^{(k)} = \psi_b^{(k)}(y_0) \sqrt{n_b^{'(k)} n_b^{(k)}(y_0)} (\hat{\Sigma}_b^{(k)}(y_0))^{-1} (\mu_b^{'(k)} - \mu_b^{(k)}(y_0)) \tag{5.22}$$

The values $\alpha_i^{(k)}$ can be interpreted as pixel weights that vote strongly when the bin-count of the bin they fall into is lower than the target bin-count, encouraging movement towards areas similar to the target histogram. The $\nu_b^{(k)}$ values are vectors that encourage the tracker to move so that bin spatial centres align with the target's spatial means. By moving all the terms that do not involve y to the right-hand side of the equation, the mean-shifted position, y, can be written as

$$y = \frac{\sum_{i=1}^{N} A_i g(\| \frac{y_0 - x_i}{h} \|^2) x_i - \sum_{b=1}^{B} V_b}{\sum_{i=1}^{N} A_i g(\| \frac{y_0 - x_i}{h} \|^2)} \tag{5.23}$$

where A_i and V_b are defined as:

$$A_i = \sum_{k=1}^{K} \alpha_i^{(k)} \rho(y_0)/\rho^{(k)}(y_0) \tag{5.24}$$

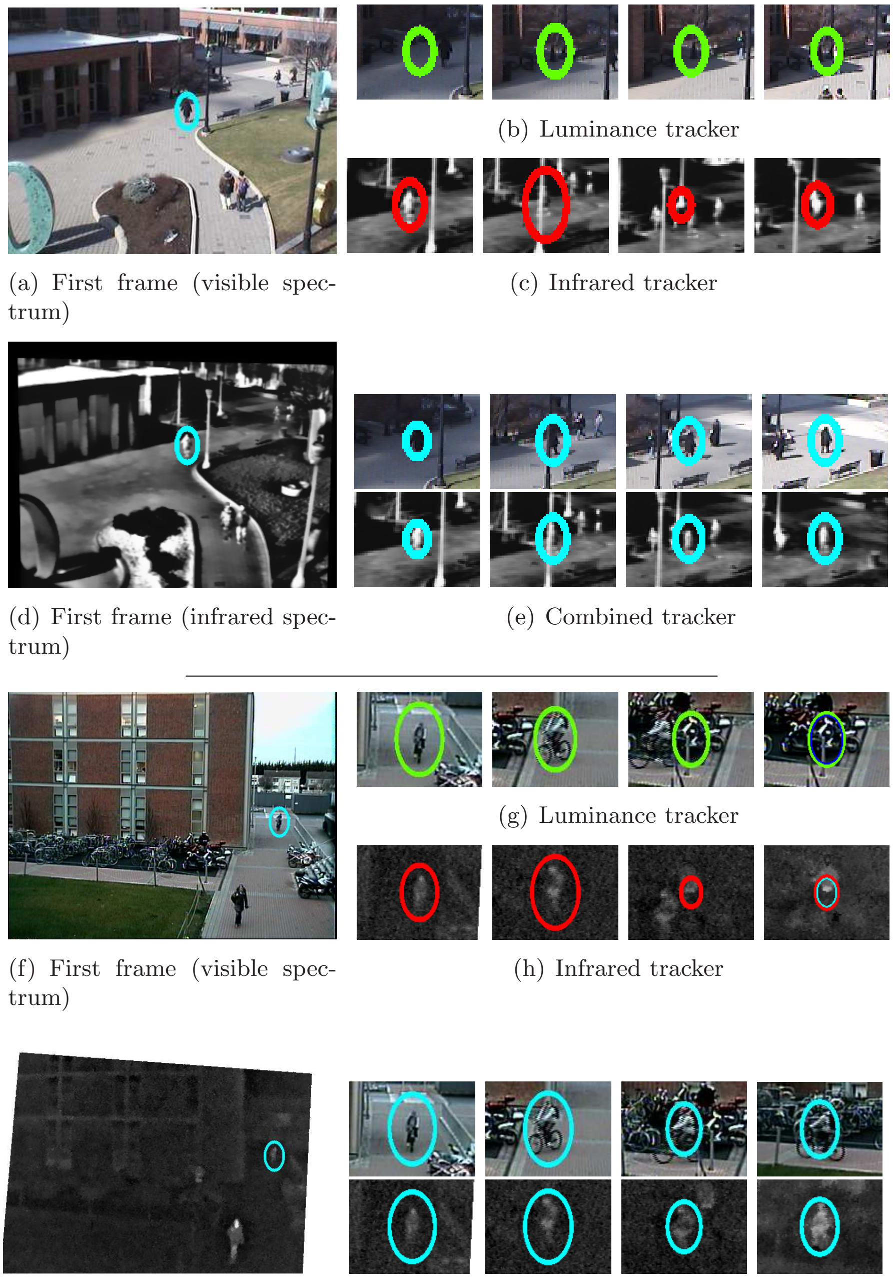
$$V_b = \sum_{k=1}^{K} \nu_b^{(k)} \rho(y_0)/\rho^{(k)}(y_0) \tag{5.25}$$

The combined mean-shift algorithm for multiple spatiogram trackers thus derived is used as follows: Given a starting image position y_0 near where the object is located, equation (5.23) is used to compute the next mean-shifted position, which should move towards the true object position. The new position y replaces y_0 in the equation and the procedure is iterated until convergence i.e. until y and y_0 are within the same pixel. To use equation (5.23), we first compute the similarity scores for each tracker using (5.7), then compute the combined score using (5.9). Using (5.21), the values of $\alpha_i^{(k)}$ are computed for each i^{th} pixel and k^{th} tracker. With (5.22), the 2-d vector values of $v_b^{(k)}$ are computed for each b^{th} bin and k^{th} tracker. Finally, A_i and V_b are computed and inserted into the mean-shift equation.

5.6 Tracking experiments

Three sets of experiments are shown in this section. The first experiment demonstrates how multimodal tracking significantly outperforms tracking using any one single feature. The next experiment illustrates the efficiency of the derived mean-shift procedure for tracking. Finally, quantitative tracking results are given, comparing the proposed tracking framework to standard histogram- and template-based tracking methods.

In the first experiment, the use of single features is compared to the combined-feature framework for tracking. Figure 5.6 shows the tracking results for two multi-modal video sequences. The data used is aligned visible spectrum and thermal infrared video, with each pixel represented by its colour components and infrared brightness. In the experiments, five features were used: Y, U, V, thermal brightness and edge orientation, with 8 bins per feature. An exhaustive search in an 11×11 window around the previous object location was used, and varied the scale by +/-10%, choosing the scale that returned the largest similarity score, as in [22] and [11]. The spatiogram models for the object are extracted in the first frame and remain fixed for the duration of the experiment. For the single-feature trackers, the object model used is a spatiogram. For the multi-feature tracker, a bank of spatiograms is used. Figures 5.6(b) and (g) show the luminance-based tracker, and (c) and (h) are the infrared-based tracker. Results



(a) First frame (visible spectrum)

(b) Luminance tracker

(c) Infrared tracker

(d) First frame (infrared spectrum)

(e) Combined tracker

(f) First frame (visible spectrum)

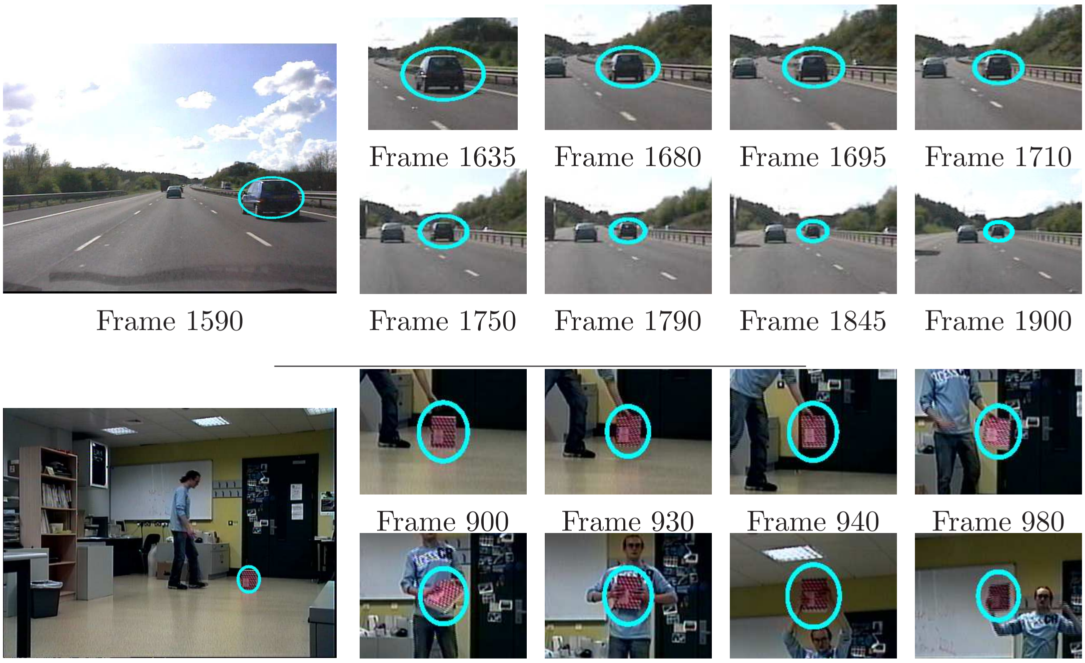
(g) Luminance tracker

(h) Infrared tracker

(i) First frame (infrared spectrum)

(j) Combined tracker

Figure 5.6: Tracking results using single features versus combined tracking: pedestrian and cyclist tracking. The left column shows the initial frame position of all trackers for two sequences (visible spectrum and thermal images shown in each case). The smaller images to the right show zoomed versions of the object tracked in subsequent frames of each sequence: (b),(g): Luminance-based spatiogram tracking. (c),(h): Infrared brightness spatiogram tracking. (e),(j): Combined tracking.



Frame 1635 Frame 1680 Frame 1695 Frame 1710

Frame 1590 Frame 1750 Frame 1790 Frame 1845 Frame 1900

Frame 900 Frame 930 Frame 940 Frame 980

Frame 880 Frame 990 Frame 1010 Frame 1080 Frame 1160

Figure 5.7: Illustrative tracking results using the mean-shift procedure for the combined tracker, using Y, U and V features (first sequence) and YUV and infrared features (second sequence).

for the other three features are omitted for clarity of presentation, but were always less effective than either luminance or infrared. The combined tracker, using all five features, is shown in (e) and (j). In the first difficult tracking sequence, taken from the OTCBVS benchmark dataset [27], the tracker attempts to follow a woman in dark clothing through occlusion and distraction by crowds. In frame 812, the luminance tracker fails as the woman walks into an area under shadow. In frame 1009, the infrared tracker fails and locks onto a passing person. There is little to distinguish people in infrared since, due to the camera pixel saturation, hot bodies appear bright white. The infrared tracker settles on a street-light until another person passes who it begins to track in frame 1230. The combined tracker tracks the person throughout the entire sequence, despite severe occlusion and background distraction. The second sequence in figure 5.6 was captured with the multimodal camera rig, described in chapter 2 and in [98], and shows similar results in tracking a cyclist. Both the luminance and infrared tracker fail when the cyclist turns the corner. The luminance tracker locks onto another bicycle in the bike-rack, while the infrared tracker locks onto another person who is standing in the bike-rack. The combined tracker, however, successfully tracks the cyclist for the entire duration of the sequence. Both sequences in figure 5.6 show that combining features outperforms any single feature in tracking.

Our second experiment, shown in figure 5.7 illustrates tracking results using the mean-shift procedure derived for the bank of spatiograms. In both sequences, 32 bins per feature were used and the mean-shift kernel was initialised at the location where the object was found in the previous frame. Mean-shift tracking was then performed using the derived procedure at three different scales (the current object size and +/- 10%). The scale that gave the largest similarity score was selected as the correct scale. YUV colour features were used in both experiments and infrared brightness was also used in the second sequence. In the first sequence, which is taken from the PETS2001 video dataset, the tracking of a blue car with a moving background is shown during a rapid change in scale. In the second sequence, which is a multimodal sequence (infrared band is not shown), the tracking of a book over a complex background is shown. As the book is at room temperature, the thermal features only add noise to the tracker but it still successfully keeps a lock on the target. The sequences required, on average, 7.68 and 10.95 iterations per frame, respectively, to converge. This is about 40 times faster than an exhaustive search in a $11 \times 11 \times 3$ local scale window. Using standard histogram or spatiogram mean-shift tracking would require over 340 times as many bins (32^3 instead of 3×32). In an interpreted MATLAB implementation, the mean tracking speed (over 36 different tracking tests) is just over 9 frames/second, which

Source	Type	Temp	Hist	Hist Bank	Spat 8 BPF	Spat 2 BPF	Spat 3 BPF	Spat Bank
DCU	person	100.0	29.1	29.0	29.1	34.4	100.0	**100.0**
PETS'01	vehicle	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
DCU	person	5.3	90.9	100.0	72.6	30.0	100.0	**100.0**
PETS'03	person	67.5	69.5	100.0	100.0	100.0	100.0	**100.0**
DCU	person	42.4	100.0	100.0	100.0	100.0	100.0	**100.0**
OTCBVS	person	100.0	97.5	100.0	51.2	100.0	87.1	**100.0**

Table 5.1: This table indicates the percentage of frames in which the object was successfully tracked by each tracker for 6 different sequences. The trackers are: a template tracker (Temp), a histogram tracker (Hist), a histogram-bank tracker (Hist Bank), three spatiogram trackers (Spat) using different numbers of bins-per-feature (BPF) and the proposed spatiogram-bank tracker.

includes reading bitmap images from hard-drive. It is envisaged that an optimised version would run comfortably in realtime.

In the third set of experiments, table 5.1 shows some quantitative tracking results comparing the proposed tracking framework to histogram- and template-based tracking methods. 6 sequences were used, taken from the public OTCBVS database, PETS'01, PETS'03 and a self-captured multimodal collection (marked DCU in the table). All sequences (except the PETS sequences) include an infrared channel, along with the RGB channels. Ground-truth was generated by manual annotation of bounding boxes on the objects to be tracked. Tracking was judged to have failed if the tracker's bounding box no longer included any part of the object. The figures indicated are the percentage of frames of successful tracking before failure. Histogram tracking ('Hist' column) was based on [22] using exhaustive search. The bank of histograms tracker ('Hist bank' column) used the same approach but multiplied the matching scores of each channel. For template tracking ('Temp' column) the standard sum-of-squared difference method was used [86]. The tracker in the 'Spat' column is a tracker using the full joint-spatiogram. All binned distribution models used 8-bins per feature. Object models were fixed at the start of the sequence and were not updated. Histogram and template tracking were chosen as they represent the two extremes of modelling feature spatial distribution. Histograms contain no spatial information and, at the other extreme, templates encode rigid spatial information. Spatiogram banks ('Spat bank' column), encoding a small amount of coarse spatial information, did best in the trials. The hist-bank tracker, although achieving a high success rate, often shrank in scale and tracked only part of the object. The full joint-spatiogram (‘Spat’ column), failed on a number of sequences and these failures can be attributed to the curse of dimensionality. The Spatiogram model contained 4096 bins ($8 \times 8 \times 8 \times 8$) and was therefore quite sparse, as the tracked objects generally contained less than 4096 pixels. Since, the spatiogram similarity measure performs a bin-wise comparison, it can return poor matches on sparse spatiograms. To counter this, the experiments were also performed with spatiograms with a smaller number of bins-per-feature (BPF). The columns marked *Spat 2 BPF* and *Spat 3 BPF* in table 5.1 show the results of spatiogram tracking with 2 and 3 bins per feature respectively. This gives a total of 16 (2^4) and 81 (3^4) bins in total, counteracting the sparse representation, but due to the coarseness of the quantisation, neither tracker performs as well as the spatiogram-bank.

5.7 Discussion and future work

In this chapter, the use of spatiograms for object tracking was reviewed and a spatiogram-based tracking framework was proposed to address the problems associated with multimodal data, specifically, the issue of exponential growth in memory and computation requirements as more features are added. The proposed framework splits the features over separate spatiograms, providing a compact and efficient object model. The validity of this framework was discussed and justified with illustrative examples and experimental results showing robust object tracking on a variety of sequences. A mean-shift procedure for efficient object localisation was derived for the proposed framework and mean-shift tracking was demonstrated on two sequences. In the experiments, the use of multi-feature tracking was shown to significantly out-perform single feature tracking and specifically, the proposed spatiogram-bank tracking framework was shown to perform more robust tracking than traditional methods, including those based on histograms and templates.

5.7.1 Spatiogram-bank feature splitting

The spatiogram banks used for the tracking in this thesis use one spatiogram per feature. The framework does however allow features to be split arbitrarily over any number of spatiograms. For example, thermal and HSV-colour features could be split over H, S, V and I spatiograms, or they could be split over HS and VI spatiograms, as well as other permutations. Future work may examine different methods of splitting features over spatiogram banks, other than using one spatiogram per feature, as used in this work.

5.7.2 Adaptive tracking

Despite displaying successful tracking in many sequences, it was noted that the mean-shift tracking would sometimes fail very quickly and lose track of the object. Additionally, both the exhaustive search and mean-shift procedures would often choose too large a scale for tracking the object when using either the spatiogram or spatiogram-bank object model. Examples of failed tracking are shown in 6.5. The failures were discovered to be due to the similarity measure used. In the next chapter, the drawbacks of the original spatiogram similarity measure are detailed, including why mean-shift tracking with this measure can suddenly fail and why it has difficulty with scale selection. A new similarity measure is proposed, based on deriving the Bhattacharyya coefficient for Spatiograms, and this measure is shown to overcome the original measure's faults.

In most sequences where failure of the proposed method was observed, there were three main reasons: (i) the object became occluded by the background or another object, (ii) a *distractor*, similar to the target object, confused the tracker and caused it to follow it instead, or (iii) the properties of the object changed so that the current object model was invalid, such as when a lighting change occurred. These issues correspond to the three open problems in tracking, detailed in chapter 2: *occlusion*, *feature failure* and *model failure*. In this thesis, the problem of object tracking during occlusion is not addressed, but the other two issues are tackled in the next chapter.

The most difficult tracking scenarios are those in which the environment or the object appearance changes. The next chapter will deal with how adaptation can be used within the spatiogram bank framework to robustify tracking in these difficult cases. The issue of *feature failure*, when the model used does not distinguish the object strongly from the background, can be tackled by adaptively weighting the features of the model to best discriminate object from background. A dynamic feature weighting architecture is investigated within the spatiogram-bank framework for this purpose. Also, the issue of *model failure* - where the object's appearance changes so that the current model is no longer a valid depiction of it - is addressed in the next chapter by adaptively updating the object model to cater for changes in pose or lighting.

Chapter 6

Adaptive Spatiogram Tracking

6.1 Introduction

The previous chapter introduced the concept of the spatiogram-bank tracker and showed that it provided both a compact object descriptor and outperformed standard methods of object tracking. In this chapter, the tracking framework is improved upon in three aspects. Firstly, an improved similarity measure for spatiograms is derived and evaluated. Secondly, strategies for updating the spatiogram-bank object model are tested. And thirdly, an adaptive weighting scheme for feature weighting is proposed for the spatiogram-bank tracking framework that caters for difficult tracking scenarios.

6.1.1 Chapter overview

Noting that spatiogram based tracking can sometimes select too large a scale for tracking, and that it can often quickly lose the object lock when using the mean-shift approach, the original measure for Spatiogram comparison is examined and shown to be deficient in a number of respects. A new similarity measure is derived from the Bhattacharyya coefficient and shown to be a more robust measure for object tracking.

Next in this chapter, two of the common causes of tracking failure are tackled, namely model failure and feature failure. Model failure, caused by changes in the object's appearance, is addressed by updating the object model to account for such changes. A number of updating strategies are evaluated in an object tracking context.

The second common cause of tracking failure, termed *feature failure*, is when the current model has difficulties separating the object and background. The spatiogram-bank tracking framework is shown to provide a flexible framework for dynamic feature weighting that allows the tracker to adaptively choose feature weights that best distinguish the object from the background clutter. This approach is shown to outperform the standard equally-weighted spatiogram-bank tracker and the state-of-the-art Collins tracker [20].

6.2 Improved similarity measure

In the previous chapter, the similarity measure used to compare Spatiograms, based on the average of the two Mahalanobis distance, was described. This measure, proposed in the original work, gives high similarity scores to spatiograms whose histogram bins counts are similar and whose spatial means are aligned. However, as now discussed, this similarity measure has significant shortcomings and these limitations are addressed in the next section by the derivation of an improved measure.

6.2.1 Disadvantages of the original measure

The originally proposed similarity measure, shown in equation (5.7), seemed to be chosen arbitrarily and has two main disadvantages. Firstly, it is not tolerant of small spatial changes in the feature bin centroids (i.e. changes in the μ values). This is clear from the $\hat{\Sigma}_b$ term, whose value is less than either of the individual variances. This means that small spatial changes in bin spatial means are heavily punished by the original measure. Secondly, a good similarity measure should have the property that if an image region spatiogram is compared to itself, the measure should return its maximal value. This is not true of the original measure. With $N(x; \mu, \Sigma)$ representing a normalised Gaussian evaluated at x, we can write the similarity between a spatiogram and itself as

$$\begin{aligned} \rho(S,S) &= \sum_{b=1}^{B} \sqrt{n_b^2}\,[N(\mu_b; \mu_b, (2\Sigma_b^{-1})^{-1})] \\ &= \sum_{b=1}^{B} \frac{n_b}{2\pi|(1/2)\Sigma_b|^{1/2}} = \sum_{b=1}^{B} \frac{n_b}{\pi|\Sigma_b|^{1/2}} \end{aligned}$$

This shows that comparing a spatiogram to itself does not equal a constant. Indeed, with the original similarity measure, it is possible for a patch which is **different** from the target patch to be a better match to the target, than the target itself! This is because the normalisation constant of the original similarity measure adds more weight to spatially-tighter feature clusters, and is the reason for the inaccurate scale selection that was observed in a number of sequences. On uniform backgrounds, the size of the object's bounding box will increase so that the object itself forms a spatially-tight cluster, giving a large score with this measure. For example, when tracking a black blob on a white background, the similarity will be increased if the tracker scale increases, as the blob will appear smaller, hence giving a smaller value of Σ_b for the black pixel bin. This weighting also leads to a non-smooth similarity surface, which often has many spiked peaks. This difficulty is overcome by deriving a new similarity measure in the next section.

6.2.2 Derivation of new measure

To derive the new similarity measure, the 2^{nd} order spatiogram is converted back to a histogram, by adding the extra dimension of space. Here, for simplicity, the derivation is done using one spatial dimension. However, the generalisation to two dimensions is straightforward. For bin b, its contents, n_b, is divided over an infinite number of spatial bins, $n_{b,k}$, where k is an integer ranging from $-\infty$ to $+\infty$. This is expressed as:

$$n_{b,k} = \frac{n_b \phi_b(k\Delta\mathbf{w})\Delta\mathbf{w}}{\sum_{i=-\infty}^{+\infty} \phi_b(i\Delta\mathbf{w})\Delta\mathbf{w}} \tag{6.1}$$

where $\Delta\mathbf{w}$ is the spatial size of each bin and ϕ_b is a normalised Gaussian with the mean and covariance of bin b. Since it is now possible to (theoretically) create a histogram from any spatiogram, spatiograms can be compared using the Bhattacharyya coefficient [22]. This has a relationship with the probability of Bayes error [146], and therefore is more similar to a probability than equation (5.7). Given two spatiograms (converted to histograms), $n_{b,k}$ and $n^{'}_{b,k}$, they are compared using the Bhattacharyya coefficient as follows:

$$\begin{aligned} \rho(n, n^{'}) &= \sum_{b=1}^{B} \sum_{k=-\infty}^{+\infty} \sqrt{n_{b,k} n^{'}_{b,k}} \\ &= \sum_{b=1}^{B} \sum_{k=-\infty}^{+\infty} \sqrt{\left(\frac{n_b \phi_b(k\Delta\mathbf{w})\Delta\mathbf{w}}{\sum_{i=-\infty}^{+\infty} \phi_b(i\Delta\mathbf{w})\Delta\mathbf{w}}\right)} \\ &\qquad \sqrt{\left(\frac{n^{'}_b \phi^{'}_b(k\Delta\mathbf{w})\Delta\mathbf{w}}{\sum_{i=-\infty}^{+\infty} \phi^{'}_b(i\Delta\mathbf{w})\Delta\mathbf{w}}\right)} \end{aligned}$$

As $\Delta\mathbf{w} \to 0$, the denominators of both fractions disappear, since ϕ_b and $\phi_b^{'}$ are both normalised Gaussians, therefore:

$$\sum_{i=-\infty}^{+\infty} \phi_b(i\Delta\mathbf{w})\Delta\mathbf{w} \approx \int_{-\infty}^{+\infty} \phi_b(x)\, dx = 1$$

This gives:

$$\rho(n, n^{'}) = \sum_{b=1}^{B} \sqrt{n_b n_b^{'}} \int_{-\infty}^{+\infty} \sqrt{\phi_b(x)\phi_b^{'}(x)}\, dx \tag{6.2}$$

This can be simplified further by noting that the product of two Gaussians is Gaussian, and also that the square-root of a Gaussian is Gaussian. The resulting Gaussians are not necessarily normalised, however, and therefore do not usually integrate to one. What remains are constant terms which can be thought of as weights for each bin comparison. Given that $\sqrt{N(x; a, A)} = qN(x; a, 2A)$, with $q = 2(2\pi)^{m/4}|A|^{1/4}$ for m dimensions, and $N(x; a, A)N(x; b, B) = zN(x; c, C)$ with $z = N(a; b, A + B)$, equation (6.2) can be simplified to produce the compact new measure:

$$\begin{aligned}
\rho(n, n^{'}) &= \sum_{b=1}^{B} \sqrt{n_b n_b^{'}} \int_{-\infty}^{+\infty} \sqrt{z_b \hat{\phi}_b(x)}\, dx \\
&= \sum_{b=1}^{B} \sqrt{n_b n_b^{'}} \sqrt{z_b} \int_{-\infty}^{+\infty} \sqrt{\hat{\phi}_b(x)}\, dx \\
&= \sum_{b=1}^{B} \sqrt{n_b n_b^{'}} \sqrt{z_b} \int_{-\infty}^{+\infty} q_b \hat{\hat{\phi}}_b(x)\, dx \\
&= \sum_{b=1}^{B} \sqrt{n_b n_b^{'}}\, [q_b \sqrt{z_b}] \\
&= \sum_{b=1}^{B} \sqrt{n_b n_b^{'}} \left[q_b \sqrt{N(\mu_b; \mu_b^{'}, \Sigma_b + \Sigma_b^{'})} \right] \\
&= \sum_{b=1}^{B} \sqrt{n_b n_b^{'}} \left[q_b Q_b N(\mu_b; \mu_b^{'}, 2(\Sigma_b + \Sigma_b^{'})) \right]
\end{aligned}$$

where $\hat{\phi}_b$ and $\hat{\hat{\phi}}_b$ are normalised Gaussians. Terms q_b and Q_b are given by

$$\begin{aligned}
q_b &= 2\sqrt{2\pi} \left| \Sigma_b + \Sigma_b^{'} \right|^{1/4} \\
Q_b &= 2\sqrt{2\pi} \left| (\Sigma_b^{-1} + (\Sigma_b^{'})^{-1})^{-1} \right|^{1/4}
\end{aligned}$$

Noting that $q_b Q_b = 8\pi|\Sigma_b\Sigma_b^{'}|^{1/4}$, the final similarity measure becomes:

$$\rho = \sum_{b=1}^{B} \sqrt{n_b n_b^{'}} \left[8\pi|\Sigma_b\Sigma_b^{'}|^{1/4} N(\mu_b; \mu_b^{'}, 2(\Sigma_b + \Sigma_b^{'})) \right] \tag{6.3}$$

6.2.3 Analysis of the new measure

Comparison to itself Firstly, comparing equations (5.7) and (6.3), of the old and new measures respectively, it is clear that the new measure is more tolerant of small spatial changes, as the covariance is equal to twice the sum of individual covariances. Secondly, if two identical spatiograms are compared using the new measure, we obtain:

$$\begin{aligned} \rho &= \sum_{b=1}^{B} \sqrt{n_b^2} 8\pi|\Sigma_b|^{1/2} N(\mu_b; \mu_b, 4\Sigma_b) \\ &= \sum_{b=1}^{B} 8\pi n_b \frac{|\Sigma_b|^{1/2}}{2\pi|4\Sigma_b|^{(1/2)}} = \sum_{b=1}^{B} n_b = 1 \end{aligned}$$

This shows that any spatiogram compared to itself will always receive a similarity score of 1 using the new measure, which is its maximal value.

Noise effects on similarity scores The plot in figure 6.1 shows the effect of noise on the proposed spatiogram similarity measure, compared to the original measure and to histogram similarity. The original similarity score is normalised by dividing by its maximum value so that it reaches a value of one at zero noise. The image used (top right corner of figure 6.1) was selected as it contained pixel clusters with different variances. Adding Gaussian noise causes a sharp decrease in the similarity score of the original measure, due to its intolerance of small spatial changes. Histogram based matching is quite insensitive to noise and even returns a relatively high matching score when the original image is lost in the noise signal. The proposed measure has a linear response within a large window of added noise.

Spatial effects on similarity scores Comparing equations 5.7 and 6.3, it is clear that the new similarity measure is more tolerant to spatial movement of colours, since it allows greater variance. This was also verified experimentally, as shown in figure 6.2. Using the RGB colour-space and quantising pixels into 8 bins per colour channel, a target image region was compared with other overlapping regions by shifting the region left and right by 20%. Note that the previously used similarity measure is normalised

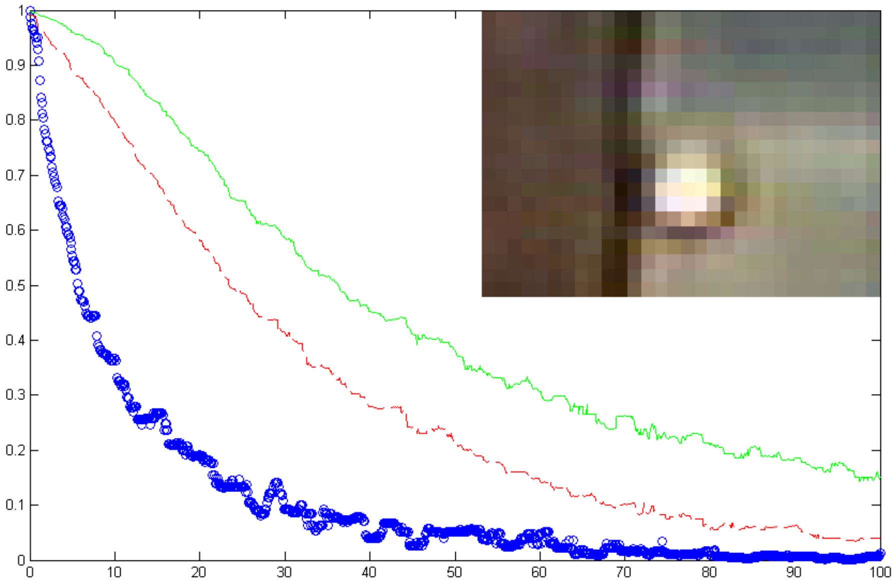


Figure 6.1: The effects on the similarity measures of adding Gaussian noise to a target image (top right): Histogram similarity (green, top curve), original spatiogram similarity measure (blue circles), proposed measure (red). Added noise RMS shown on x-axis.

so that its maximum value is one. As shown in the graph, the previously used similarity measure is intolerant of small spatial changes, where a shift of only a few pixels causes the measure to report a very significant difference between the target and the shifted region. A histogram-based comparison is completely tolerant of spatial changes, since it stores no spatial information. The new measure achieves a balance, as it is tolerant of spatial changes but not oblivious to them. Therefore it is a more discriminative model than a histogram.

Similarity surfaces Figure 6.3 shows the similarity surfaces generated by (c) the original measure, (d) histogram matching and (e) the proposed measure for two tracking examples, corresponding to localising a rigid highly-textured object (a book) and a non-rigid human object (a football player). In the top row of fig 6.3(c) the original measure produces a *spiked* similarity surface and finds the best match on the ball, instead of the player. This is because it weights bins with low variances higher, such as the tightly clustered white pixels of the ball. Its surface in fig 6.3(c), bottom row, is also not

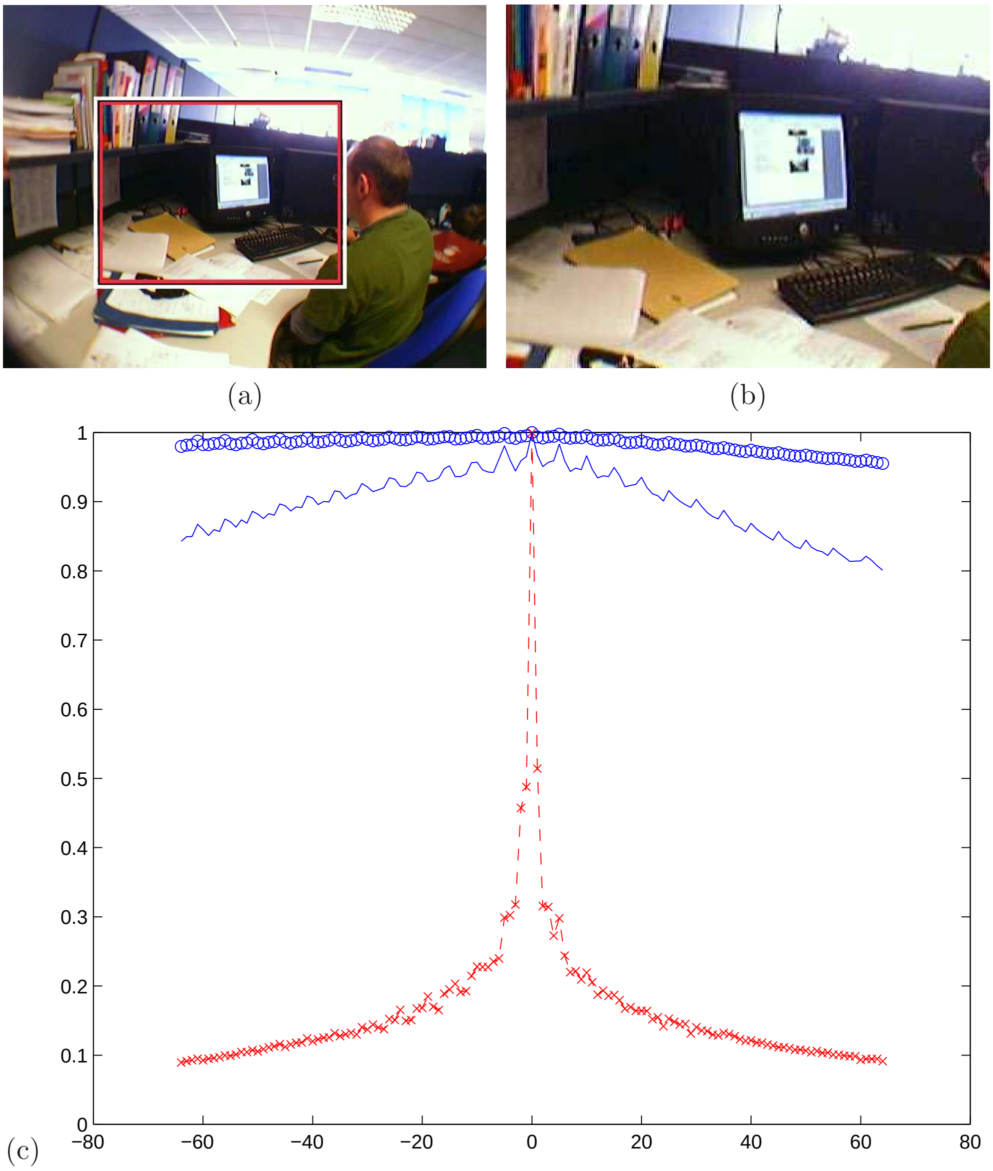


Figure 6.2: (a) Test image with target section highlighted, (b) Close-up of target section, (c) Similarity scores between target patch and similar patch moved horizontally from target position ±20%: Histogram similarity using Bhattacharya coefficient (blue circles), Previous spatiogram similarity measure (dashed-red crosses), the proposed measure (solid line).

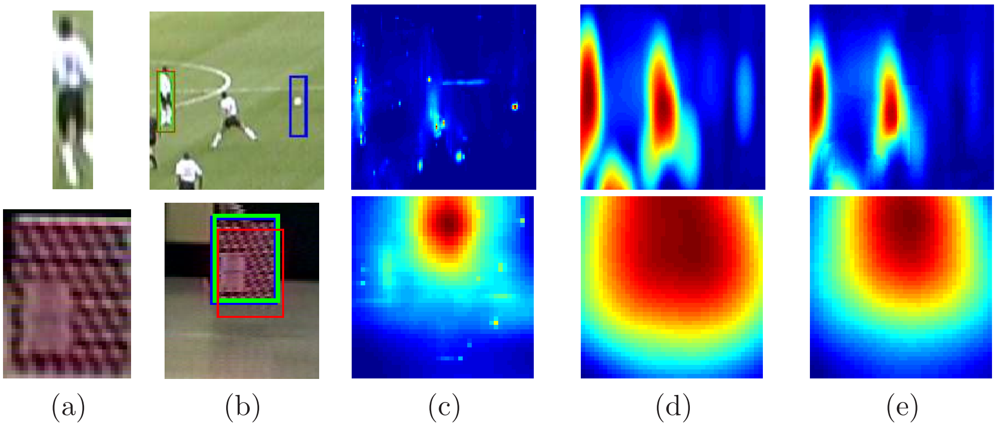


Figure 6.3: (a) Target objects, (b) Search areas with best matches shown: Original measure (blue), Histogram matching (red), Proposed measure (green). Similarity surfaces: (c) Original measure, (d) Histogram matching and (e) Proposed spatiogram measure

very smooth, but it correctly localises the book. Figure 6.3(d) shows the histogram similarity surfaces. Since it contains no spatial information, it provides poor object localisation, as can be seen by its poor lock on the book in fig 6.3(b). The proposed measure's surfaces, shown in fig 6.3(e), are smooth, as in histogram matching, but also have good object localisation, indicated by the tighter peaks.

Marginal spatiogram similarity surfaces The most important consideration in using marginal spatiograms for tracking is whether this approximation of the model allows robust discrimination of the tracked object from the background. In figure 6.4 some illustrative *similarity surfaces* are given, in order to discuss the differences between tracking using a full joint-spatiogram model and the bank of spatiograms model that uses marginal spatiograms. The newly derived measure is used for all surfaces.

Figure 6.4(a) and (b) show the initial position of the object to be tracked: a motorcyclist driving away from the camera. Object models are extracted from the rectangular region surrounding the object. Figure 6.4(c) and (d) show the same object, but 100 frames later. The similarity surfaces for the four individual features, {H,S,V,IR}, are shown in figures 6.4(e), (f), (g) and (h). These surfaces are generated using the four marginal spatiograms of the object model extracted from the first frame. The scale was adjusted to match the object size. Figure 6.4(i) shows the spatiogram bank similarity surface, obtained by computing the product of four individual feature surfaces. Figure 6.4(j) shows the surface obtained using the full spatiogram

In the figure, 8 bins were used per component, giving 4096 ($8 \times 8 \times 8 \times 8$) bins in the full spatiogram. Using more bins will exacerbate the curse of dimensionality, since in the original frame the object is 66 × 104 pixels in size, giving a total of 6864 pixels. For example, using 16 bins per component would mean that there are almost ten times as many bins as there are pixels. Using less bins per component would be expected to reduce tracking performance.

Comparing figures 6.4(i) and (j), the bank of spatiograms surface in (i) appears more peaked, with very well defined *distractors* in the form of the person in the top left and the motorbikes in the bottom right. The full spatiogram surface has broad areas of potential distraction, and these areas seem to be present in the individual surfaces of Hue (H) and Value (V). Distinct differences are certainly present, but it will be the tracking performance that determines whether the bank of spatiograms model provides a good tracking framework.

Object tracking using new measure The final experiment in figure 6.5 shows results from typical object-tracking scenarios. For this test, the multi-feature spatiogram-bank mean-shift tracking framework described in the previous chapter is used. The original measure is compared to the proposed measure for person tracking (top row) and head tracking (bottom row) using thermal infrared and colour pixel features. The mean-shift procedure for the new measure is very similar to the original procedure. Firstly, the spatial similarity part ψ_b is rewritten as:

$$\psi_b = 8\pi|\Sigma_b\Sigma_b^{'}|^{1/4}N(\mu_b;\mu_b^{'},2(\Sigma_b+\Sigma_b^{'})) \tag{6.4}$$

The (k) superscripts, indicating the feature number, have been omitted for clarity. Secondly, the combined variance $\hat{\Sigma}_b$ is rewritten as:

$$\hat{\Sigma}_b = 2(\Sigma_b+\Sigma_b^{'}) \tag{6.5}$$

These alternate values are used in equations (5.21) and (5.22), and then the procedure is exactly as before. The original measure mean-shift (shown in red) fails quickly when the tracked object moves, due to its narrow similarity surface peak. The mean-shift procedure generally fails when an object moves faster than the peak width. The proposed measure (shown in yellow) successfully tracks both objects.

Discussion The newly derived similarity measure for spatiograms has been shown to be superior to the original measure over a series of experiments. It has been shown to be more robust to noise and spatial changes in feature position that the original

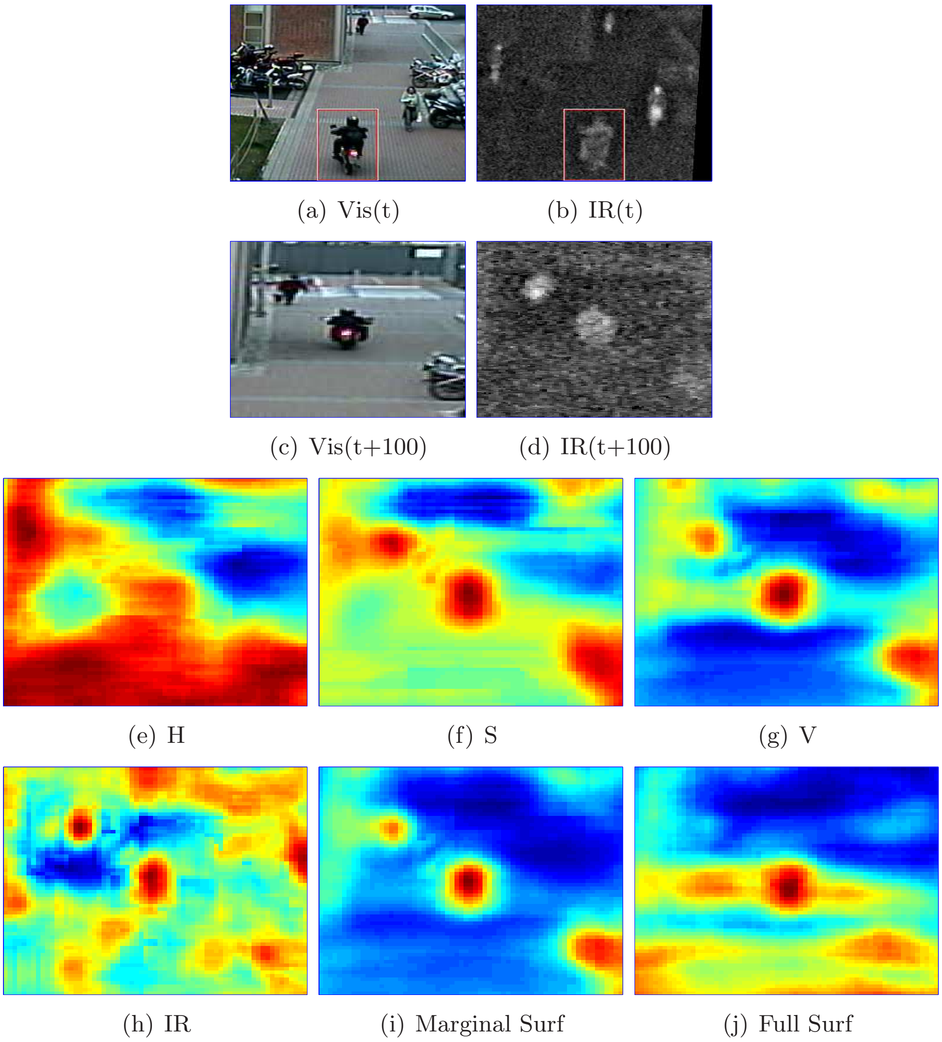

(a) Vis(t)
(b) IR(t)
(c) Vis(t+100)
(d) IR(t+100)
(e) H
(f) S
(g) V
(h) IR
(i) Marginal Surf
(j) Full Surf

Figure 6.4: Marginal spatiogram surfaces: (a) shows a target object at time t and (c) shows this object 100 frames later. The corresponding infrared frames are shown in (b) and (d). Spatiograms are extracted in frame (a) and compared to all parts of frame (c). Figures (e), (f), (g) and (h) show the similarity surfaces computed by comparing the extracted models in each feature-space. Shown in (i) is the product of all four similarity surfaces. Shown in (j) is the similarity surface using the joint-feature spatiogram. All spatiograms used 8 bins per component.

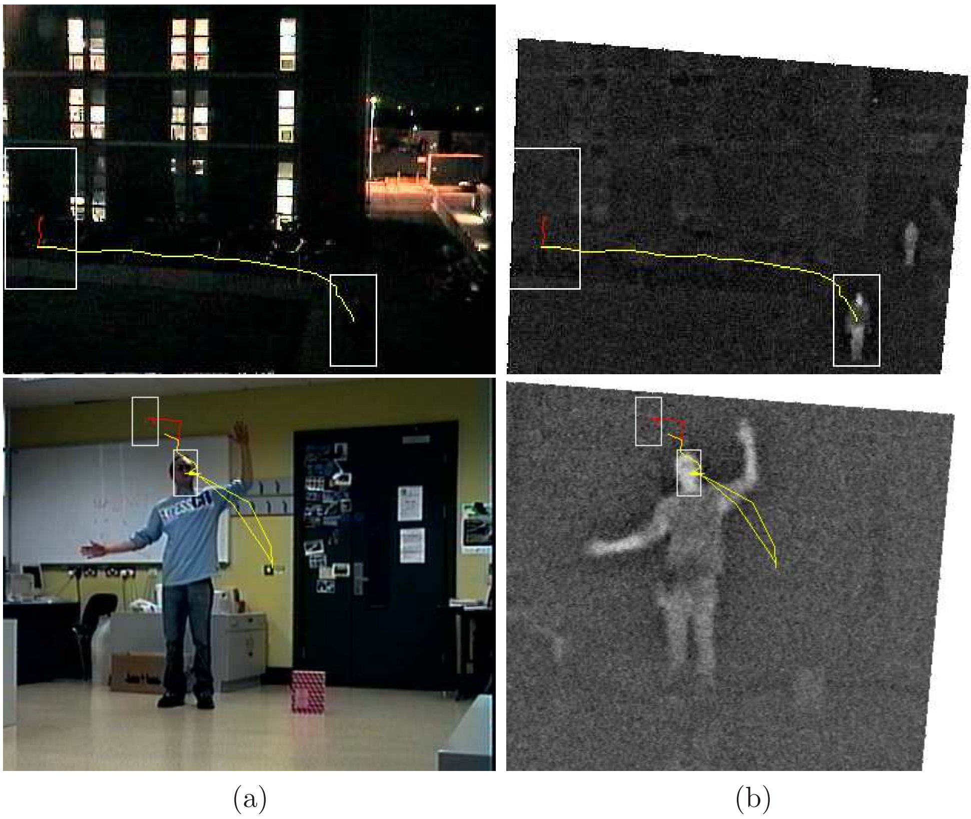


Figure 6.5: Multi-feature mean-shift tracking results using: (in red) Original measure, (in yellow) Proposed Measure. (a) Visible images and (b) Infrared images.

measure. Like the original measure, it is more discriminative than histograms, since it also considers spatial information as well as feature counts. In all further experiments in this thesis, the new similarity measure will be used.

6.3 Adapting to model failure

As mentioned previously, the problem of *model failure* describes scenarios where the object's appearance changes (a change in lighting or pose, for example) so that the current model is no longer an accurate depiction of the object. In this section an approach to tackle this problem in the context of spatiogram-bank tracking is proposed and evaluated.

6.3.1 Related work

In the experiments of the previous chapter, the object model extracted in the first frame was fixed as the object model for the entire tracking sequence. While this strategy was successful for the sequences used, it is likely that an object will change its appearance over longer sequences, due to gradual lighting changes or changes in object pose, causing the original model to be a poor representation of the object. Noting this potential problem, the object model can be updated continuously. For example, by using the best match in the current frame and mixing it with the object model, thereby gradually altering the model to cater for object changes.

However, by updating the model, there is problem that it may *drift* away from the correct solution. Each time the model is updated, background pixels may be included due to inaccurate object localisation and they will pollute the object representation, eventually leading to tracking failure. A remedy for this problem, proposed in both [86] and [20], is to assume the object model extracted in the first frame remains a reliable object representation, and to use it to anchor the updated model, thereby constraining potential drift from the original, good solution.

In [86], Matthews et al. use a template representation for the object. Two models are retained: the original object model and the best match in the last frame. In each frame, the object is tracked by performing a gradient ascent on the similarity surface, first using the original model to find a local peak and then initialising the gradient ascent procedure at this peak but using the other model to localise the object.

In [20], Collins et al. use a histogram representation for the object. The model used for tracking is a straightforward average of the histograms of the last-best-match and the initial object histogram.

In [88], objects are represented by Gaussian mixture models. These models are updated by adapting the model parameters online in each new frame, using a recursive estimate for the Gaussian model parameters obtained from the pixels sampled from the current object position. The adaptation is selective however, and is stopped when tracker failure is detected. This is determined to have happened if the data likelihood drops below an adaptive threshold.

In [162], the tracked object is represented by three appearance models: the original model, which remains fixed, the last best matching and a gradually updated template. The matching scores returned by each model are adaptively weighted using weights that depend on how well each model matches. The weights are slowly updated using an α forgetting-factor.

6.3.2 Spatiogram updating

In the context of using spatiograms for tracking, the strategies of Collins et al. [20] and Matthews et al. [86] can be followed by mixing the original spatiogram model with the best matching spatiogram in the current frame and using this model for object localisation. While mixing histograms is simply a weighted sum of individual histogram bins, mixing spatiograms requires the spatiograms to be converted back to their moment form first, before mixing. We will refer to the mixing of spatiograms as *Beta-blending*, in reference to the β parameter that controls the ratios of spatiograms that are mixed. The blending procedure is detailed in the next section.

Beta-blending spatiogram models Given the current spatiogram model and the best match in the current image, the update procedure is a blending of the two spatiograms and simply involves adding the bin-counts and moments of the two spatiograms. First we convert from spatial means and variances to moment sums:

$$m_{(x),b} = \mu_{(x),b} n_b \tag{6.6}$$

$$m_{(y),b} = \mu_{(y),b} n_b \tag{6.7}$$

$$s_{(x),b} = (\Sigma_{(x),b} + \mu^2_{(x),b}) n_b \tag{6.8}$$

$$s_{(y),b} = (\Sigma_{(y),b} + \mu^2_{(y),b}) n_b \tag{6.9}$$

where $m_{(x),b}$ and $m_{(x),b}$ are the first-order moment sums of bin b in the x and y directions, $s_{(x),b}$ and $s_{(x),b}$ are the second-order moment sums, n_b, μ_b and Σ_b are the spatiogram parameters. After both spatiograms have been converted to moments sums, the updated model is computed as the weighted sum of the moments, using a weighting parameter, β, to control the update rate $0 \leq \beta \leq 1$:

$$n_b^{'} = \beta n_b^{(1)} + (1-\beta) n_b^{(2)} \tag{6.10}$$

$$m_b^{'} = \beta m_b^{(1)} + (1-\beta) m_b^{(2)} \tag{6.11}$$

$$s_b^{'} = \beta s_b^{(1)} + (1-\beta) s_b^{(2)} \tag{6.12}$$

The moment sums are then converted back to spatiogram parameters to obtain the updated spatiogram:

$$\mu_{(x),b}^{'} = \frac{m_{(x),b}}{n_b^{'}} \tag{6.13}$$

$$\mu'_{(y),b} = \frac{m_{(y),b}}{n'_b} \tag{6.14}$$

$$\Sigma'_{(x),b} = \frac{s_{(x),b}}{n'_b} - \left(\frac{m_{(x),b}}{n'_b}\right)^2 \tag{6.15}$$

$$\Sigma'_{(y),b} = \frac{s_{(y),b}}{n'_b} - \left(\frac{m_{(y),b}}{n'_b}\right)^2 \tag{6.16}$$

6.3.3 Experimental setup

In the experiment in this section, a total of 15 objects from different sequences were used to investigate tracking performance of the bank-of-spatiograms model, comparing 5 update strategies. All sequences are multi-modal containing both visible spectrum and infrared imagery. Table 6.1 gives details of the objects used, their sizes and the lengths of the sequences. The 5 different object model updating strategies that were investigated are as follows:

No update The object model extracted in the first frame remains fixed for the entire duration of tracking.

Instant update The best match to the object model in the current frame is used as the object model for tracking in the next frame.

Gradual update The object model is gradually updated, using the best matching spatiogram in the current frame. The updating procedure is illustrated by:

$$M_{t+1} = \beta M_t + (1 - \beta)B_t \tag{6.17}$$

where M_t is the object model at time t and B_t is the best matching spatiogram in frame t. In the tests $\beta = 0.95$.

Mixture Two object models are retained: the original model and the best match in the current frame. Tracking in the next frame is done using a model that is an average of the two models, *i.e.* with $\beta = 0.5$.

Product Again in this strategy, two object models are retained: the original model and the best match in the current frame. This time however, the models are not blended, but kept separate. For each object position hypothesis, the scores from both models are computed and multiplied to give the final score for that position.

For each object to be tracked, the initial bounding box was supplied in the first frame and the trackers attempt to locate the object in all subsequent frames. The object model used is a bank of spatiograms, with one spatiograms for each of the 4 features: H, S, V and I (IR brightness). To find the object in the next frame, an exhaustive search method was adopted, using an 11×11 window and performing this search at three scales: 90%, 100% and 110% of the previous object radius. The bounding box of the area most similar to the current object model is selected as the object position in the frame. Manually annotated ground truth was used to assess tracker performance. Results of this experiment are given in the next section.

Object number	# frames in sequence	Size in Pixels	Object type
1	19	22 × 13	white vehicle
2	29	59 × 20	red car
3	92	15 × 10	black car
4	97	39 × 212	human torso (scale change)
5	101	38 × 46	human face
6	113	16 × 23	human face (scale change)
7	115	15 × 17	white car
8	129	20 × 26	human face (appearance changes)
9	141	14 × 31	human face
10	172	28 × 54	pedestrian
11	224	13 × 30	person (night-time)
12	234	27 × 51	human face
13	247	13 × 39	pedestrian (night-time)
14	201	33 × 37	human face (nearby distractor)
15	288	12 × 29	pedestrian (crowded scene)

Table 6.1: Objects in tracking database for model updating experiment.

6.3.4 Experimental results

While many measures of tracking performance have been used, such as the number of failed tracks and average centroid error, all results of this experiment are shown on a single plot, similar to a precision-recall curve. Figure 6.6 illustrates how tracking performance in a single frame is measured. For each sequence, and each frame, a tracker's precision and recall of the object pixels can be computed using hand-annotated ground-truth bounding boxes. This is done simply by dividing the overlap area of the bounding boxes by the ground-truth box area (for recall) or the tracker's bounding

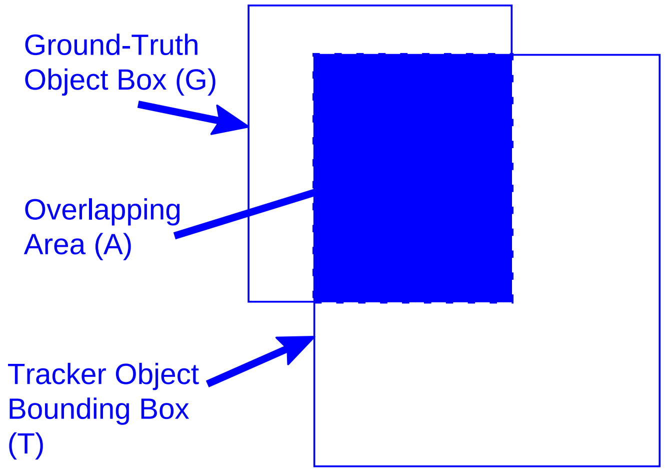


Figure 6.6: Computation of tracking performance using ground-truth bounding-box: The precision and recall of tracking can be written as $p = A/T$ and $r = A/G$. Combining them with the F_1 measure gives the tracking score, $F_1 = 2A/(G + T)$

box area (for precision). Precision and recall can be combined using the F_1 measure, giving $F_1 = 2pr/(p+r) = 2A/(G+T)$, as shown in figure 6.6. Using the computed F_1 measure, tracking failure can be judged to have occurred if this measure drops below a *failure threshold*. For different failure thresholds, the average percentage of correctly tracked object frames can be computed. Using this approach, figure 6.7 shows the results of the tests on each of the four model updating strategies. At very low failure thresholds, tracking is deemed to have failed only when the object is lost completely. At higher failure thresholds, the tracker must be more accurate in order to be judged successful and must not only remain on the object, but also be at the correct scale. In order not to bias the results for longer tests, since the sequences are of different lengths, the percentage of correctly tracked frames was computed for each sequence, then the average percentage was computed over all sequences.

From figure 6.7 it is clear that the instant update strategy performs poorly. This strategy can cause the tracker to drift quickly away from the object. The gradual update strategy performs better, but also drifts away from the object, albeit more slowly. Overall, the best updating strategy was the product update.

Due to the strong invariance of the infrared channel, the fixed object model (no update) performs well. However, it can be seen in figure 6.8, when compared to the product strategy, that the fixed model is too rigid to cater for changes in the pose of tracked objects. Here, tracking results are shown for object number 8 in table 6.1, a human face. The product strategy gives a better lock on the object since it adapts to

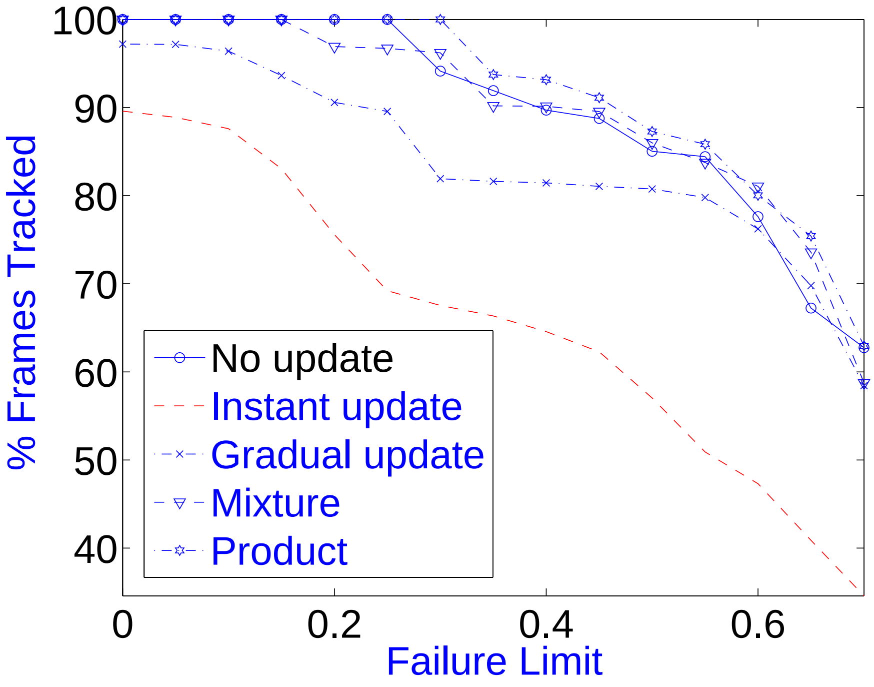
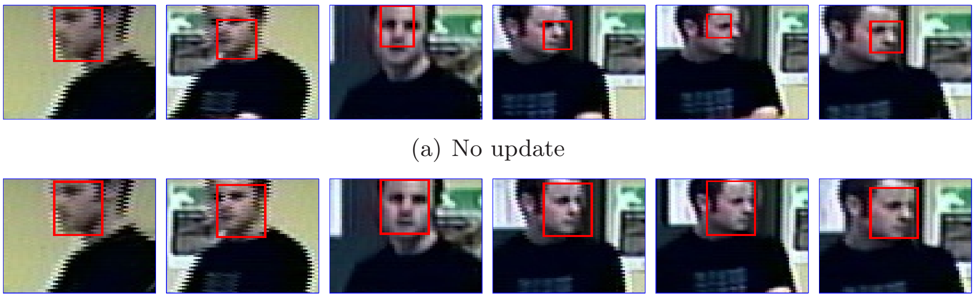


Figure 6.7: Model updating results: mean percentage of frames in sequence where successful tracking was achieved for a given failure tolerance

changing object pose, whilst the fixed model returns inaccurate tracking.

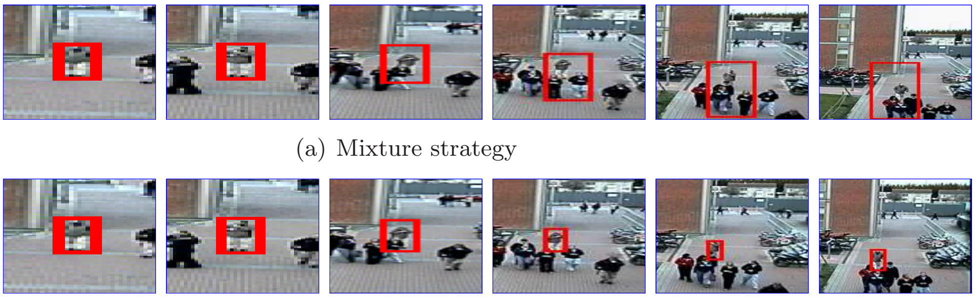
The product strategy appears marginally more accurate than either the fixed model or the mixture strategy, since it accounts for variations of the model by using both the last match and the original spatiogram. While the mixture strategy also attempts to do this, by blending the models some discriminating information may be lost. In figure 6.9, the mixture strategy and the product strategy are compared in tracking a pedestrian (object 15 in table 6.1). Due to the mixture's blending of the original and best-match models, it is severely affected by the partial occlusion and this leads to the tracker's incorrect scale lock. The product strategy, by retaining a separate copy of the original object model, keeps a lock on the object despite occlusion and similar nearby distractors.



(a) No update

(b) Product strategy

Figure 6.8: Example of comparing the *no update* and product strategies: because the product strategy uses information from the previous best match, it can adapt to the changing pose of the tracked object and provide a better lock on the object.



(a) Mixture strategy

(b) Product strategy

Figure 6.9: Example of comparing the product- and mixture- strategies: because the product strategy retains a separate copy of the original object model, it provides better tracking through partial occlusion. Using the mixture strategy, the model is polluted during occlusion.

6.3.5 Discussion

There are many other updating strategies that were not considered in this experiment. While the gradual model updating failed quickly, a selective update might be expected to fare better. For example, only updating the model when its matching score falls below a threshold. This strategy might perform well on some sequences, but selective updating is not a robust solution and will only delay the drift problem [86]. When an object's appearance changes so significantly that the original model is no longer a reasonable representation, then model updating is obviously necessary. However, it is unrealistic to expect a simple tracker to recognise the difference between an occlusion and a dramatic change in object appearance. Information from independent higher-level modules, are needed to robustly make such decisions. For example, a global lighting change might be detected by such a module and could inform trackers that brightness features are unreliable. Similarly, a multi-object tracker might be called upon to resolve possible occlusion scenarios. When drastic changes in object appearance happen suddenly (within a single frame), and affect all features, a tracker should not be expected to retain its lock. Instead, higher-level modules, such as an object-detector, are needed to reinitialise the model in these circumstances.

The use of the initial object model as a *reference distribution* allows accurate tracking even while the appearance of the tracked object varies, and also avoids the problem of *model drift.* By combining the original and best matching spatiogram, the object model will stay anchored to the original, but will also adapt quickly to changes in object appearance, leading to a tighter lock on the tracked object. However, the initial object model cannot always be expected to well represent the object in longer tracking sequences. As Collins et al. [20] conclude, *"Ultimately, the approach of maintaining a reference distribution needs to be discarded, as it limits the amount of variation that can be tolerated as the object appearance evolves."* Future works in this area should undoubtedly address this problem, perhaps requesting feedback from a detection module when the original model receives a poor matching score.

6.4 Adapting to feature failure

In tracking, the problem of *feature failure* occurs when the model used does not discriminate strongly between the object and the background. In this section a dynamic feature weighting approach is proposed to tackle this problem using the spatiogram-bank tracking framework. This proposed approach is compared to non-adaptive spatiogram-bank tracking and to Collins adaptive tracking, a current state of the art approach to adaptive tracking.

6.4.1 Related work

Two recent advances in adaptive tracking were proposed in [8] and [20]. The similarities and differences between Avidan's adaptive tracking [8] and Collins' adaptive tracking framework are discussed here.

In both [8] and [20], spatial information relating to the tracked object is discarded and pixels are treated independently as samples coming from the object or the background distribution. Both methods create weight images by determining the best feature spaces for tracking, assigning higher weights to pixels that are more likely to belong to the object distribution and then using the mean-shift algorithm on this weight image for tracking in the next frame.

Collins' method evaluates a series of pre-defined features to see which of them best allows the object to be tracked. Avidan's method avoids having to evaluate many features and instead trains classifiers to separate the object and background pixels using a least-squares algorithm. Avidan's method is therefore a more efficient framework if a large number of features were to be used. Collins' method, on the other hand, provides a more flexible non-linear mapping of pixel features to pixel weights, with better handling of multimodal distributions in both the object and background features.

The Collins' adaptive tracking method is now described in order to provide context for experiments later in this chapter where its tracking performance is compared to the proposed adaptive spatiogram-bank tracker.

Collins' adaptive tracking The Collins' adaptive tracking method [20] is an algorithm for feature selection that aims to select the best features for object tracking. Any set of features can be used as a candidate set, but the original work targetted fast online selection of features and therefore chose features that were easy to compute. The seed features used were given by:

$$F = \{w_1 R + w_2 G + w_3 B | w_* \in [-2, -1, 0, 1, 2]\} \tag{6.18}$$

which, after removing redundant features, leaves 49 candidate features. These include raw R, G, and B values, as well as brightness ($R+G+B$) and approximate chrominance features such as $R - B$. Each feature is then evaluated as follows. First, histograms of the object and the background are created. The *background* is a predefined rectangular area surrounding the object, but excluding the object's pixels. For each feature-bin, the log likelihood ratio of object to background probability is computed using:

$$L(i) = \log \frac{\max\{p(i), \delta\}}{\max\{q(i), \delta\}} \tag{6.19}$$

where p and q represent the normalised histograms of the object and background respectively, and δ is a small value (set to 0.001 in the original work) that prevents dividing by zero or taking the log of zero. This function is then used to map the pixels non-linearly to a new *tuned* feature space. The variance ratio is then used to evaluate how well the new tuned feature separates the object and background classes. This ratio is given by:

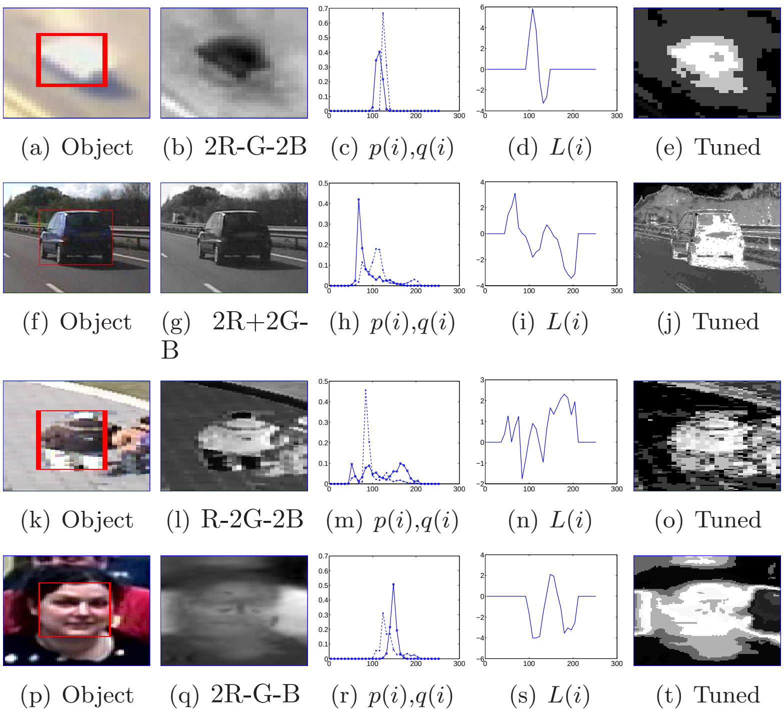
$$VR(L; p, q) = \frac{var(L; (p+q)/2)}{var(L; p) + var(L; q)} \tag{6.20}$$

where the *var* function is computed as by:

$$var(L; p) = \sum_{i=1} p(i)L^2(i) - [\sum_{i=1} p(i)L(i)]^2 \tag{6.21}$$

After ranking all evaluated features by their variance ratio, the top K tuned features are then used for tracking in the next frame. Tracking is performed separately with each of the K tuned features by using the mean-shift algorithm to find the nearest local mode in their respective weight images. The K estimates of the object position are combined using a naive median estimator, where the x position is given by the median of the x positions of all estimates and similarly for the y position.

Figures 6.10 and 6.11 show examples of tuned features for six objects. For each object, the feature used is the top ranking feature using the variance ratio. For the face images, $2R - G - B$ is the best feature, as it emphasises the skin hue. Although there is a strong red object near the face in figure 6.10(p), the tuned feature can handle multimodal distributions in the background using the log-likelihood ratio and the red t-shirt does not appear strongly in figure 6.10(t). On the other hand, while the tuned feature of figure 6.10(k), in 6.10(o), strongly separates it from the average background, the person on the right acts as a significant distractor. The tracker could easily jump to the other person, since they are not strongly distinguished. Scenarios such as this, where there is a spatial clustering of high likelihood pixels (a distractor) in the background, led Collins et al. to propose an alternative to the variance ratio: the *peak difference*. The peak difference is a feature quality measure, like the variance ratio, that attempts to determine how good a feature is for tracking. Unlike the variance ratio, it does not


(a) Object (b) 2R-G-2B (c) $p(i)$,$q(i)$ (d) $L(i)$ (e) Tuned

(f) Object (g) 2R+2G-B (h) $p(i)$,$q(i)$ (i) $L(i)$ (j) Tuned

(k) Object (l) R-2G-2B (m) $p(i)$,$q(i)$ (n) $L(i)$ (o) Tuned

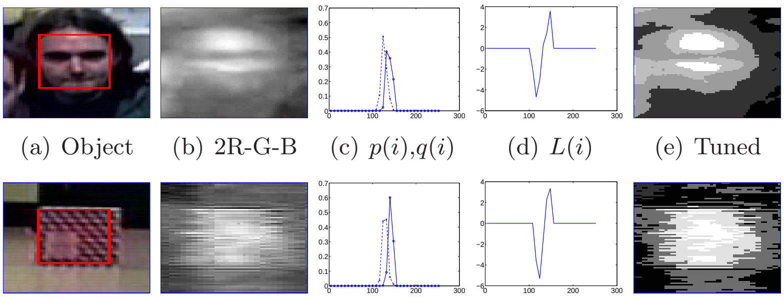
(p) Object (q) 2R-G-B (r) $p(i)$,$q(i)$ (s) $L(i)$ (t) Tuned

Figure 6.10: Collins adaptive tracking examples (1)

measure the difference between the object and the average background, but targets the main distractor in the surrounding background. Specifically, feature performance is measured as follows with the peak difference method: firstly, the weight image is obtained from the tuned feature, as before with the variance ratio. Next, the weight image is smoothed with a Gaussian kernel with its size comparable to the object size. The peak difference is then computed by measuring the difference between the largest peak inside the object bounding box and the largest peak outside it (the maximum distractor). Choosing features in this way provides better tracking features in the presence of non-uniform background and distractors, as shown in figure 6.12. Later in this chapter, the Collins peak difference tracker is compared to the proposed adaptive spatiogram-bank tracker.

6.4.2 Proposed approach

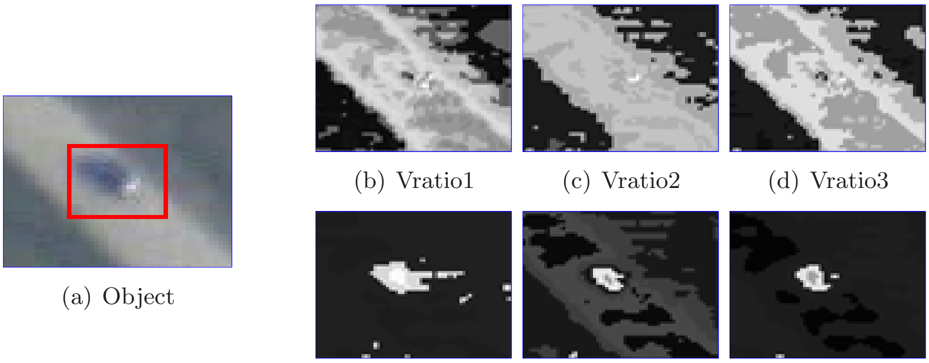
Weighting architecture While the Collins method provides a good architecture for feature selection, it uses the chosen features separately for tracking, only combining their results afterwards. In the proposed approach, the tracking procedure uses the fea-



(a) Object (b) 2R-G-B (c) $p(i)$,$q(i)$ (d) $L(i)$ (e) Tuned

(f) Object (g) R-2G+B (h) $p(i)$,$q(i)$ (i) $L(i)$ (j) Tuned

Figure 6.11: Collins adaptive tracking examples (2)



(a) Object

(b) Vratio1 (c) Vratio2 (d) Vratio3

(e) Pdiff1 (f) Pdiff2 (g) Pdiff3

Figure 6.12: Collins peak difference vs. variance ratio: choosing features with the best average separation between the inside and outside of the object bounding box, as the variance ratio does, can result in poor tracking features if the background if not uniform. The features chosen using the peak difference method, shown in (e), (f) and (g), are superior to those chosen by the variance ratio, shown in (b), (c) and (d).

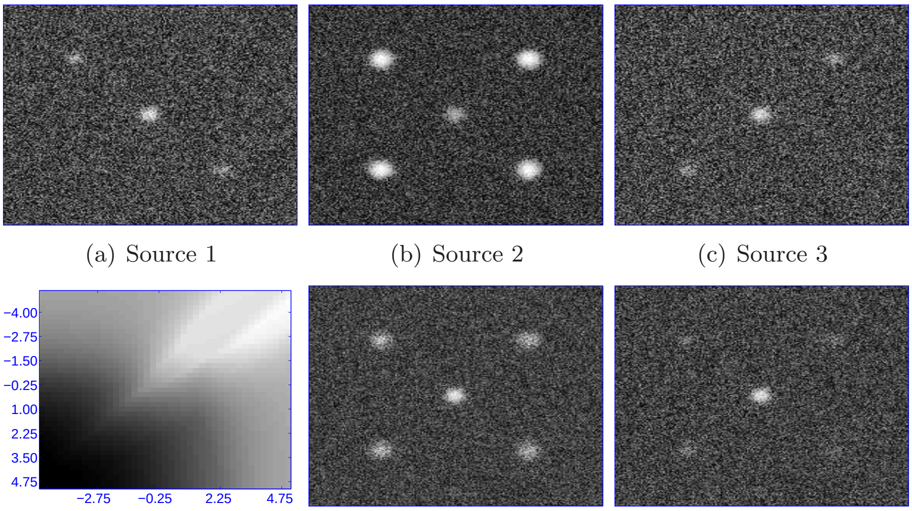
tures simultaneously, within the spatiogram-bank framework, but individually weights each feature's contribution to the process. When multiple features are used for tracking, prior work has shown that assigning different weights to features can benefit tracking [125, 135]. In the spatiogram-bank framework, once the weights for each feature have been determined, these weights must then be used to combine the scores returned by each feature for a given object position hypothesis. In this work, a weighting scheme suggested by Solberg et al. in [4] is used. Given a set of weights, w_i, for each tracker, the following weighting formula computes the combined score:

$$\rho(y) = \Pi_{i=1}^{K} \rho^{(i)}(y)^{w_i} \tag{6.22}$$

where, as before, K is the number of spatiogram-banks (considered equal to the number of features in this thesis) and y is the location of interest. Solberg et al. recommend using this formulation, where each source is assigned a weight according to the reliability of the information it provides. This formulation for combining features was also used by Jones et al. [62] in their work on fusing data from disparate imaging systems for surveillance for detecting people in a surveillance context. It may also be interpreted in another way. If the *log* of this equation is computed, it is similar to the weighted sum used in *democratic integration* [135] to fuse multiple cues. The next section describes how these weights are selected adaptively.

Weight selection criteria As the aim of the tracker is to accurately locate the object amongst the background clutter, the *reliability the information* from a source can be gauged by how well it distinguishes the object from the background. Inspired by the the work of Collins et al. [20], it is proposed to select the feature weights to maximise the ratio of the object score to the most prominent background distractor. The object score is defined as the score at the current object position. The distractor score is defined as the maximum score of all locations that are outside the object bounds. The object radius, which defines the object bounds, was set as half the object size.

In figure 6.13 a synthetic example of selecting weights for tracking is shown. Figure 6.13(a), (b) and (c) show the similarity surfaces for three features. The centre of each surface is the true object position. When these 3 surfaces are fused using equal weights, the resulting surface is shown in figure 6.13(e). By selecting the weights to optimise the object-to-distractor ratio (OD ratio), figure 6.13(f) is obtained. This optimal surface clearly allows the object to be more easily distinguished from the background distractors. Figure 6.13(d) shows the OD ratio surface. The x-axis is $log(w_3/w_2)$ and the



(a) Source 1 (b) Source 2 (c) Source 3

(d) Weighting score surface (e) Equal weights (f) Optimal weights

Figure 6.13: Synthetic data example of optimal weighting. The sources in (a), (b) and (c) are to be combined. Each source represents a similarity surface for one particular feature, with the central peak representing the tracked object and the others are distractors. Optimum weights chosen were: $w_1 = 0.43$, $w_2 = 0$ and $w_3 = 0.57$. All images shown in log scale.

y-axis is $log(w_2/w_1)$. The peak is located near the top right corner, with $w_1 = 0.43$, $w_2 = 0$ and $w_3 = 0.57$

Determining optimal weights To find the weights that maximise the object-to-distractor ratio (OD ratio), a search must be conducted in a K-dimensional weight-space, where K is the number of features used for tracking. In this work, the search was conducted as follows.

The input required is (i) an initial set of weights, (ii) a set of candidate positions and their corresponding scores from each feature and (iii) knowledge of which candidate corresponds to the true object position. Firstly, the current distractor is located. This is simply the background candidate that gives the highest score using the current set of weights. The gradient vector, V, of the OD ratio in weight-space is then computed (derivation given in Appendix A.2). The search continues by rescaling the gradient vector, V, to length $L = L_0$ and moving along this vector in the weight space. If the OD ratio increases, the procedure is repeated, recomputing the gradient, etc. If the OD ratio decreases, we return to the previous position in weight-space and the length L is halfed. This continues until either the rescaled V leads to a higher OD ratio value, or L becomes smaller than L_{min}. In the experiments, the values used were $L_0 = 0.1$ and $L_{min} = 0.001$. When L becomes smaller than L_{min}, the procedure is determined to have converged.

Since there is usually only a small set of candidate positions and scores, this procedure is quite fast. This allows the method to be run with multiple initialisations whilst not adversely affecting the computational time for tracking. The initialisations that were used were: (i) equal weights for all features (1 run), (ii) zero weights for all features except one (K runs), (iii) equal weights for all features except one, which was given weight zero (K runs). In addition, the weights used in the last frame are also used to initialise a search. In total, this gives $2K + 2$ different initialisations for the optimal weight search procedure.

To find a set of background candidates, the area around the current object position is sampled at 16 pre-defined location. These positions are shown in figure 6.14. These position were selected so as to give a good overall coverage of potential areas of distraction, without expending too much computation. All points are at a distance of $0.5w$ or $0.75w$ from the object centre in the horizontal direction (with w as the object width) and $0.5h$ or $0.75h$ from the object centre in the vertical direction (with h as the object height). Mean-shift trackers can also be initialised in similar areas to find local distractors, but this simple method was found to give a good enough sampling of background distractors to provide accurate weightings, as well as being faster.

Since the object must be well localised in scale, as well as in space, the current object position is evaluated at various scales, obtaining K similarity scores per scale. The scales used are the current object scale multiplied by $S \in \{0.9^4, 0.9^3, 0.9^2, 0.9, 1.1, 1.1^2, 1.1^3, 1.1^4\}$. These scales correspond to repeated decreasing or increasing the original scale by 10%. These scores are not added to the candidate list, since they are very close to the object score and would always be selected as distractors, dominating the weight selection process. Instead these scores are used to screen the weights selected by each of the $2K + 2$ initialisations. For each set of weights returned, a *scale change value* (SCV) is computed. If the maximum score, using the weights, is the object score at the original scale then the SCV is zero. If the maximum score, using the weights, occurs at scale 0.9 or 1.1 then the SCV is one. If the maximum score, using the weights, occurs at scale 0.9^2 or 1.1^2 then the SCV is two, etc. For all the $2K+2$ weight sets, the minimum scale change is computed. Of all weight sets that comply to this minimum scale change, the one with the highest OD-ratio is chosen as the weights for this frame. This heuristic ensures that the selected weights not only localise the object at the correct spatial position, but also at the correct scale.

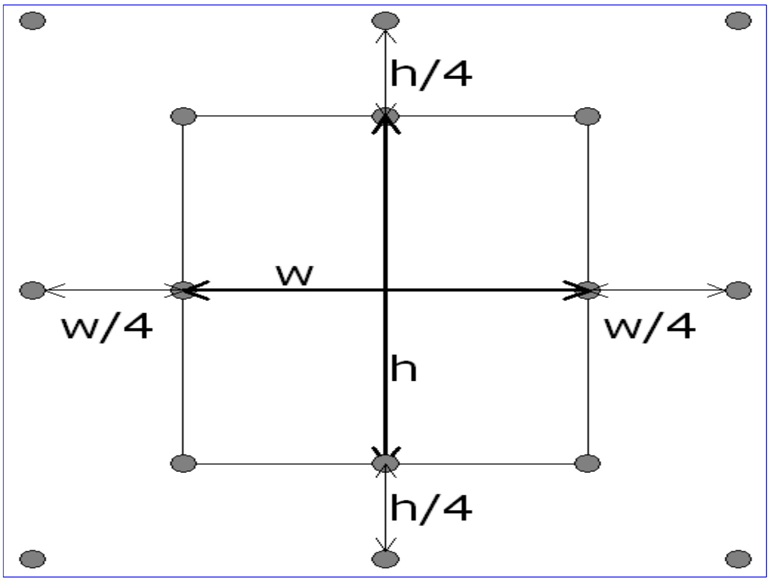


Figure 6.14: Background sampling positions around the tracked object: the 16 circles represent the nearby points that are evaluated in order to find feature weights that best discriminate the object from these background samples.

The adaptive spatiogram tracker used in these experiments used the product update strategy, as that was the best performer in the updating strategy tests in the section 6.3. The object was therefore represented by two models: the original model and the last-best-match model. At every evaluated position, $2K$ matching scores are returned, one for each of the K features of the 2 models. To reduce this to K features for the weight selection process, the corresponding scores for each feature are multiplied. Therefore, as before, each feature is assigned a single weight. It is also possible to assign different weights to the features of the two models. For example, instead of having a weight for the infrared brightness feature, to assign a different weight to the infrared brightness of the original model and the infrared brightness of the previous-best-match model. This approach was found to favour the previous-best-match model, since it was more likely to resemble the currently tracked object, and drift could quickly occur.

6.4.3 Adaptive versus non-adaptive tracking

In many instances, tracking tests using thermo-visual sequences are trivial for simple trackers, as the infrared causes people to appear as bright (hot) objects against a dark (cold) background. In order to investigate the benefits of adaptive feature weighting and to challenge the trackers, colour sequences alone are used in this batch of tests, without an infrared component. The standard spatiogram-bank tracker is compared to the adaptive model, each using a set of seven features. The first three features are standard H, S and V pixel colour values. The next 4 features are local edge orientation gradient histogram counts, similar to those used in [8] for multi-feature tracking. Specifically, in a 5×5 window around each pixel a histogram of gradient orientation is computed using 4 bins (0^o, 45^o, 90^o, 135^o). Each pixel casts a weighted vote for its orientation using its gradient magnitude as a weighting. Each pixel therefore has a 4 bin histogram of the gradient orientation in its local area. These histograms are normalised to sum to 1 and then multiplied by 255 to scale them to standard pixel values. These 4 new *bands* make up the 4 features in addition to HSV. Both trackers use a coarse-to-fine search in a 13×13 area in each frame, then select the best match from 3 scales (±10% of the object radius).

The results of four tracking tests are shown in figures 6.15, 6.16, 6.17 and 6.18 and illustrate the tracking performance of the adaptive method compared to the standard spatiogram-bank. These figures are now described in detail.

Background distraction Figure 6.15 illustrates a tracking scenario where the background is complex and causes significant distraction to the non-adaptive tracker. A person in a white jacket is to be tracked, walking in front of a highly textured bike rack. The non-adaptive tracker has a poor scale lock due to the movement of the person and the complexity of the background. By frame 430, the tracker has locked onto the background and tracking is lost. The adaptive tracker chooses to use only the V (brightness) feature and achieves perfect tracking. Since the person and the background are made of black and white pixels, neither hue nor saturation can assist in tracking. Similarly, the edge information is distracting due to the background.

Occlusion In figure 6.16, a person in black, walking towards the camera, is occluded by other similarly-dressed people. Both trackers succeed to track the person until the occlusion. The adaptive tracker increases the weights of the edge features during the occlusion to help discriminate the person from the others, indicated by the spike in the blue line with crosses in the graph around frame 2790. The non-adaptive tracker loses track and instead begins to track the other person.

Noise A noisy night-time scene is shown in figure 6.17, where the trackers attempt to track a darkly coloured car. The adaptive tracker quickly down-weights unhelpful features such as hue and saturation, using primarily the brightness feature, occasionally with some edge information. The saturation of surrounding pixels are similar to the object, so using this feature, as the non-adaptive tracker does, might not reduce the OD-ratio. However, the hue and saturation features do cause significant *scale distraction* causing the non-adaptive tracker to increase in size. The weight selection method of the adaptive tracker avoids the scale distraction by removing these features from use.

Lighting changes Figure 6.18 shows the tracking of a man during a strong lighting change. This sequence is from the publicly available OTCBVS dataset. During the most severe cloud cover, the object darkens causing the non-adaptive tracker size to increase, starting in frame 181. The adaptive tracker manages to keep a good lock on the object by increasing the contribution of the edge features.

These sequences strongly support the use of adaptive weighting in selecting the best features for object tracking. Additionally, the choice of selected features seems to be in accordance with the intuitive expectation of which features would be helpful in the scenarios shown.

Despite the results of figure 6.16, the adaptive method should not be particularly robust to large occlusions, since it will try to choose features that emphasise what is currently contained within the tracking window, whether that is the object or the occluder. However, the fact that the original model is retained will cause features that also help discriminate the original model to be selected, and could explain its success.

6.4.4 Comparison to the Collins adaptive tracker

In this section, the adaptive spatiogram-bank tracker is compared to the Collins adaptive tracker. The Collins tracker worked as follows: To select background samples, a bounding box of 2.5 times the object width/height was centred on the object, and pixels which did not correspond to the object were considered background pixels. A total of 52 features were evaluated in each frame; the original 49 linear combinations of R, G and B, as well as 3 infrared based features: I, $3I - R - G - B$ and $I + R + G + B$. The maximum distractor method (peak-difference) was used, as described in section 6.4.1, and the number of features for tracking was set at $K = 3$. For each feature, the meanshift procedure was run at three scales (±10% of object radius) and the scale returning the highest sum of weights from the weight-image was selected as the correct scale.

The adaptive spatiogram-bank tracker uses H, S and V features in the second test, shown in figure 6.21, for colour face tracking. In all other tests, it used 4 features: R,

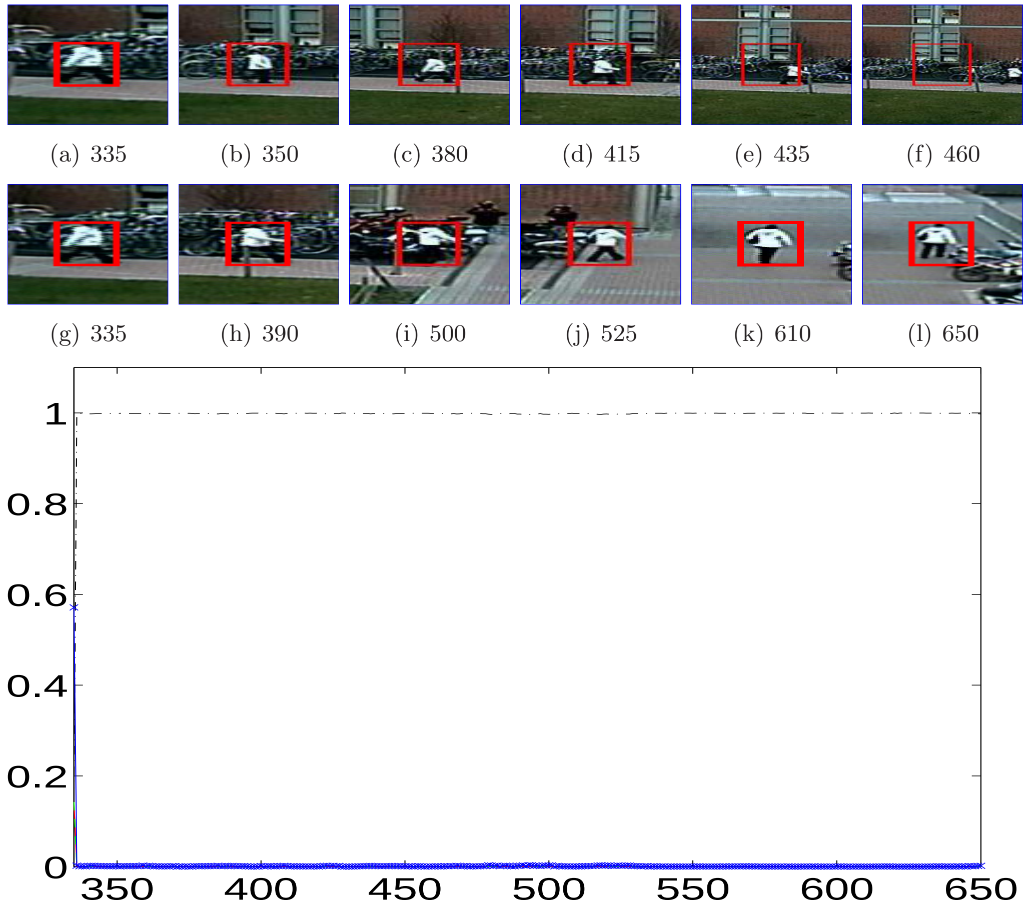
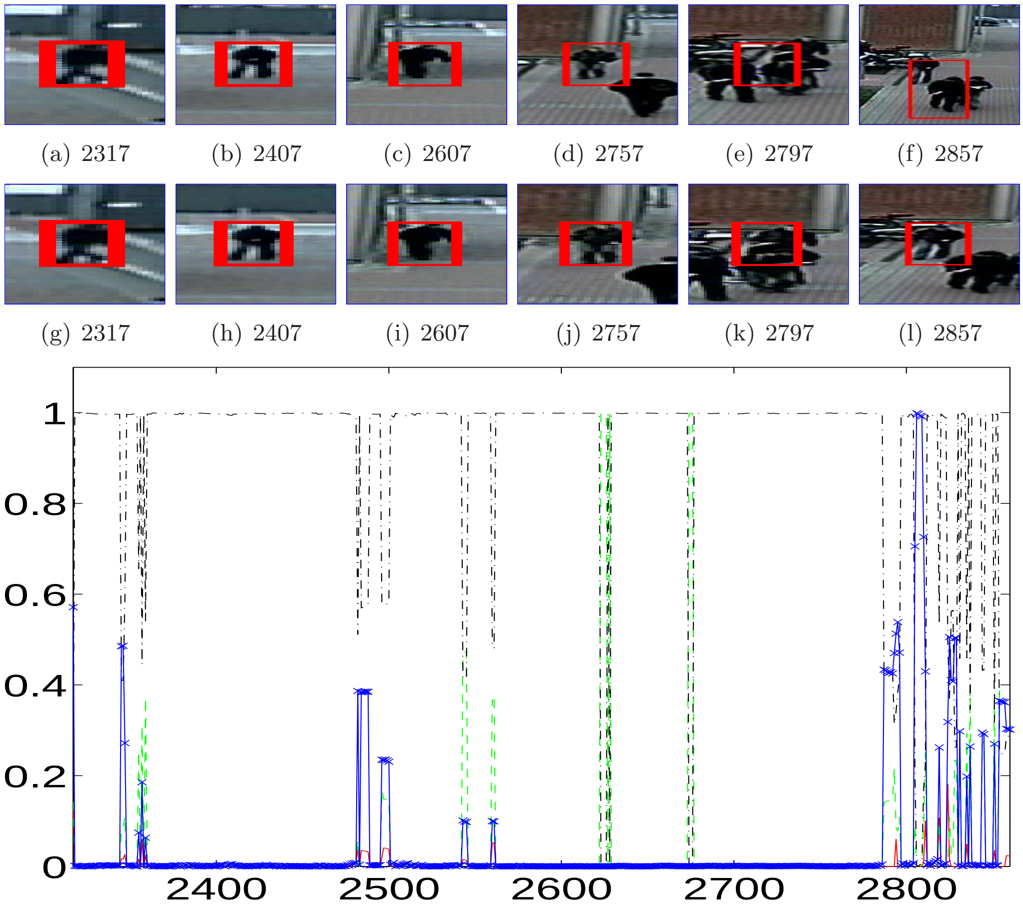
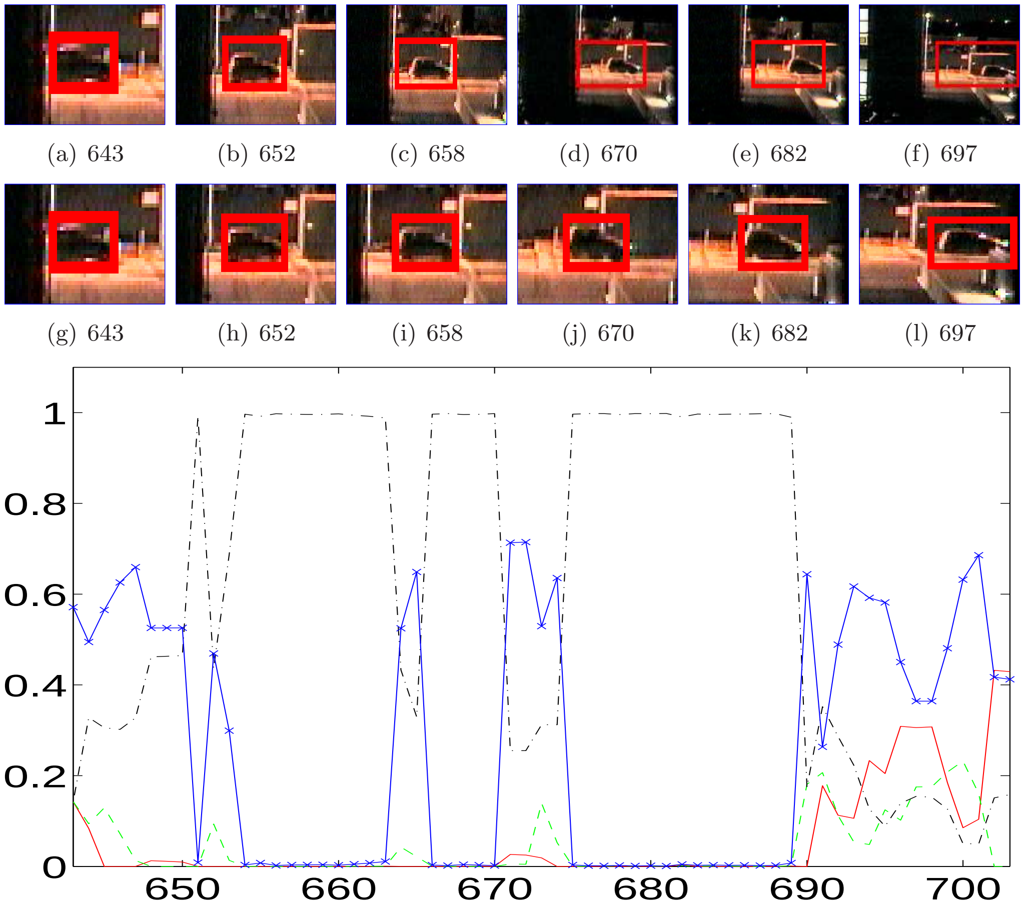


Figure 6.15: Handling background distraction: Tracking results comparing the standard and the adaptive Spatiogram-Bank tracker. Unweighted tracker (top row), adaptive spatiogram-bank (next row). Both trackers used 7 features (H, S, V and 4 gradient histogram features). Weighting for each feature shown in the bottom graph: Hue (red solid line), Saturation (green dashed line), Brightness (black dotted line), Edge features (combined weight shown in blue solid line with crosses). Note that the brightness feature dominates the tracking, so the other features are not seen in this graph.


(a) 2317 (b) 2407 (c) 2607 (d) 2757 (e) 2797 (f) 2857

(g) 2317 (h) 2407 (i) 2607 (j) 2757 (k) 2797 (l) 2857

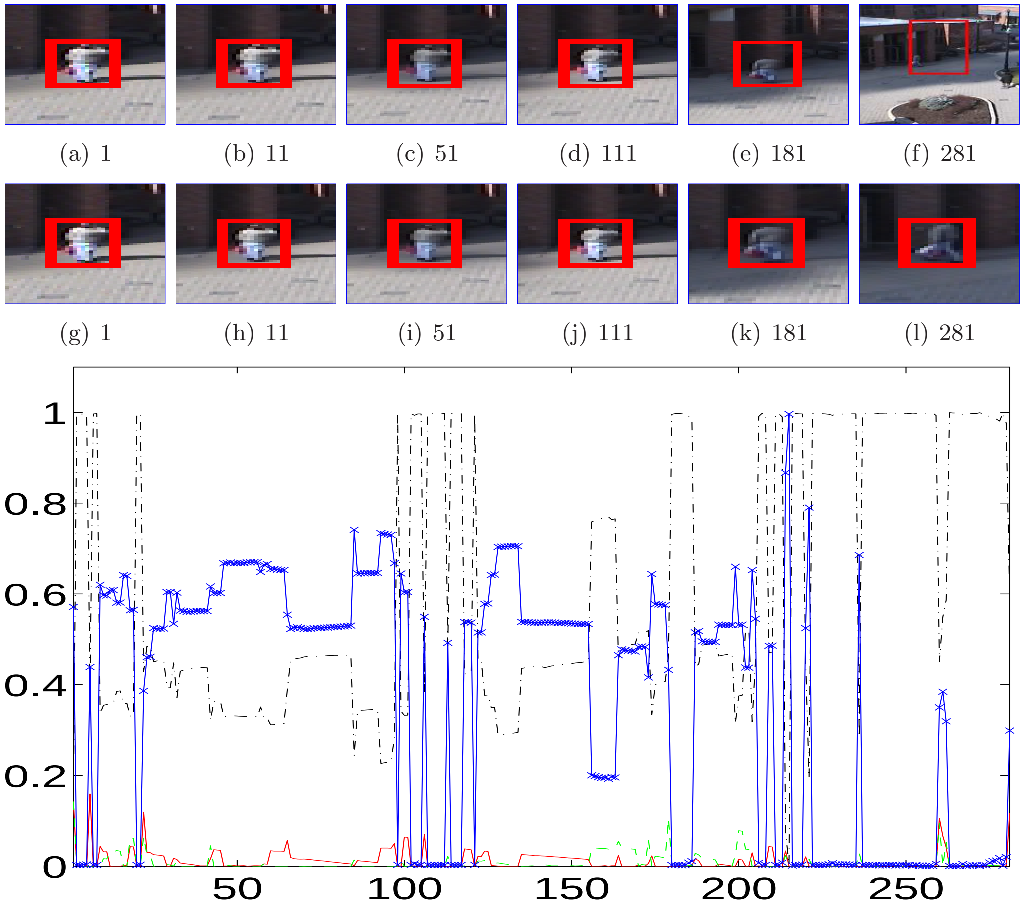
Figure 6.16: Handling occlusion: Tracking results comparing the standard and the adaptive Spatiogram-Bank tracker. Unweighted tracker (top row), adaptive spatiogram-bank (next row). Both trackers use 7 features (H, S, V and 4 gradient histogram features). Weighting for each feature shown in the bottom graph: Hue (red solid line), Saturation (green dashed line), Brightness (black dotted line), Edge features (combined weight shown in blue solid line with crosses).



(a) 643 (b) 652 (c) 658 (d) 670 (e) 682 (f) 697

(g) 643 (h) 652 (i) 658 (j) 670 (k) 682 (l) 697

Figure 6.17: Noise handling: Tracking results comparing the standard and the adaptive Spatiogram-Bank tracker. Unweighted tracker (top row), adaptive spatiogram-bank (next row). Both trackers use 7 features (H, S, V and 4 gradient histogram features). Weighting for each feature shown in the bottom graph: Hue (red solid line), Saturation (green dashed line), Brightness (black dotted line), Edge features (combined weight shown in blue solid line with crosses).


(a) 1 (b) 11 (c) 51 (d) 111 (e) 181 (f) 281

(g) 1 (h) 11 (i) 51 (j) 111 (k) 181 (l) 281

Figure 6.18: Changing lighting: Tracking results comparing the standard and the adaptive Spatiogram-Bank tracker. Unweighted tracker (top row), adaptive spatiogram-bank (next row). Both trackers use 7 features (H, S, V and 4 gradient histogram features). Weighting for each feature shown in the bottom graph: Hue (red solid line), Saturation (green dashed line), Brightness (black dotted line), Edge features (combined weight shown in blue solid line with crosses).

G, B and IR. The search method was the same as in the previous section.

Pedestrian in black Figure 6.20 shows the Collins tracker failing to track a girl wearing a black coat, as she walks in front of a bike rack, while the proposed adaptive tracker successfully tracks her movement. The girl is easily tracked in infrared, and as such, the proposed adaptive tracker assigns all the weight to IR until frame 1450, when a gradual hand-off to the colour RGB features occurs. After this point, the colour features perform robust tracking since the background is uniformly coloured (see frame 1519).

Up until frame 1429, of the $K = 3$ features it uses, the Collins tracker consistently chooses 2 features: I and $3I - R - G - B$. As the girl nears the end of the bike-rack, there is a person standing in it, acting as a distractor. As a result, the peak-difference of the IR-based features decreases, leading to colour features to be chosen instead. The 3 features chosen in frame 1429 are I, $R-G-2B$ and $R-2G-B$. The peak-difference of IR drops from a previous average of about 0.75 to 0.23. The peak-difference of the visual features is only 0.055. The Collins tracker is forced to choose $K = 3$ features, even if some of these feature degrade tracking. The tracker loses the girl and remains tightly fixed to the bike-rack. This drawback of the Collins approach was overcome by replacing the last two IR features ($3I - R - G - B$ and $I + R + G + B$) with I and I. This meant that if I was the highest-ranked feature, then only I would be used for tracking and this allowed Collins to succeed on this sequence. For the remaining examples that use infrared, this modification to the Collins method was used.

Colour face tracking In this test, no infrared features were used for either tracker, only colour features. Figure 6.21 shows tracking results of the Collins tracker and the proposed adaptive tracker. All subfigures are close-ups of the current tracker bounding-box, cropped from the full frame. Since the colours of the face are individually to be found outside the bounding box, it is difficult for the Collins method to discriminate the object and background. Black pixels make up a significant part of the face, but only a small fraction of the lower background, so the tracker enlarges the bounding box scale. Eventually, the nearby person turns their head, essentially causing an occlusion, which causes further expansion of the tracking box. The Collins method takes into account only distractors at the object scale. Figure 6.19 illustrates why this approach can miss potential distraction from the object itself at different scales. In general, when the Collins tracker is initialised on a face, it will expand to track the entire head, since pixel information alone is not enough to discriminate the face from the surrounding area.

The adaptive tracker has no trouble keeping a good lock, since it exploits spatial information and not just pixel information. It uses primarily the brightness feature, but uses also the hue feature around frame 185 when the other person turns away, and is thereby no longer a distractor.

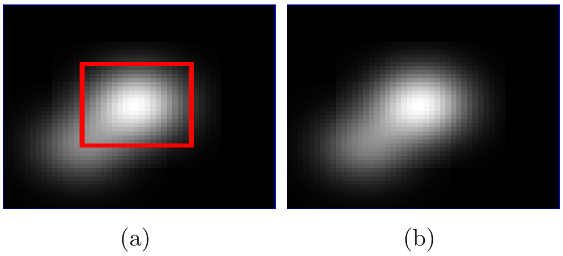


Figure 6.19: Illustration of Collins scale distraction: in figure (a) is shown a weight image generated by a tuned-feature using the Collins method. By comparing the central peak to the peak outside the bounding box, it is clear that the object-to-distractor ratio is large and it is likely that the Collins method would select it as a viable tracking feature. (b) shows the same weight image (only without the bounding box). It is clear that although there are no nearby distractor peaks, the object peak itself has essentially widened which will lead to an increase in tracking scale. Here, the distractor is a larger scale version of the object itself and the Collins method does not account for this, examining only spatial distractors.

Cyclist tracking In figure 6.22 a cyclist is tracked as they move towards the camera, then veering to the left past the bikerack. Just where the cyclist turns, another person is standing in the bikerack. Due to their strong IR features, the Collins tracker jumps to this person instead of the cyclist, as colour pixel feature alone are not enough to distinguish the cyclist from the bikerack; both are composed of primarily black and white pixels. The adaptive spatiogram-bank tracker uses a combination of colour and IR features to distinguish them. This is indicated by the mixture of weights used between frame 1690 and 1705. Between frames 1580 and 1650, colour features are heavily weighted, since the object is on an untextured uniform background, making it easy to track. When cycling over the bike-rack, only the infrared feature is used, due to the complexity of the background.

Car tracking In a sequence from the VACE dataset, a car is seen from an aerial perspective in figure 6.23. The individual features used by the Collins tracker are not sufficient to distinguish the object from the very similar background. The weight images used by the Collins tracker for the first frame in this sequence are shown later in this chapter, in figure 6.34(g). The adaptive tracker uses a combination of multiple features to differentiate the object and background, illustrated by the early part of the weighting graph, where both IR and colour features are weighted. The Collins tracker quickly fails, whereas the adaptive spatiogram-bank tracker succeeds in tracking the car until it parks.

T-shirt logo In the final illustrative tracking comparison, shown in figure 6.24, a t-shirt logo is tracked. As the surrounding area is also part of the person's body, there is no temperature difference and therefore IR features do not aid in tracking. The complex logo is made up of dark and bright pixels, but without spatial information, it is difficult for the Collins tracker to keep a lock and gradually slides off the logo. It eventually locks onto an area of the shoulder and wall, containing bright and dark pixels. The adaptive tracker performs excellently, using primarily blue colour features, but also using IR to discriminate the logo from the hands during the partial occlusion around frame 559.

6.4.5 Comparison to histogram- and template-tracking

In this section, the adaptive spatiogram-bank tracker is compared briefly to histogram-based tracking and template-based tracking. The histogram-based tracker is the standard mean-shift tracker [22] and uses an $8 \times 8 \times 8 \times 8$ histogram in $RGBI$ space. The template tracker represents the object as a 4-band image and finds the best match in subsequent frames by minimising the sum-of-squared-difference (SSD). To cater for object pose changes, both trackers used the *mixture* update strategy. That is, the model used in each frame is an equal-parts mixture of the original model and the last best match. Here, *mixture* refers to averaging the histograms or templates. Full details of the trackers are given in section 6.4.6, where the full set of extensive tests are described.

Illustrative results are shown in figure 6.25 for the tracking of a vehicle. This sequences is from the VACE dataset and was captured from an unmanned aerial vehicle (UAV). In the top row, the mean-shift histogram tracker is shown to fail quickly. Since it encodes no spatial information, it is attracted to the roadside which is of similar colour to the car. It expands in scale and quickly fails. The template tracker does well until the car reaches a junction, just before frame 202. Since the right bank of the roadside is 'missing', it slides off to the left, losing track in frame 210. The proposed adaptive tracker successfully tracks the vehicle through the junction.

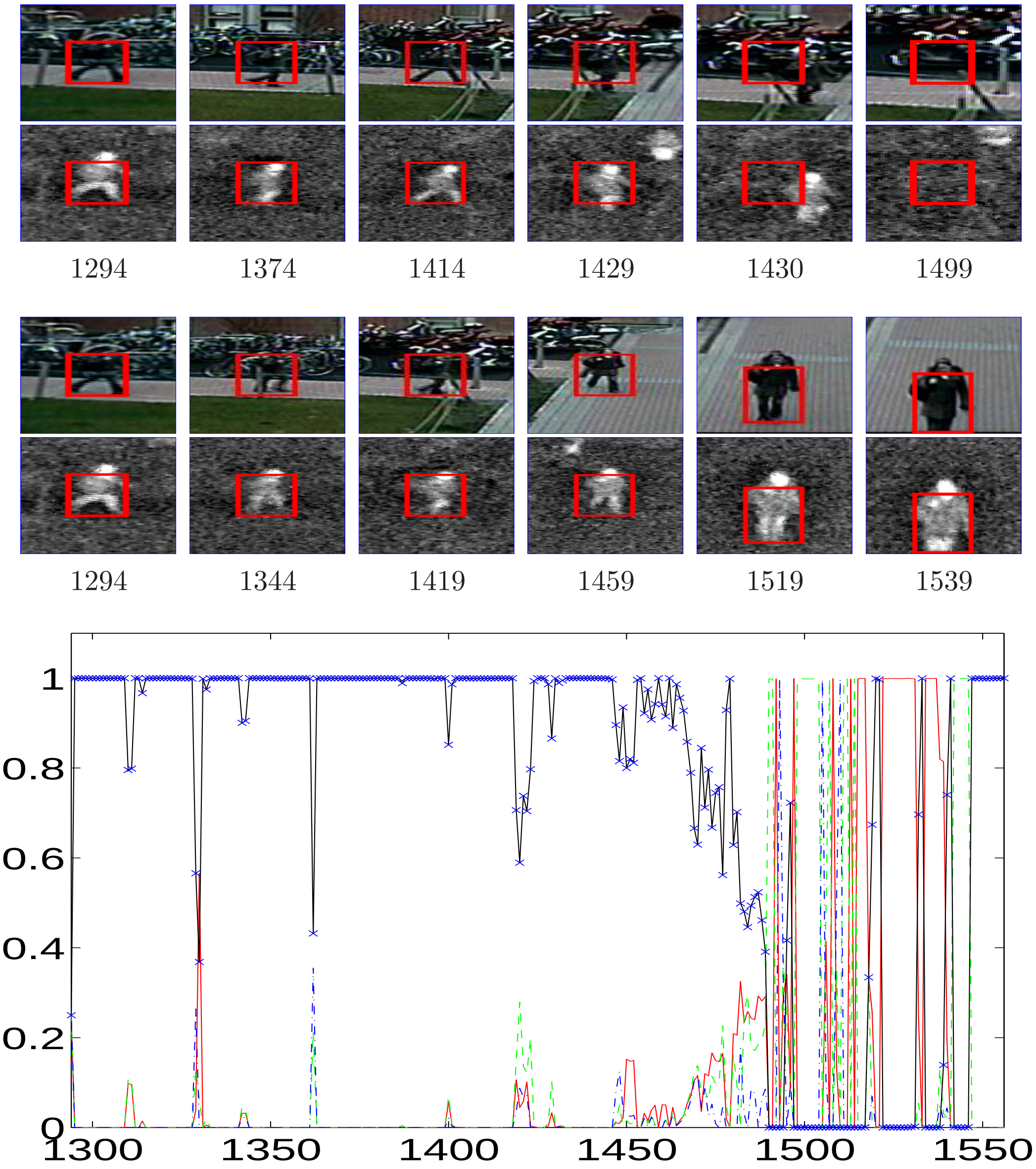


Figure 6.20: Person tracking comparison: Collins tracker (top 2 rows), adaptive spatiogram-bank using R, G, B and IR features (next 2 rows). Weighting for each feature shown in the graph: Red (red solid line), Green (green dashed line), Blue (blue dotted line), IR (solid black line with crosses).

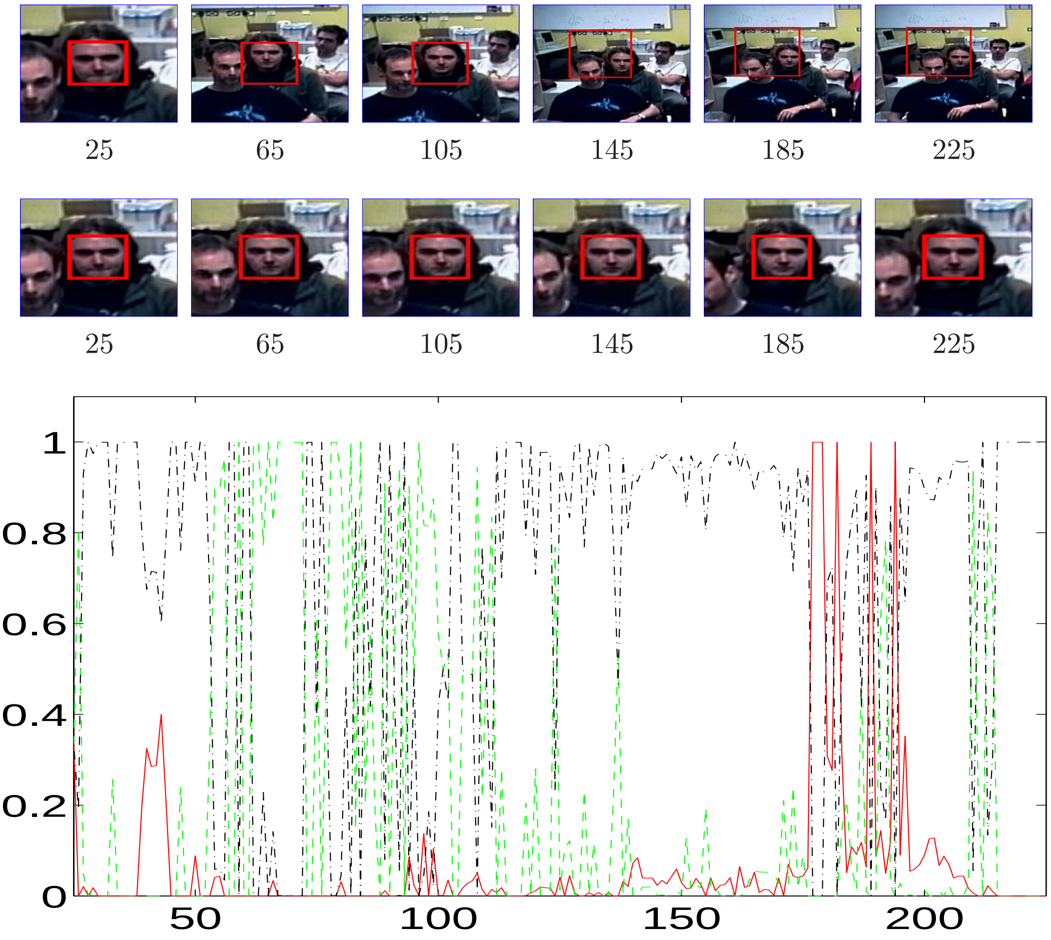


Figure 6.21: Colour tracking comparison: Collins tracker (top row), adaptive spatiogram-bank using H, S and V features (next row). No IR features were used by either tracker. Weighting for each feature shown in the graph: Hue (red solid line), Saturation (green dashed line), Brightness (black dotted line).

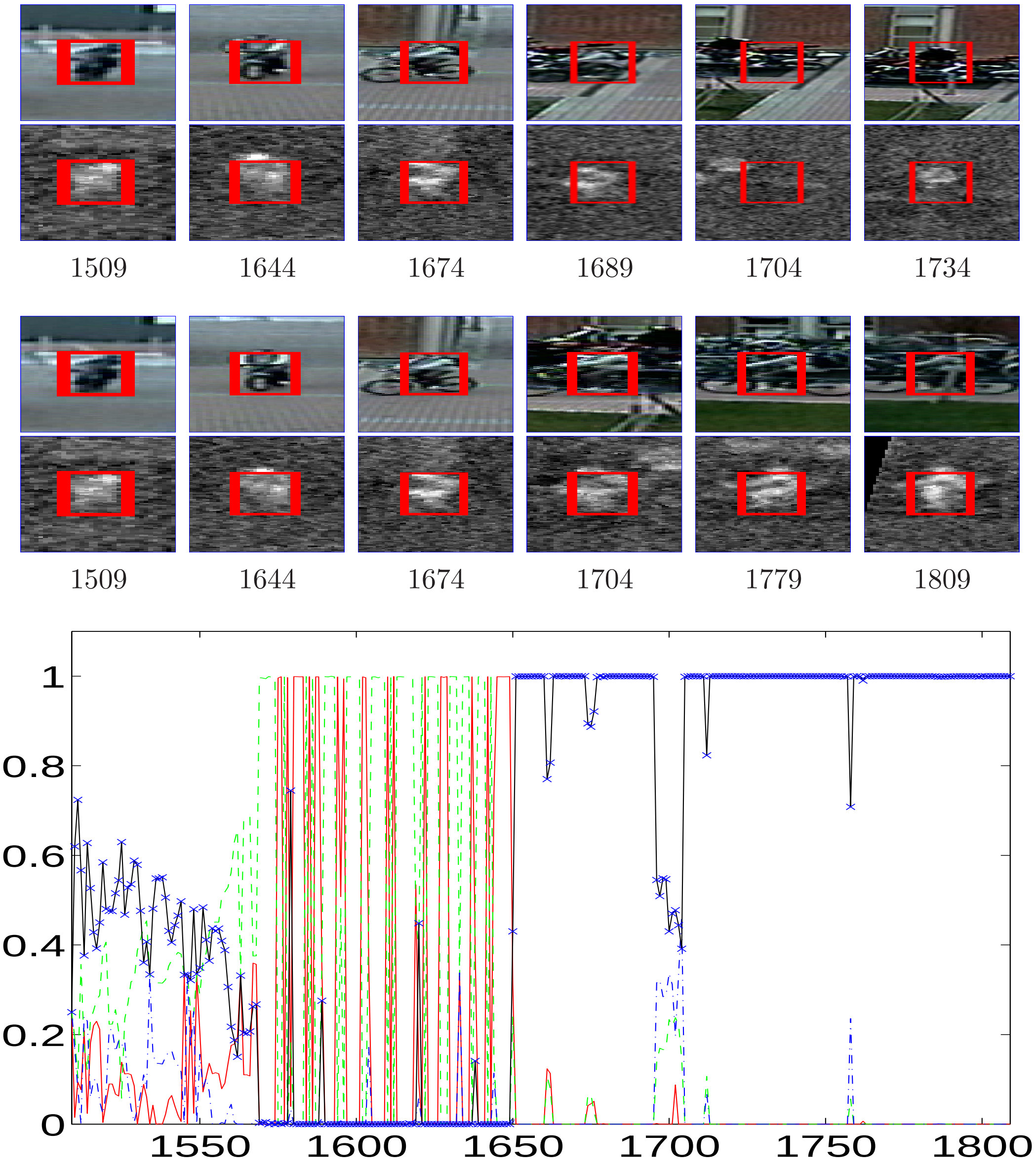


Figure 6.22: Cyclist tracking comparison: Collins tracker (top 2 rows), adaptive spatiogram-bank using R, G, B and IR features (next 2 rows). Weighting for each feature shown in the graph: Red (red solid line), Green (green dashed line), Blue (blue dotted line), IR (solid black line with crosses).

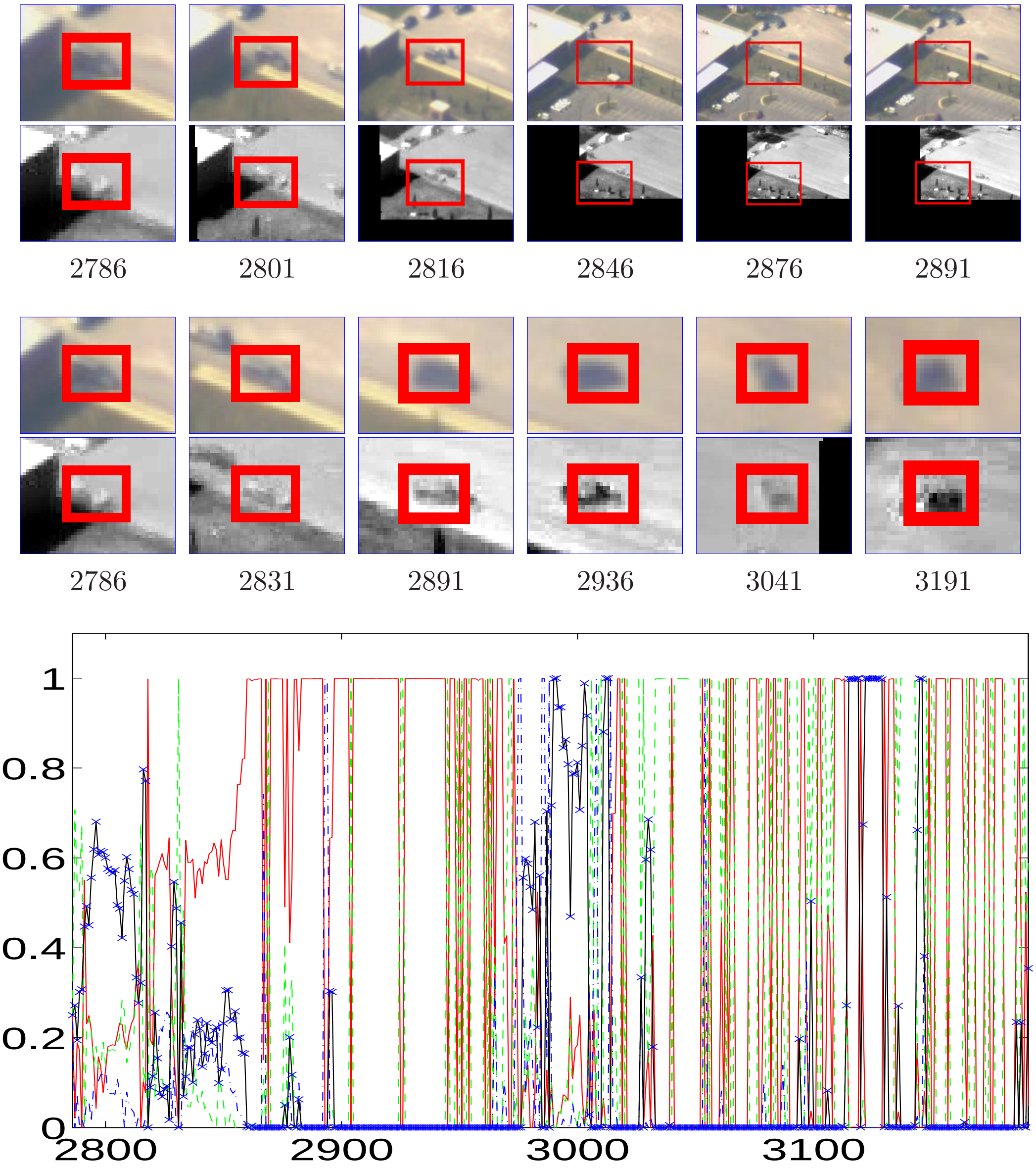


Figure 6.23: Vehicle tracking comparison: Collins tracker (top 2 rows), adaptive spatiogram-bank using R, G, B and IR features (next 2 rows). Weighting for each feature shown in the graph: Red (red solid line), Green (green dashed line), Blue (blue dotted line), IR (solid black line with crosses).

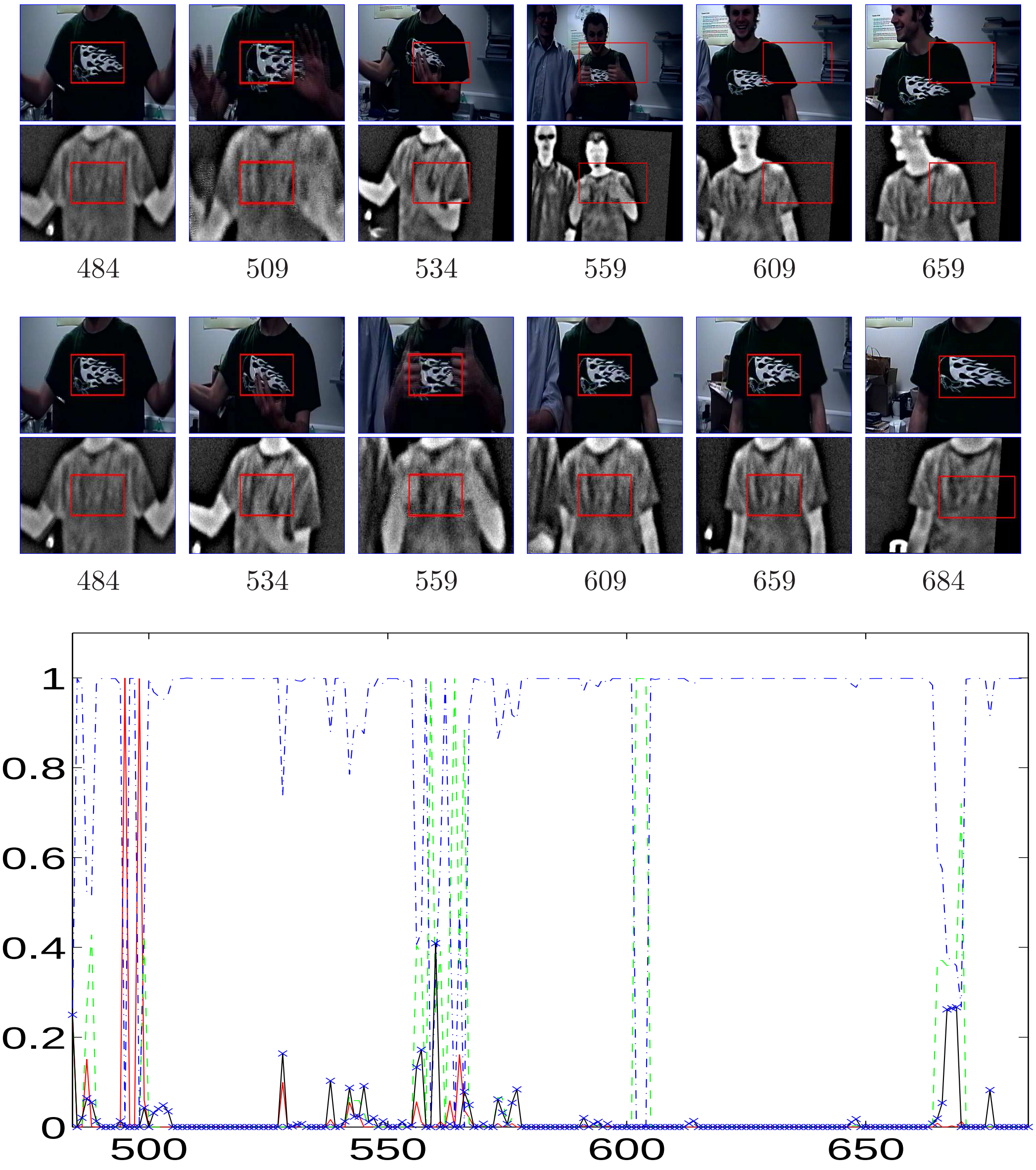


Figure 6.24: Logo tracking comparison: Collins tracker (top 2 rows), adaptive spatiogram-bank using R, G, B and IR features (next 2 rows). Weighting for each feature shown in the graph: Red (red solid line), Green (green dashed line), Blue (blue dotted line), IR (solid black line with crosses).

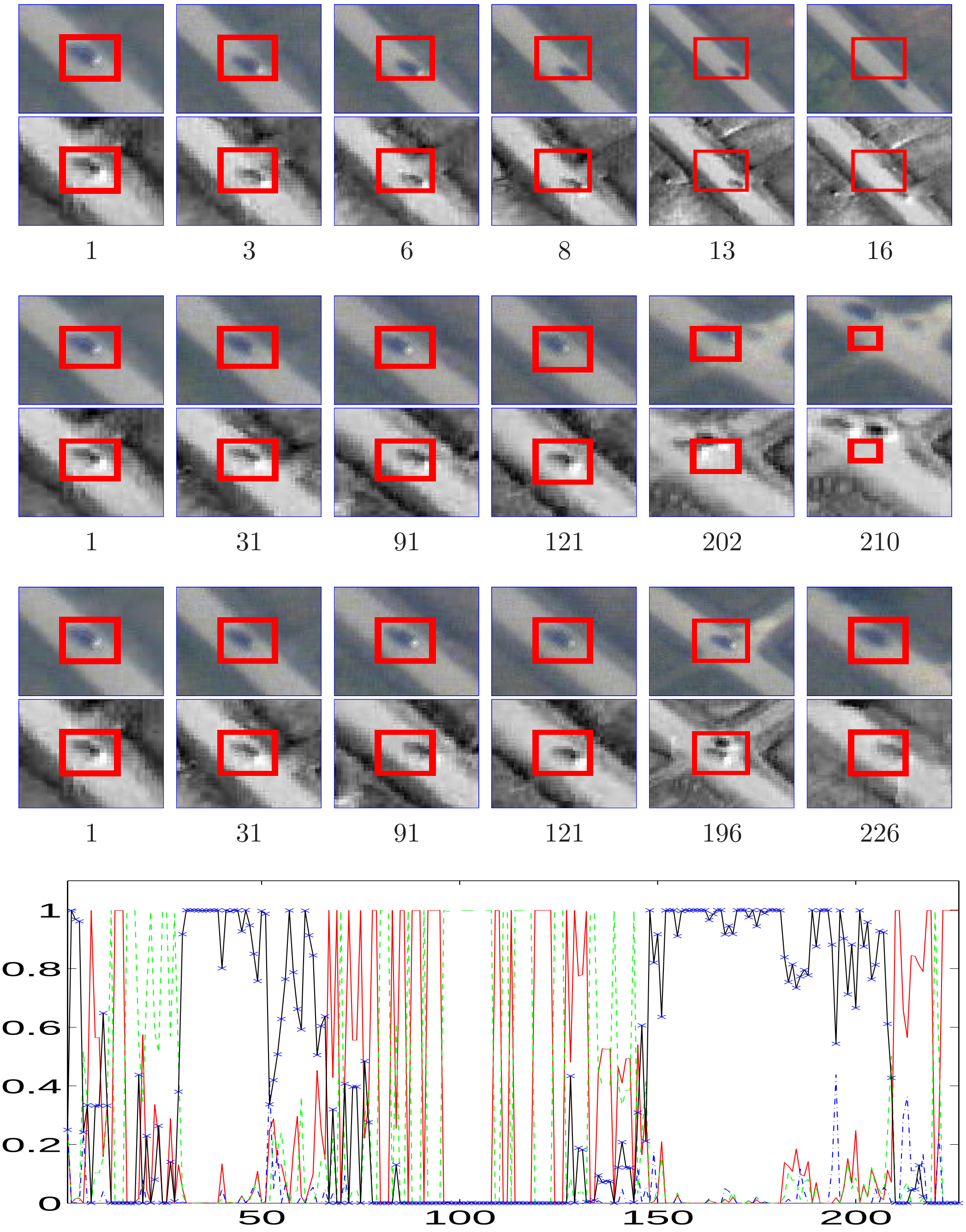


Figure 6.25: Tracking results comparing the adaptive Spatiogram-Bank tracker to mean-shift and template tracking: Mean-shift tracker (top 2 rows), Template tracker (rows 3 and 4), adaptive spatiogram-bank using R, G, B and IR features (rows 5 and 6). Weighting for each feature shown in the graph: Red (red solid line), Green (green dashed line), Blue (blue dotted line), IR (solid black line with crosses).

6.4.6 Tracker comparison

In the final experiment, an extensive comparison of various trackers is conducted on multiple thermo-visual video sequences. The proposed adaptive tracker is compared to a number of other trackers, namely (i) a template tracker, (ii) a mean-shift histogram tracker, (iii) the Collins adaptive tracker and (iv) the spatiogram-bank tracker with fixed weights. Examples of the objects used in the experiment are shown in figure 6.26. A total of 41 tracking sequences are used, with an average sequence length of 290 frames and a maximum length of 707 frames. The average object width and height are 33 pixels and 44 pixels respectively. The sequences were primarily captured using the thermo-visual capture rig that was built as part of this work, but the dataset also includes objects from the OTCBVS [27] and VACE [66] datasets. The operation of each tracker is now described.

The template tracker object representation is simply a 4-band image of the object, composed of 3 colour bands (RGB) and a thermal infrared band. The template is compared to candidate regions using the standard sum-of-squared-differences (SSD) measure. An exhaustive search in a 13×13 window is conducted at three scales ($\pm 10\%$) in each frame. The rectangle with the lowest SSD is selected as the correct match. To give the tracker some flexibility to adapt to pose changes, the template used for tracking in each frame is a straight-forward average of initial template and the best match in the last frame.

The mean-shift histogram tracker used is based on the seminal paper of Comaniciu et al. [22] where the object is represented by a histogram. Here, an $8 \times 8 \times 8 \times 8$ histogram is used for the 3 colour channels (RGB) and the infrared channel. It was necessary to coarsely quantise the colour values to reduce computational time, but primarily to prevent the histogram being too sparse and hence causing the tracker to fail due to the curse of dimensionality. The mean-shift procedure is repeated at three scales (original scale $\pm 10\%$) in each frame, and the scale with highest similarity to the histogram model (using the Bhattacharyya coefficient) is selected. Similarly to the template tracker, the histogram model is allowed to adapt. The model used for tracking in each frame is an average of the original histogram and the best match in the previous frame. The use of HSV features for mean-shift and template tracking was also investigated, but these features proved detrimental to tracking performance overall.

The Collins adaptive tracker worked as described earlier. To select background samples, a bounding box of 2.5 times the object width/height was centred on the object, and pixels which did not correspond to the object were considered background pixels. A total of 52 features were evaluated in each frame; the original 49 linear combinations of R, G and B, as well as 3 infrared based features. In initial tests, these features were set as I, $3I-R-G-B$ and $I+R+G+B$, which seemed reasonable, but this resulted in very poor performance. Instead, the 3 infrared-based features used are I, I and I. As explained in section 6.4.4, it is important to add the infrared brightness 3 times. In night-time sequences for example, the Collins method would choose $K=3$ features and if only one infrared feature was available, then 2 useless visual features would cause tracking failure. The maximum distractor method was used and the number of features for tracking was set at $K=3$. For each feature, the mean-shift procedure was run at three scales (±10% of object radius) and the scale returning the highest sum of weights from the weight-image was selected as the correct scale. Another possible scale selection criteria used in [19] was not implemented. 32 bins were used for feature binning. The original work does not explicitly discuss how the object scales returned by tracking features should be fused. In these experiments, the naive median was used to fuse the returned widths and heights, as was already done with the x and y position.

The adaptive spatiogram-bank tracker is composed of 4 spatiograms, using the R, G, B and I pixel features. 32 bins were used for each spatiogram. To locate the object, a coarse-to-fine search in a 13×13 window is conducted spatially, then at three scales (±10% of the object radius) in each frame. The product updating strategy was used to cater for gradual changes in the object appearance. The bounding box with the greatest similarity is selected as the correct match. Weights are updated in each frame using the proposed weight selection strategy. The non-adaptive spatiogram-bank tracker is identical, except that all weights are fixed at 0.25.

For every tracking experiment, each tracker is initialised by supplying a manually annotated bounding box on the object to be tracked. In a practical system, this bounding box would be supplied by a detection algorithm, or by some other means. Ground truth tracking was manually annotated for each sequence using a Matlab-based annotation tool. For all trackers, simple linear prediction is used to centre the search window on a probable location of the object. This is done by fitting a line to the plot of the last four values of each parameter (x, y, $width$, $height$) using a least-squares fitting. Figure 6.27 shows the results of the evaluation, using a similar plot to the one used in section 6.3.2 to display multiple tracking results. To robustly measure model stability, a tracker was deemed to have failed if its F_1 score dropped below the failure threshold for 25 consecutive frames. The failure threshold is shown on the x-axis, and on the y-axis is shown the average percentage of a sequence that is tracked before failure.

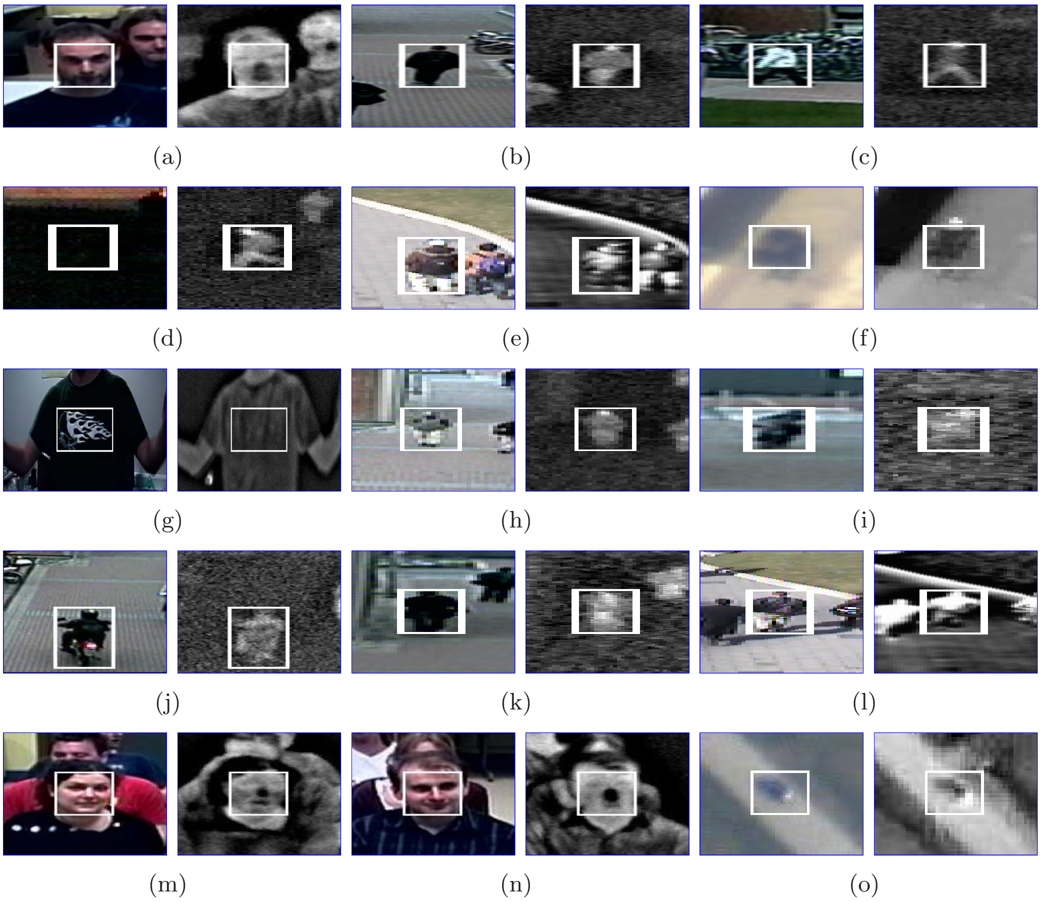


Figure 6.26: Examples of the thermo-visual objects used in the tracking experiment. Each object is shown in the visible spectrum and in thermal infrared.

Thermo-visual tracking Discussion As figure 6.27 demonstrates, the adaptive spatiogram-bank tracker outperforms the non-adaptive tracker, as it can robustly select the best features for tracking, avoiding features that might cause distraction. Both methods outperform the standard mean-shift and template tracking approaches. The Collins method performs slightly worse than the mean-shift tracker. Some of the causes of its failure, discussed in earlier sections, are examined in more detail later.

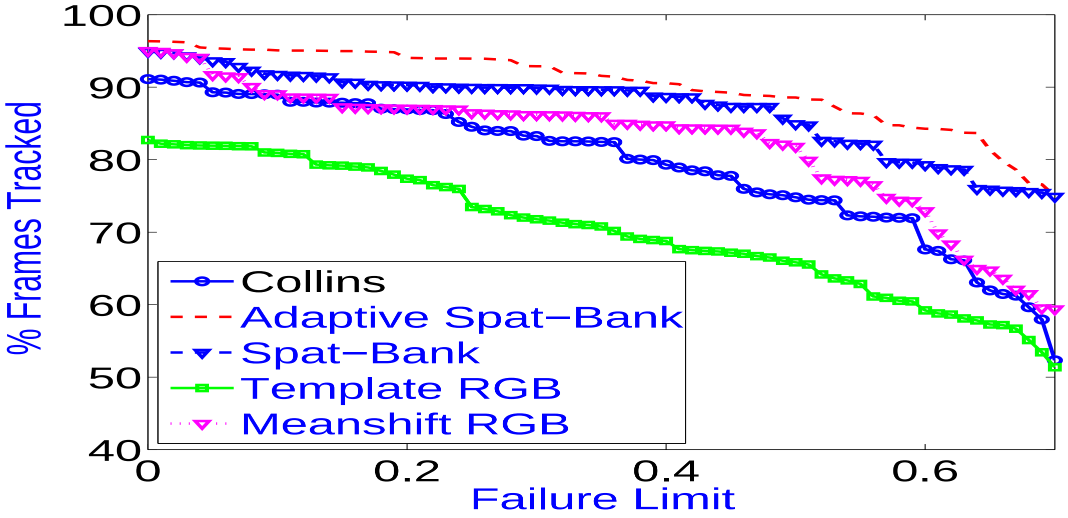


Figure 6.27: Thermo-visual tracking evaluation results graph: using 41 thermo-visual sequences

The mixture update strategy was used for both the mean-shift and the template tracker in figure 6.27. Figure 6.28 compares the results of using *no update* and the mixture update strategy. While there is no clear winning strategy, it appears that the updating gives a slight improvement in tracking precision for both trackers.

Visual spectrum tracking In some sequences, the use of infrared alone is enough to provide good tracking. This is because of the strong brightness difference than exists between some objects and the background, due to their temperature difference. In the tests described above on the 41 thermo-visual sequences, the proposed adaptive tracker used a mean weight for IR of 0.4716 (median = 0.5071) over the whole set of tests. Similarly, the fraction of all features chosen by the Collins method that were infrared-based is 0.4995 (median = 0.5045) over the whole set of tests.

Since the infrared features performed so well, the use of visual features *without infrared* are investigated. Making the tracking more difficult, by removing infrared, also

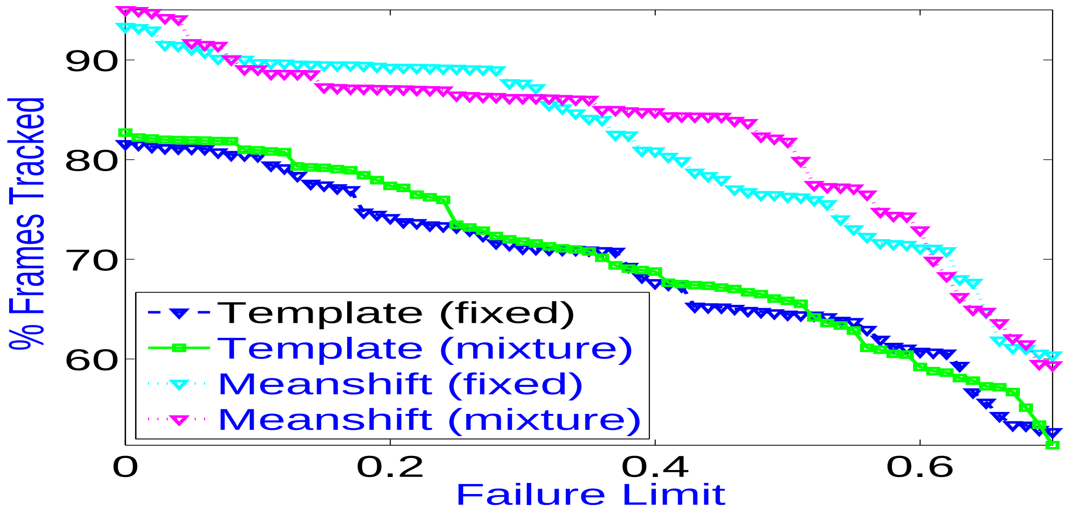


Figure 6.28: Updating strategies for Mean-shift and Template tracking: the curves show the slight improvement in tracking that is achieved using the mixture update strategy, instead of not updating the model. Results shown over the 41 thermo-visual tracking sequences.

allows better discrimination between the trackers' performances. The same sequences were used for these tests, but the infrared data was removed. The trackers had the same parameters and configurations as before, except for the following changes.

The Collins tracker used only the original 49 colour seed features. The mean-shift tracker used an $8 \times 8 \times 8$ RGB colour histogram. The template-tracker used a 3-band RGB template to represent the object. The proposed adaptive tracker used 10 features: R, G, B, H, S, V and the 4 edge orientation histogram features. The non-adaptive tracker was identical, but used equal weights for all features.

Figure 6.29 compares the adaptive and non-adaptive spatiogram-bank trackers. Clearly, the adaptive approach gives superior tracking. The largest increase in performance is seen when failure-limits of between 0.4 and 0.6 are used. These limits are the most reasonable, as lower limits can be achieved when the tracker has changed to a large scale and high limits (> 0.8) are not reliable since they depend on the imperfect manual annotation of the ground-truth. Figure 6.30 shows the tracking results of others trackers, compared to the proposed approach, over the 41 sequences using visual features only. In this plot, the performance benefit of the proposed adaptive tracker is much more clear than in the previous experiment which used thermo-visual sequences. The mean-shift tracker performs badly in these tests without the use of infrared. The

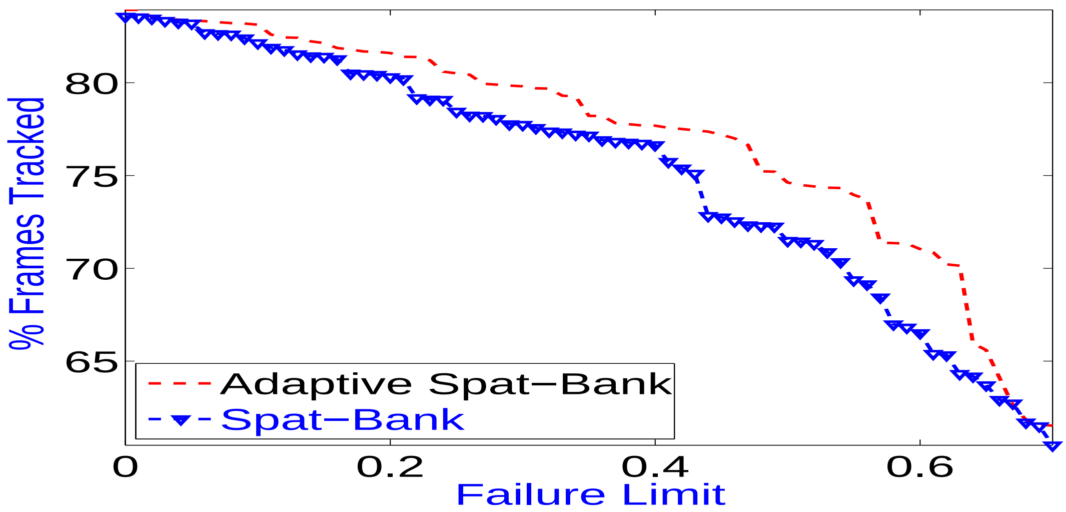


Figure 6.29: Visual spectrum tracking evaluation : comparing non-adaptive and adaptive spatiogram-bank tracking. Both trackers use only visual features in all 41 tracking sequences.

Collins method performs slightly better. Some of the causes of Collins failure are now examined.

Both the Collins and Avidan tracking methods create weight images for tracking in each frame. These weight images emphasise the most *trackable* features and cause any pixels that might resemble the background, including other parts of the object, to be down-weighted. This leads to a difficulty in determining the correct object scale.

As a practical example of the Collins tracker's drawbacks, if the tracker is initialised on a human face, without spatial information the tracker can fluctuate its position or slide downwards, since the neck and forehead are similar uniform skin colour regions. This makes the tracker more vulnerable to added distraction, as a hand placed under the chin may be tracked instead. Figure 6.31 shows such an example, where pixel information is not sufficient to distinguish the object that is to be tracked from the surroundings. In scenarios where the light source is above the subject, faces appear to be very bright at the top and dark near the chin. In these cases, it was found that the Collins tracker would slide upwards or downwards, depending on whether the bright or dark pixels were more heavily weighted in the weight image. Spatiogram-based tracking was found to remain in a fixed position on the object. This finding agrees with the original findings of Birchfield and Rangarajan [11] that spatiograms provide more accurate object tracking than using histograms and ignoring the spatial layout of

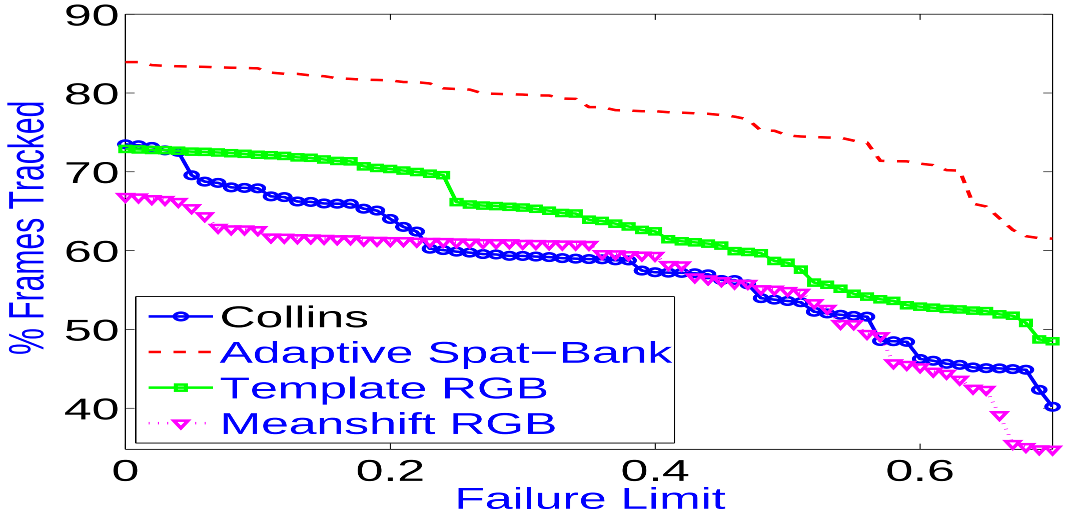


Figure 6.30: Visual spectrum tracking evaluation results graph: using the same 41 sequences as before, but without the thermal infrared channel.

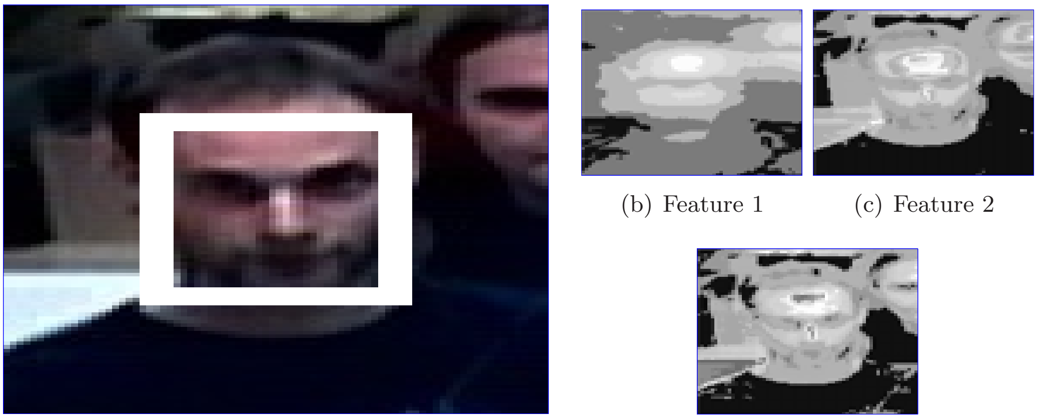
the pixels.

The adaptive spatiogram-bank tracker performs remarkably well, given its limited feature set, when compared to the Collins method. In cases where the features it uses are not appropriate for tracking, the spatiogram-bank tracker may benefit from being embedded in a Collins-like feature selection mechanism. Some directions for future work in this direction are given later in this chapter.

Some examples of failure of the proposed adaptive tracker are now examined, with a view to providing insights into its drawbacks and potential for further improvement.

Failures Figure 6.32 shows an example of tracking failure. HSV Colour and edge features were used in this test. There is an occlusion of the object by a lamppost, but this is not the primary cause of failure. The failure is caused by the large weighting assigned to a single edge feature, namely the 45^o edge feature. In frame 1132, this feature is given a weight of 1, since the person's leg is at a diagonal angle, similar to their pose in the first frame, resulting in the feature being a good object/background discriminator for this frame. As the leg changes angle, and the person is occluded, tracking is eventually lost. Failure might be avoided by using a tighter bounding box in the original frame, since more than half the pixels are part of the background. Other suggestions for improvement are given after the next example.

In figure 6.33 another tracking failure example is given. Similarly to the last ex-



(a) Target object

(b) Feature 1

(c) Feature 2

(d) Feature 3

Figure 6.31: Illustrating the poor scale selection of the Collins tracker: the target object is shown in (a) and the top 3 weight images are shown in (b), (c) and (d). Depending on the scale selection mechanism, the use of these weight images will either lead to (i) expansion of the tracker to merge with the other face, (ii) the tracker shrinking to lose the chin or (ii) the tracker shrinking towards the forehead. Each case leads to a higher potential for distraction from the nearby face. Without spatial context, the face cannot be accurately tracked using the Collins method.

ample, HSV colour and edge features were used for tracking. This time the brightness feature is the culprit. Initially, vertical edges are given a high weight, between frames $50-60$ and $80-100$. This causes slight distraction due to a lamppost in frame 95. During occlusion by another person, all features are used, leading to the tracking regaining a good lock in frame 185. In frame 230 however, the tracker begins to lose its lock due to the change in colour of the ground. The tracker gradually slides off the object and loses track. Interestingly, if R, G and B features are added, the tracking of this objects works, using a mixture of features (dominated by R and vertical edges) during the transition to a different coloured background. This indicates perhaps that the HSV and edge features are not discriminative enough in this example.

Failures also occurred in thermo-visual sequences, but these failures were caused by either severe occlusion, or an inability to distinguish the object from the background with the given features. For example, a darkly coloured car entering a dark shaded region of the road. The sample failures shown in figures 6.32 and 6.33 demonstrate objects that might be tracked if weights were better selected. The non-adaptive spatiogram-bank tracker, weighting the features equally, also fails on these sequences however.

Two of the potential limitations of the proposed adaptive method are due to the feature selection approach, which only uses knowledge from the previous frame. These limitations might become apparent, firstly, during occlusions or secondly, during sudden appearance changes in one or more features.

If the object is occluded, the weights may adapt to emphasise features of the occluding object, causing a tracking loss. Equally weighting features might give some advantage here, as the holistic properties of the object might allow better discrimination from the occluder.

Also, during a sudden appearance change in one feature-space, equal weighting might be advantageous, since the democratic voting helps counter the ‘outlier feature’ that has been affected by the change. It was found that when no strong distractors were present, the proposed approach will usually select a very high weight for the best feature. Were this feature to change dramatically in the next frame, tracking could be lost.

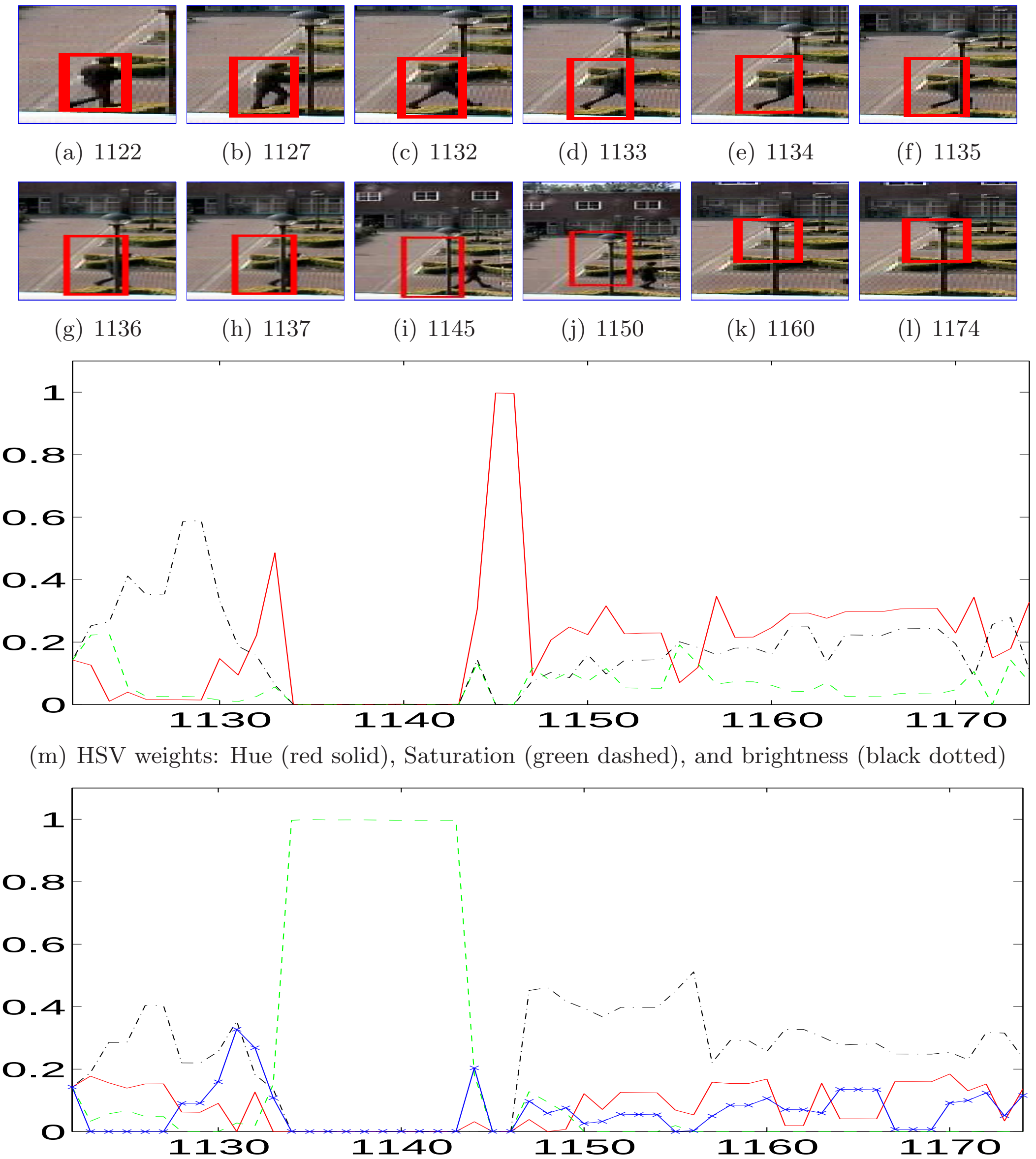
To counter these potential drawbacks of the adaptive spatiogram-bank tracker, further work might incorporate more temporal information into the object model, such as examining the stability of tracking features over time. This might be done by including a third spatiogram-bank model, along with using the original model and the last-best-match. This model would be gradually updated (see gradual update strategy in section 6.3.2), hence keeping a kind of short-term memory of the object appearance. This use of three models would be similar to the approach of [162].

6.5 Conclusion

6.5.1 Chapter summary

This chapter contains three main contributions. Firstly, an improved similarity measure for spatiograms is derived from the Bhattacharyya coefficient. Secondly, a series of object tracking tests are conducted to investigate object model updating strategies. And thirdly, an adaptive spatiogram-bank tracker is proposed that dynamically weights tracking features to improve tracking performance.

In section 6.2, an improved spatiogram similarity measure is derived from the Bhattacharyya coefficient. This similarity measure is shown to provide accurate object localisation and to produce smooth similarity surfaces. Some drawbacks of the original measure are demonstrated analytically and on real data. These drawbacks are shown to be overcome by the new measure, such as the original measure’s low tolerance to spatial changes. The new measure is shown to be superior to the original measure



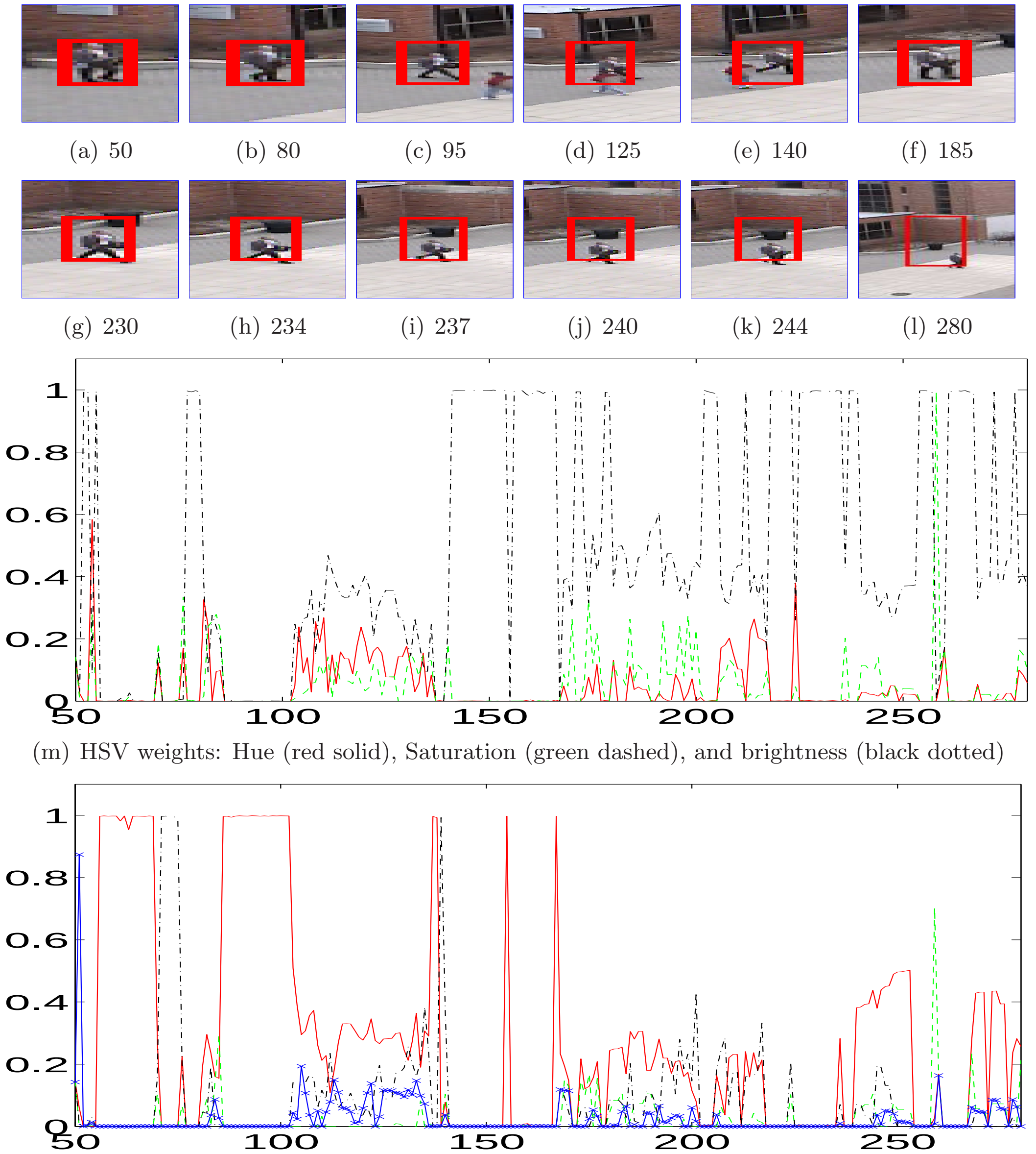
(a) 1122 (b) 1127 (c) 1132 (d) 1133 (e) 1134 (f) 1135

(g) 1136 (h) 1137 (i) 1145 (j) 1150 (k) 1160 (l) 1174

(m) HSV weights: Hue (red solid), Saturation (green dashed), and brightness (black dotted)

(n) Edge weights: vertical (0^o, red solid), 45^o (green dashed), 90^o (black dotted), 135^o (blue crosses)

Figure 6.32: Sample tracking failure using adaptive tracker: the 45^o diagonal edge feature becomes dominant in frame 1132 due to its ability to distinguish the current object and the original object from the surrounding area. Notice the similarity between the leg angle in frame 1122 and 1132. At 1145, the features change, since the 45^o edge feature is no longer distinguishing the object, but by then tracking has failed.



(a) 50 (b) 80 (c) 95 (d) 125 (e) 140 (f) 185

(g) 230 (h) 234 (i) 237 (j) 240 (k) 244 (l) 280

(m) HSV weights: Hue (red solid), Saturation (green dashed), and brightness (black dotted)

(n) Edge weights: vertical (0^o, red solid), 45^o (green dashed), 90^o (black dotted), 135^o (blue crosses)

Figure 6.33: Sample tracking failure using adaptive tracker:

for mean-shift tracking and to provide more accurate localisation than a histogram descriptor.

Tracking tests are conducted in section 6.3.2 investigating strategies for updating the object model during tracking. It was shown that by anchoring the tracking model to the original model from the first frame, object drift can be avoided.

A method for adaptively weighting the tracking features is proposed in section 6.4. By coarsely sampling the area around the object, background samples are obtained. Feature weights are selected to maximise the ratio of the object's score to the score of the maximum distractor. The optimisation in weight-space is achieved using a gradient ascent approach with multiple initialisations. The multiple weight-sets that are returned by the initialisations are pruned by selecting only those that cause minimal scale change. Numerous illustrative examples demonstrate the advantages of the dynamic weight selection approach, compared to the non-adaptive spatiogram-bank tracker, as well as the Collins tracker. Finally, an extensive series of tracking experiments were conducted using 41 thermo-visual tracking sequences comparing the proposed adaptive approach to multiple alternative tracking methods.

6.5.2 Future work

The work in this chapter, as well as providing an insight into the usefulness of the spatiogram-bank tracker, has opened other potential avenues of investigation. A number of interesting directions for future work are now discussed.

Handling unknown data When thermal and visual images are aligned, there are usually some boundary pixels that do not have corresponding pixels in one of the modalities. The homographic warping to align the images will leave some pixels that are not overlapping. An interesting direction for future work will be to examine how to cater for this *unknown* data during tracking. Should infrared be ignored if it is missing from some part of the search window? Are there ways to estimate the missing values? These are both valid questions for later work.

Particle filtering While the experiments in this chapter used a fixed window size in which to search for the tracked object, other more efficient methods of searching the parameter space may be used. For example, in [152], Wang et al. embed the feature selection process in a particle filter. Particles that do not correspond to the object are used as *background particles* and features are selected in an online manner to best distinguish object and background particles. Haar-wavelet features are used in their work. The adaptive spatiogram-bank tracker could similarly be used in a particle filter to more widely explore the search space. If object orientation was included as a parameter, it would increase the parameter space and such a search strategy would be required. Also, if the current object becomes occluded, the noise variance can be increased to allow the particles to cover a wider area in searching for the object, as was done in [162].

Spatial layout matching A fascinating area of future work lies in performing spatiogram similarity matching using only the spatial information and discarding, or down-weighting, the feature information. For example, in image matching it might be desirable to find images that have a similarly layout of colours, but not necessarily the same colours. Landscape scenes would usually be composed of sky in the top half of the image and grass or mountains in the bottom half. The colour of the sky and landscape might change but the spatial layout may turn out to be a robust descriptor. One method of comparing two N-bin spatiograms in this manner would be to compute the volume of overlap between the Gaussian models in each pair of bins, generating an $N \times N$ matrix. When normalised, the mutual information in this *joint distribution* might be a good spatial layout descriptor.

Adaptive quantisation In binning pixels for histograms or spatiograms, equal width bins were used to quantise pixel values. To discriminate between two classes, such as the object and background, non-linear value binning can improve this discrimination without the need to increase the number of bins. One such example is the method of Fayyad and Irani [36] that iteratively selects bin boundaries by minimising a class-based entropy function. A minimum description length principle is used as a stopping criteria. This use of non-linear quantisation for *bin-based* tracking is another possible direction for future work.

Other similarity measures In this chapter, the Bhattacharyya coefficient was derived for Spatiograms and was used as a measure of comparison between Spatiograms. Other measures of similarity are also possible and we discuss some of them here. A possible direction for future work might be to investigate which measures provide the best trade-off between tracking robustness and computational efficiency.

In comparing two distributions, the histogram intersection has frequently been used. It can be shown that this measure is equivalent to the probability of mistaking a pixel coming from one distribution as coming from another, given equal priors. In the one-dimensional case, two normalised histograms, h_X and h_Y, can be compared as follows:

$$H(X,Y) = \sum_{b=1}^{B} \min\{h_X(b), h_Y(b)\} \tag{6.23}$$

Since the min function is used, the first derivative is not continuous, so may not have a smooth similarity surface suitable for efficient search. Other measures of distribution similarity (or dissimilarity) include the KL-divergence, the chi-squared measure, as well as the L_n norms. Examples of such norms include the L_2 norm, which is the Euclidian distance, and the L_1 norm, known as the Manhattan (or city-block) distance.

As briefly mentioned in chapter 2, there are other measure that take adjacent bin similarity into consideration. This property is important in gracefully reducing similarity during lighting changes, for example. The *earth-mover's distance* and the diffusion distance are two examples of such measures. These measures are more computationally challenging for tracking, since they must examine bin-pairs, although the diffusion distance has been shown to be less challenged in this respect compared to the earth-mover's distance. Both methods have been shown to outperform bin-wise similarity measures, such as the Bhattacharyya measure, in certain cases [80].

Choosing complementary features As Collins et al. state in their conclusion: *"Weight images produced by two high-ranking features are often highly correlated and, therefore, not much new information is introduced by adding the second feature."* This is not a major concern in the Collins tracking framework, since the features are used independently for tracking, and their results combined using a naive median. If the features were to be used together, such as in a spatiogram, it then becomes important to choose complementary features, since the addition of redundant features would not increase tracking performance. A new method of feature selection is required that selects features that are complementary, in order for them to be used in a spatiogram-bank tracker. One possible method is suggested here for future evaluation.

The problem of efficient feature selection is of importance in constructing classifiers in order to reduce the dimensionality of the classifier, but to retain maximum classification performance [42, 73, 102]. In tracking, the object and background are to be discriminated, and the first stage of the Collins method provides a good method for quickly evaluating how well each feature can separate the object and background, creating weight images from tuned features.

Using the weight images, one method to choose complementary features is to select

K features in order to approximate the perfect weight image using a linear combination of their feature weight images. This is akin to sparse linear regression [17], where a subset of signals are to be chosen such that a linear combination of these signals best approximates the target signal. The desired *ideal* weight image has the value 1 inside the object area and value -1 in the background area. This is designated as the *target* image. Features are chosen incrementally in order to minimise the squared error between the target image and its approximation, generated using a linear combination of the features selected so far.

The target image, T, and all weight images, W_i, are first shifted to have a zero mean value, by adding a constant. The target approximation, A, is initialised to an empty image (all zeros). To choose the next more useful feature, a multiplier, m_i, is first computed for each feature as follows:

$$m_i = \frac{\sum_{p \in P} R(p) W_i(p)}{\sum_{p \in P} W_i(p)^2} \tag{6.24}$$

where p is the pixel position, P is set of position of the entire image and R is the residual image, with $R = T - A$. This multiplier is optimal to best approximate the residual. Next, the best feature, f, is selected as the one that minimises the squared error:

$$f = \arg\min_i \sum_{p \in P} (R(p) - m_i W_i(p))^2 \tag{6.25}$$

The approximation is then updated, $A \leftarrow A + m_f W_f$, and the residual is recomputed, $R = T - A$. The process continues, incrementally selecting features, progressively improving its approximation of the ideal weight image.

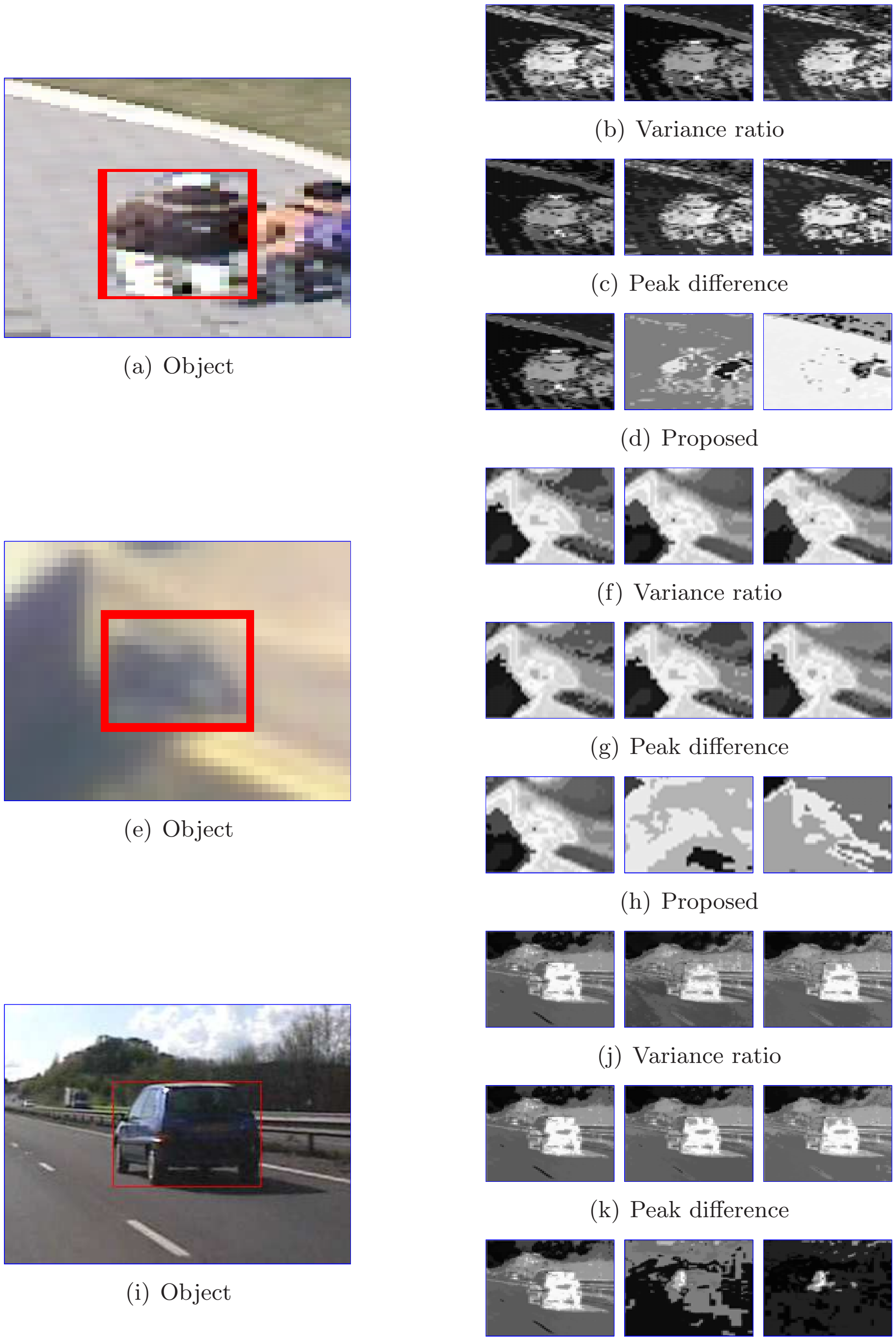
To compare this feature selection method to the Collins methods, some illustrative examples are used. Figure 6.34 compares features selected by three methods: the variance ratio, the peak-difference method and this new approach of complementary feature selection.

Figure 6.34(a) shows an example where the tracking of a person might be distracted by a nearby person walking alongside them. The features selected by the Collins methods appear useful for tracking but they do not adequately deal with the potential distractor. The complementary features seem to perform better here, down-weighting the distractor in the second and third features in figure 6.34(d).

Figure 6.34(e) shows a very difficult tracking example, where a car enters a shadowed region and strongly resembles the building and the grass in the lower part of the image. Features selected using the variance ratio and peak-difference appear highly correlated and do not provide good object/background discrimination. By combining the complementary features selected by the proposed method, the grass is down-weighted in feature 2 and the building is down-weighted in feature 3, potentially providing better tracking.

Figure 6.34(i) shows a car to be tracked. The complementary features selected appear to highlight parts of the car that the other features miss, such as the side of the car and the windows.

Collins improved upon the variance ratio by proposing the peak-difference that examined spatial correlations in pixels that could lead to distraction. Choosing complementary features as proposed examines correlations across the spatial and feature domains. However, future work will be required to investigate whether complementary feature selection helps tracking and how it should best be incorporated into a multi-feature tracker, such as the spatiogram-bank tracker.



(a) Object

(b) Variance ratio

(c) Peak difference

(d) Proposed

(e) Object

(f) Variance ratio

(g) Peak difference

(h) Proposed

(i) Object

(j) Variance ratio

(k) Peak difference

(l) Proposed

Figure 6.34: Examples comparing three feature selection approaches: variance ratio, peak difference and the proposed complementary feature selection approach.

Chapter 7

Conclusions

7.1 Thesis overview

The work in this thesis examined the use of multiple sources of information in two main areas of computer vision: dynamic parameter selection for object and event detection, and the adaptive tracking of objects. The primary focus was the fusion of visual information with thermal infrared imagery, but the general techniques developed here have been shown to be of general use for multi-source fusion, such as in using visual and audio detection signals for event detection.

Chapters 3 and 4 concern the use of mutual information thresholding on synthetic and real data, respectively. In chapter 3, mutual information thresholding is introduced as a method to automatically select thresholds for two data sources by maximising the mutual information between the thresholded signals. Synthetic tests show that this paradigm outperforms traditional adaptive thresholding techniques. In the closing remarks of chapter 3 and in chapter 4, this idea is extended to encompass other measures of signal agreement, such as Kendall's τ.

Chapter 4 covers six individual applications of maximal agreement thresholding on real data. The first three applications use pairs of sources that are not strongly independent, such as colour-band background differences. These applications examine the use of the proposed technique in foreground pixel detection, person detection and shadow pixel detection. The obtained results show that maximal agreement thresholding often provides reasonable results, despite the weak independence of the data sources. The final three applications examine the method's use with strongly independent data sources, such as visual and thermal data, or audio and visual data. Results obtained on synthetic data in chapter 3 are shown to apply to real data, such as in choosing the optimal smoothing parameter to exploit spatial information and in automatically detecting when no common information is present. The three examined applications are foreground pixel detection in thermo-visual data, event detection using audio-visual data and adaptive skin pixel modelling using colour-based and infrared-based detectors.

In chapters 5 and 6 the recently introduced spatiogram tracker is proposed as the basis for a multi-feature tracking framework, termed a *Bank of Spatiograms*. The framework splits the tracking features over multiple spatiograms, which benefits tracking performance in a number of ways: It bypasses the exponential growth in memory and computational processing required when additional features are added to the object model distribution; It also overcomes the *curse of dimensionality* that describes the difficulty in estimating high-dimensional distributions. A mean-shift procedure is derived for the new tracker, allowing rapid object localisation using a low number of position-hypothesis evaluations. In the experiments of chapter 5, the proposed bank-of-spatiograms tracker is shown to outperform traditional tracking approaches, such as histogram-based or template-based tracking.

In chapter 6, some the limitations of the original spatiogram similarity measure are described and a new measure of similarity is derived from the Bhattacharyya coefficient. This new measure is shown to address these limitations and provide accurate object localisation and smooth similarity surfaces. Next, an investigation into model updating strategies is conducted. Performing tracking on 15 thermo-visual sequences, the performance of various model-updating approaches is measured. As reported in other works, retaining the original object model is shown to be useful in avoiding model drift. Also in chapter 6, an additional benefit of the bank of spatiograms tracker is demonstrated, namely its flexibility to allow adaptive weighting of different features. A feature weighting architecture and an adaptive weight selection scheme is proposed for the spatiogram-bank tracker. By adaptively weighting features, high-quality tracking is obtained, outperforming traditional tracking approaches, as well as the state-of-the-art Collins tracker.

7.2 Summary of contributions

The thesis makes contributions in two main areas: adaptive thresholding and object tracking.

In the first half of this thesis, a new paradigm in adaptive thresholding, termed *mutual information thresholding* is developed. Unlike traditional thresholding methods, this approach examines the relationship between two input signals and chooses its parameters to maximise agreement between the resulting binary signals. The behaviour of the proposed method is explored using synthetically generated data, where it outperforms four of the most common dynamic thresholding methods. In the following chapter, numerous examples of its use are demonstrated using real multimodal data. Its performance is compared to other top-performing thresholding algorithms. The results show that by selecting parameters to maximise agreement between independent sources, these adaptive methods outperform pre-learned static models.

In the area of object tracking, a new tracker, termed the *bank of spatiograms* tracker is proposed in this thesis. The core of the tracker is the recently proposed spatiogram, which generalises the histogram descriptor by additionally storing coarse spatial information. The bank of spatiograms framework allows tracking features to be split arbitrarily over multiple spatiograms, thereby reducing memory overhead and computational time, as well as avoiding the curse of dimensionality associated with using a small number of samples compared to the dimensions of the distribution. Its tracking performance is shown to be superior to traditional tracking approaches in multispectral data.

Noting some significant drawbacks in the original spatiogram similarity measure, a new measure of similarity is derived from the Bhattacharyya coefficient and shown to overcome the limitations of the original approach. The proposed tracker is then extended to allow dynamic weighting of the tracking features and a weight-selection mechanism is proposed that attempts to best localise the tracked object in scale space, as well as the spatial dimension. Extensive tracking experiments are conducted and they demonstrate the tracker's encouraging performance on difficult thermo-visual video data, outperforming traditional tracking approaches, as well as the state-of-the-art in adaptive tracking.

7.3 Future work

The experiments described in this thesis point to numerous avenues of potential investigation for future research. Here, some directions are suggested that may lead to profitable lines of inquiry.

In the experiments of chapter 4, both Kendall's τ and mutual information were used as agreement measures. One would imagine that if there existed a set of parameters corresponding to some real common information, then it should appear as a peak in the surfaces of both agreement measures. Investigating how the measures could be combined, perhaps by examining peak proximity, might lead to improved performance.

Chapter 3 showed a synthetic data example where the noise properties would change over time. The Online-MI thresholding method was described and shown to outperform standard MI thresholding in such fluctuating data, given the correct window size. The automatic selection of optimal window size is an area that has not yet been thoroughly examined but the quality value may be useful for this task.

The experiments in chapter 4 showed that maximising agreement led to improved results. One of the most interesting avenues of future investigation lies in examining how to extend the idea of agreement to more than two sources. If P sources are used, the resulting binary maps have a joint histogram of size 2^P. There are two main challenges to extending the work in this direction. Firstly, a P-source agreement function (based on the joint histogram, or also on the spatial information) must be composed, one that is robust to outliers so that a single noisy source does not spoil the results. And secondly, since the parameter space becomes exponentially larger as more sources are added, computationally efficient methods are needed to search this space. A dynamic-programming approach, similar to the dynamic bounding algorithm used for skin detection in chapter 4, could be a worthwhile line of inquiry, adapting two parameters at a time to ascend the agreement surface iteratively.

Using only a small number of samples to compute the object's spatiograms can lead to problems. Even given the splitting of features to reduce dimensionality, the curse of dimensionality can still strike if the object is very small. In such instances, the use of adjacent-bin similarity would be very beneficial, instead of using the Bhattacharyya coefficient, which compares each bin only to its corresponding-bin. In future, it might be possible to extend the diffusion distance [80] to compare spatiograms, as this measure has been shown to outperform Bhattacharyya and other bin-wise measures. It is also computationally efficient, which is a key requirement for real-time tracking.

In the tracking experiments, each object was represented by a rectangular bounding box in each frame. This was appropriate for most of the objects in this work, but the addition of an orientation parameter would benefit the tracking of other types of object, such as human arms. This additional parameter would increase the dimensions of the search space, and smarter ways of searching this space would be required to maintain efficiency. The particle filter, alluded to in the discussion of the previous chapter is one such technique that seems useful in this regard. Additionally, any particles (position hypotheses) that lie on background areas would not be wasted computation, as they could provide valuable information on potential distractors for the adaptive tracker [152].

Some initial work in complementary feature selection for tracking was shown in the closing discussion of the previous chapter. The correlated nature of the features selected by the Collins method was demonstrated, but future work is required to investigate the performance gain in tracking by using these complementary features. Additionally, it will be interesting to know if the features should be updated in each frame or if there are smarter ways to perform the updating, such as when a significant distractor is detected.

7.4 Closing remarks

The work in this thesis examined the use of frameworks for fusing data from multiple sources of information. While thermal infrared and visual spectrum video data were used extensively in the thesis experiments, trends in current research and technology point to many new sources of information that will soon become widely available. Examples of such information sources are Global positioning (GPS) data, data from wearable sensors, such as skin conductivity sensors and heartrate monitors, Radio Frequency Identification tags (RFID), as well as video data from multiple spectral bands. The fusion of these multiple sources of information to tackle previously difficult problems will lead to many exciting developments in the years ahead.

Appendix A

Appendix A

A.1 Proof of quality score bounds

Let us begin by first simplifying the variable names as follows:

$$a = \frac{1}{N} C_{1,1} \tag{A.1}$$

$$b = \frac{1}{N} C_{1,0} \tag{A.2}$$

$$c = \frac{1}{N} C_{0,1} \tag{A.3}$$

$$d = \frac{1}{N} C_{0,0} \tag{A.4}$$

where we have $a, b, c, d \geq 0$ and $a+b+c+d = 1$. Recall that N is the number of samples (pixels) and the values of $C_{u,v}$ are the number of pixels that have binary classification u in data source 1 and binary classification v in data source 2. We have assumed that

$$\begin{array}{rccc} & a & \geq & (a+b)(a+c) \\ \Rightarrow & a & \geq & a^2 + a(b+c) + bc \\ \Rightarrow & a^2 + a(b+c-1) + bc & \leq & 0 \\ \Rightarrow & K & \geq & 0 \end{array} \tag{A.5}$$

where $K = -a^2 - a(b+c-1) - bc$ is a non-negative number. We can express mutual information as a sum of four values:

$$\begin{aligned} I(X;Y) = & \quad a\log\tfrac{a}{(a+b)(a+c)} & + & \quad b\log\tfrac{b}{(b+a)(b+d)} & + & \quad c\log\tfrac{c}{(c+a)(c+d)} & + & \quad d\log\tfrac{d}{(d+b)(d+c)} \\ I(X;Y) = & \quad T_a & + & \quad T_b & + & \quad T_c & + & \quad T_d \end{aligned} \tag{A.6}$$

with our normalising factor given by:

$$\begin{aligned} I_{max} &= -a\log a &+& -(1-a)\log(1-a) \\ &= T_1 &+& T_2 \end{aligned} \tag{A.7}$$

We now show that, using the assumption in equation(A.5), that the following inequalities are true:

$$T_b \leq 0 \tag{A.8}$$

$$T_c \leq 0 \tag{A.9}$$

$$T_1 \geq T_a \tag{A.10}$$

$$T_2 \geq T_d \tag{A.11}$$

and by proving these four inequalities, and noting equations (A.7) and (A.6), we prove that

$$I(X;Y) \leq I_{max} \tag{A.12}$$

which shows that the quality score, Q, always lies between 0 and 1:

$$0 \leq Q = \frac{I(X;Y)}{I_{max}} \leq 1 \tag{A.13}$$

A.1.1 Proof of inequality (A.8)

$$\begin{aligned} T_b &= b\log\frac{b}{(b+a)(b+d)} && \text{(A.14)} \\ &= b\log\frac{b}{(b+a)(1-a-c)} && \text{(A.15)} \\ &= b\log\frac{b}{b-ab-bc+a-a^2-ac} && \text{(A.16)} \\ &= b\log\frac{b}{b+K} && \text{(A.17)} \end{aligned}$$

Since $K \geq 0$, we have

$$\begin{array}{rcl} \frac{b}{b+K} & \leq & 1 \\ \Rightarrow \quad b \log \frac{b}{b+K} & \leq & 0 \\ \Rightarrow \quad T_b & \leq & 0 \end{array} \tag{A.18}$$

A.1.2 Proof of inequality (A.9)

This proof is essentially the same as the previous one, where we show that T_c can be written as:

$$T_c = c \log \frac{c}{c+K} \tag{A.19}$$

and therefore, since $K > 0$, this leaves

$$\begin{array}{rcl} \frac{c}{c+K} & \leq & 1 \\ \Rightarrow \quad c \log \frac{c}{c+K} & \leq & 0 \\ \Rightarrow \quad T_c & \leq & 0 \end{array} \tag{A.20}$$

A.1.3 Proof of inequality (A.10)

$$\begin{array}{cc} & -a\log(a) - a\log\frac{a}{(a+b)(a+c)} \\ = & -a(\log\frac{a^2}{(a+b)(a+c)}) \\ = & -a(\log\frac{a^2}{a^2+ab+ac+bc}) \end{array} \tag{A.21}$$

Since the denominator in the log expression is greater than the numerator, the log value will be negative, giving

$$\begin{array}{rccc} & -a(\log\frac{a^2}{a^2+ab+ac+bc}) & \geq & 0 \\ \Rightarrow & -a\log(a) - a\log\frac{a}{(a+b)(a+c)} & \geq & 0 \\ \Rightarrow & -a\log(a) & \geq & a\log\frac{a}{(a+b)(a+c)} \\ \Rightarrow & T_1 & \geq & T_a \end{array} \tag{A.22}$$

A.1.4 Proof of inequality (A.11)

First, we rewrite T_d as

$$T_d = d \log \frac{d}{(d+b)(d+c)} = d \log \frac{d}{d-K} \tag{A.23}$$

Now, in order to show that $T_2 \geq T_d$, we have to prove that

$$-(1-a)\log(1-a) \geq d \log \frac{d}{d-K} \tag{A.24}$$

$$\begin{aligned} &= \tfrac{1}{1-a} - \tfrac{d}{d-K} \\ &= \tfrac{ad-K}{(1-a)(d-K)} \\ &= \tfrac{bc}{(1-a)(d-K)} \geq 0 \end{aligned} \tag{A.25}$$

The last line follows from the fact that $bc \geq 0$, $(1-a) \geq 0$ and $(d-K) = (d+b)(d+c) \geq 0$. This shows that

$$\begin{array}{rccc} & \frac{1}{1-a} - \frac{d}{d-K} & \geq & 0 \\ \Rightarrow & \frac{1}{1-a} & \geq & \frac{d}{d-K} \end{array} \tag{A.26}$$

and since we also know that $1-a \geq d$ from $a+b+c+d=1$, therefore

$$\begin{array}{rccc} & (1-a)\log\frac{1}{1-a} & \geq & d\log\frac{d}{d-K} \\ \Rightarrow & -(1-a)\log(1-a) & \geq & d\log\frac{d}{(d+b)(d+c)} \\ \Rightarrow & T_2 & \geq & T_d \end{array} \tag{A.27}$$

That completes the proof.

A.2 Derivation of gradient of OD ratio

Before computing the object-to-distractor ratio (OD ratio), all scores are replaced by their *log* values, which makes the derivation more straight-forward. Given a set of K log-similarity surfaces, $s_k(p)$, $k \in \{1, 2, .., K\}$, with p being the candidate position, and

K representing the number of features, the fused surface is given by

$$f(p) = \sum_{i=1}^{K} w_i s_i(p) \tag{A.28}$$

where the weights sum to 1, so that $\sum_{i=1}^{K} w_i = 1$. The weights, w_k, are computed from K independent variables, $\{W_1, W_2, ..., W_K\}$:

$$w_k = \frac{W_k}{\sum_{i=1}^{K} W_i} \tag{A.29}$$

Since subtraction in log-space is the same as division, the object-to-distractor ratio (OD ratio) is computed as:

$$g = f(p_0) - f(p_1) \tag{A.30}$$

where p_0 is the object position and p_1 is the position of the strongest distractor. The goal is to change the independent W_k variables in order to maximise the OD ratio, g. We can see how the W_k variables affect the weights as follows:

$$\frac{\partial w_k}{\partial W_k} = \frac{(\sum_{i=1}^{K} W_i) - W_k}{(\sum_{i=1}^{K} W_i)^2} \tag{A.31}$$

And if $a \neq k$

$$\frac{\partial w_a}{\partial W_k} = \frac{-W_a}{(\sum_{i=1}^{K} W_i)^2} \tag{A.32}$$

Since the weights sum to one, they are not independent and influence each other. If we have a, $a \neq k$, this gives:

$$\frac{\partial w_a}{\partial w_k} = \frac{\partial w_a}{\partial W_k}\frac{\partial W_k}{\partial w_k} = \frac{-W_a}{(\sum_{i=1}^{K} W_i) - W_k} \tag{A.33}$$

Using this result, the partial derivative of the fused score with respect to the weights is given by

$$\frac{\partial f(p)}{\partial w_k} = \sum_{i=1, i\neq k}^{K} \frac{-W_i}{(\sum_{j=1}^{K} W_j) - W_k} s_i(p) + s_k(p) \tag{A.34}$$

Now the partial derivative of the fused score with respect to the independent W_k terms is given by:

$$\frac{\partial f(p)}{\partial W_k} = \frac{\partial f}{\partial w_k}\frac{\partial w_k}{\partial W_k} \tag{A.35}$$

$$= \left[\sum_{i=1,i\neq k}^{K} \frac{-W_i}{(\sum_{j=1}^{K} W_j) - W_k} s_i(p) + s_k(p)\right] \left[\frac{(\sum_{i=1}^{K} W_i) - W_k}{(\sum_{i=1}^{K} W_i)^2}\right] \quad (A.36)$$

$$= \sum_{i=1,i\neq k}^{K} \frac{-W_i}{(\sum_{j=1}^{K} W_j)^2} s_i(p) + s_k(p) \left(\frac{\sum_{j=1}^{K} W_j - W_k}{(\sum_{j=1}^{K} W_j)^2}\right) \quad (A.37)$$

If we now define

$$S_k = s_k(p_0) - s_k(p_1) \quad (A.38)$$

then we can write $\partial g / \partial W_k$ as

$$\frac{\partial g}{\partial W_k} = \sum_{i=1,i\neq k}^{K} \frac{-W_i}{(\sum_{j=1}^{K} W_j)^2} S_i + \left(\frac{\sum_{j=1}^{K} W_j - W_k}{(\sum_{j=1}^{K} W_j)^2}\right) S_k \quad (A.39)$$

This represents the effect each of the independent W_k variables have on the OD ratio, g. The gradient vector is then simply

$$V = \left[\frac{\partial g}{\partial W_1}, \frac{\partial g}{\partial W_2}, ..., \frac{\partial g}{\partial W_K}\right] \quad (A.40)$$

This is used in a gradient ascent procedure, recomputing the distractor position, p_1, at each step.

References

[1] Video surveillance & sensor networks (vssn) dataset, 2006 (http://mmc36.informatik.uni-augsburg.de/VSSN06_OSAC/). 84, 86

[2] W. Abd-Almageed, C. E. Smith, and S. Ramadan. Kernel snakes: non-parametric active contour models. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 240–244, October 2003. 34

[3] T. Adamek, N. O'Connor, and A. F. Smeaton. Word matching using single closed contours for indexing handwritten historical documents. *International Journal on Document Analysis and Recognition, Speciall Issue on Analysis of Historical Documents*, pages 1–13, 2006. ISSN 1433-2833. 31

[4] A.K. Jain A.H.S. Solberg, T. Taxt. A markov random field model for classification of multisource satellite imagery. *IEEE Transactions on Geo-Science and Remote Sensing*, 37(3):100–113, 1996. 191

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks Journal*, 38(4):393–422, Mar 2002. 73

[6] F. Albregtsen. Non-parametric histogram thresholding methodserror versus relative object area. In *Proc. Eighth Scandinavian Conf. Image Analysis*, 1993. 29, 30, 65

[7] H. Andrews. *Introduction to Mathematical Techniques in Pattern Recognition*. Wiley-Interscience, 1972. 152

[8] S. Avidan. Ensemble tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005. 39, 40, 41, 149, 152, 187, 195

[9] R. E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961. 39, 149

[10] M. Bertozzi, A. Broggi, T. Graf, P. Grisleri, and M. Meinecke. Pedestrian detection in infrared images. In *Procs. IEEE Intelligent Vehicles Symposium*, pages 662–667, June 2003. URL `./bib/iv2003-ir-vw---pedestrian-detection-in-car.pdf`. 11

[11] S. T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1158–1163, June 2005. 33, 146, 147, 148, 150, 156, 161, 215

[12] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory*, pages 92–100, 1998. 123

[13] P. Browne, A. F. Smeaton, N. Murphy, N. O'Connor, S. Marlow, and C. Berrut. Evaluating and combining digital video shot boundary detection algorithms. In *Proceedings of Irish Machine Vision and Image Processing Conference*, 2000. 142

[14] Bullard. A brief military history of thermal imaging. http://www.bullard.com/Thermalimager/Media_Info/military_history.shtml, 2004. 6, 7

[15] S. G. Burnay, T. L. Williams, and C. H. Jones. *Applications of thermal imaging*. Adam Hilger, 1988. 9, 10

[16] S. Challa and D. Koks. Bayesian and dempster-shafer fusion. *Sadhana - Academy Proceedings in Engineering Sciences*, 29(2):145–174, April 2004. 20

[17] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999. 224

[18] X. Chen, P. J. Flynn, and K. W. Bowyer. Visible-light and infrared face recognition. In *Workshop on Multimodal User Authentication, Santa Barbara, CA, USA.*, Dec 2003. 17

[19] R. T. Collins. Mean-shift blob tracking through scale space. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 234–240, 2003. 33, 35, 146, 211

[20] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), Oct 2005. 38, 40, 41, 152, 156, 169, 179, 180, 186, 187, 191

[21] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (5):603–619, May 2002. 156

[22] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 142–151, 2000. 33, 146, 147, 156, 161, 165, 170, 203, 210

[23] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003. 33, 35, 147

[24] T. Cootes and C. Taylor. Active shape models- - smart snakes. In *British Machine Vision Conference*. Springer Verlag, 1992. 34

[25] R. Cucchiara. Multimedia surveillance systems. In *VSSN 05: Proceedings of the third ACM international workshop on Video surveillance & sensor networks, New York, NY, USA*, pages 3–10, 2005. 15

[26] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175–185, June 2000. 27

[27] J. Davis and V. Sharma. Fusion-based background-subtraction using contour saliency. In *Proc. IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum*, June 2005. 16, 26, 38, 164, 210

[28] J. W. Davis and V. Sharma. Robust background-subtraction for person detection in thermal imagery. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 8*, Jun 2004. 11, 13

[29] J.W. Davis and M.A. Keck. A two-stage template approach to person detection in thermal imagery. In *Workshop on Applications of Computer Vision*, volume 1, pages 364–369, 2005. 11, 83, 97

[30] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision*, 2000. 23, 27

[31] A. Elgammal, R. Duraiswami, and L. S. Davis. Probabilistic tracking in joint feature-spatial spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2003. 146

[32] C. K. Eveland, D. A. Socolinsky, and L. B. Wolff. Tracking human faces in infrared video. *Image and Vision Computing*, 21(7):579–590, July 2003. 11

[33] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy – automatic naming of characters in tv video. In *Proceedings of the British Machine Vision Conference*, 2006. 142

[34] Y. Fang, K. Yamada, Y. Ninomiya, B.K.P. Horn, and I. Masaki. A shape-independent method for pedestrian detection with far-infrared images. *IEEE Transactions On Vehicular Technology*, 53(5), Sept 2004. 11

[35] D. Farin, P. H. N. de With, and W. Effelsberg. Robust background estimation for complex video sequences. In *International Conference on Image Processing (ICIP)*, Sept 2003. 26

[36] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1029, 1993. 222

[37] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew. On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006. 99

[38] G. Fumera and F. Roli. A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):942–956, June 2005. 38

[39] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 1999. 25

[40] E. Goubet, J. Katz, and F. Porikli. Pedestrian tracking using thermal infrared imaging. In *SPIE Conference Infrared Technology and Applications XXXII*, volume 6206, pages 797–808, June 2006. 16

[41] A. H. Gunatilaka and B. A. Baertlein. Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):577–589, June 2001. 18

[42] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003. 41, 223

[43] C. M. Hanson, H. Beratan, R. A. Owen, M. Corbin, and S. McKenney. Uncooled thermal imaging at texas instruments. In Wagih H. Makky, editor, *Proc. SPIE Infrared Detectors: State of the Art*, volume 1735, pages 17–26, Dec 1992. 7

[44] R. W. Hardin. Uncooled ir focal plane arrays going global. *SPIE OE Reports*, (195), Mar 2000. 8, 12

[45] I. Haritaoglu, D. Harwood, and L. S. Davis. W4s: A real time system for detecting and tracking people in 2.5 d. In *Fifth European Conference on Computer Vision*, pages 877–892, June 1998. 27

[46] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003. 14

[47] M. Harville. Stereo person tracking with adaptive plan-view templates of height and occupancy statistics. *J. Image and Vision Comp*, 22(2):127–142, 2004. 27

[48] M. Harville, G. Gordon, and J. Woodfill. Adaptive video background modeling using color and depth. In *International Conference on Image Processing (ICIP)*, Oct 2001. 25

[49] J. Heo, S. Kong, B. Abidi, and M. Abidi. Fusion of visual and thermal signatures with eyeglass removal for robust face recognition. In *IEEE Workshop on Object Tracking and Classification Beyond the Visible Spectrum*, pages 94–99, July 2004. 16

[50] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34(3):334–350, August 2004. 15

[51] B. Huet and E. R. Hancock. Cartographic indexing into a database of remotely sensed images. In *3rd IEEE Workshop on Applications of Computer Vision (WACV '96)*, 1996. ISBN 0-8186-7620-5. 34

[52] C. Ianasi, V. Gui, C. I. Toma, and D. Pescaru. A fast algorithm for background tracking in video surveillance, using nonparametric kernel density estimation. *FACTA UNIVERSITATIS (Nis), Series: Electronics and Energetics*, 18(1):127–144, April 2005. 25

[53] I.Haritaoglu, D.Harwood, and L.Davis. Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (22):781–796, August 2000. 23

[54] M. Irani and P. Anandan. Robust multi-sensor image alignment. In *International Conference on Computer Vision*, pages 959–966, 1998. URL `http://citeseer.ist.psu.edu/irani98robust.html`. 14

[55] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. 34, 35

[56] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *Proc. 8th Int. Conf. Computer Vision*, 2001. 24

[57] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *Workshop on Motion and Video Computing*, pages 22–27, Dec 2002. 25

[58] C. Jaynes, A. Kale, N. Sanders, and E. Grossmann. The terrascope dataset: A scripted multi-camera indoor video surveillance dataset with ground-truth. In *Procs of the IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005. 84, 86, 106

[59] A. Jepson, D. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10), 2003. 34

[60] B. F. Jones. A reappraisal of the use of infrared thermal image analysis in medicine. *IEEE Transactions on Medical Imaging*, 17(6):1019–1027, Dec 1998. 11

[61] B. F. Jones and P. Plassmann. Digital infrared thermal imaging of human skin. *IEEE engineering in medicine and biology magazine*, 21(6):41–48, Nov/Dec 2002. 10, 11

[62] G. D. Jones, R. E. Allsop, and J. H. Gilby. Bayesian analysis for fusion of data from disparate imaging systems for surveillance. *Image and Vision Computing*, 21(10):843–849, Sept 2003. 16, 191

[63] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *2nd European Workshop on Advanced Video-based Surveillance Systems, Kingston upon Thames*, 2001.

URL http://citeseer.ist.psu.edu/kaewtrakulpong01improved.html. 23, 111, 112, 113

[64] H. Kaplan. *Practical Applications of Infrared Thermal Sensing and Imaging Equipment*. SPIE-International Society for Optical Engine, 1993. Atmospheric and Glass transmission of infrared on pg 23. 7, 10

[65] J. Kapur, P. Sahoo, and A. Wong. A new method for graylevel picture thresholding using the entropy of the histogram. *Computer Graphics and Image Processing*, 29(3):273–285, 1985. 29, 43, 45, 65

[66] R. Kasturi. Performance evaluation protocol for face, person, and vehicle detection & tracking analysis and content extraction (vace-ii). Technical report, Computer Science & Engineering University of South Florida, Tampa, 2006. 210

[67] M. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938. 79, 104

[68] J. Kerr. Review of the effectiveness of infrared thermal imaging (thermography) for population screening and diagnostic testing of breast cancer. *NZHTA Tech Brief Series*, 3(3), July 2004. 10

[69] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recognition*, 19(1):41–47, 1986. 30, 66

[70] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *3rd IEEE Workshop on Visual Surveillance*, 2000. 27

[71] H. Kruppa and B. Schiele. Hierarchical combination of object models using mutual information. In *BMVC*, 2001. 44, 82

[72] H. Kruppa, M. A. Bauer, and B. Schiele. Skin patch detection in real-world images. In *Proceedings of Annual Symposium for Pattern Recognition of the DAGM*, pages 109–116, 2002. 44

[73] D.-D. Le and S. Satoh. Robust object detection using fast feature selection from huge feature sets. In *International Conference on Image Processing (ICIP)*, October 2006. 41, 223

[74] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005. 20, 38

[75] I. Leichter, M. Lindenbaum, and E. Rivlin. A probabilistic framework for combining tracking algorithms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 445–451, 2004. 41, 152

[76] L. Li and M. K. H. Leung. Integrating intensity and texture differences for robust change detection. *IEEE Transactions on Image Processing*, 11(2):105–112, Feb 2002. 24

[77] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian. Foreground object detection from videos containing complex background. In *Eleventh ACM international conference on Multimedia*, pages 2–10, 2003. URL `http://portal.acm.org/citation.cfm?doid=957013.957017`. 23

[78] J. Lim and D. Kriegman. Tracking humans using prior and learned representations of shape and appearance. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, pages 869–874, May 2004. 20, 38

[79] S.-S. Lin. Review: Extending visible band computer vision techniques to infrared band images. Technical report, GRASP Laboratory, Computer and Information Science Department, University of Pennsylvania, 2001. URL `./bib/MS_CIS_01_04_techreport.pdf`. 12

[80] H. Ling and K. Okada:. Diffusion distance for histogram comparison. In *In proc. IEEE Conf. on Computer. Vision and Pattern Recognition (CVPR), New York, 2006*, 2006. 34, 223, 230

[81] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 146

[82] G. Loy, L. Fletcher, N. Apostoloff, and A. Zelinsky. An adaptive fusion architecture for target tracking. In *IEEE International Conference on Automatic Face and Gesture Recognition (FGR)*, 2002. 38, 39

[83] R. C. Luo and M. G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Trans. Systems, Man, and Cybernetics*, 19(5):901–931, Sept 1989. 19

[84] X. P. V. Maldague. *Infrared methodology and technology*, volume 7 of *Nondestructive testing monographs and tracts*. Gordon and Breach Science Publishers, 1994. 9, 10

[85] T. Matsuyama, T. Ohya, and H. Habe. Background subtraction for non-stationary scenes. In *Proc. 4th Asian Conference on Computer Vision*, pages 662–667, 2000. 24

[86] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6), June 2004. 34, 35, 38, 146, 165, 179, 180, 186

[87] A. M. McIvor. Background subtraction techniques. In *Image and Vision Computing, Hamilton, New Zealand*, Nov 2000. 28

[88] S. McKenna, S. Jabri, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3):225–231, 1999. 179

[89] K. Moon and V. Pavlovic. Estimation of human figure motion using robust tracking of articulated layers. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2005. 32

[90] D. Nair and J.K. Aggarwal. Bayesian recognition of targets by parts in second generation forward looking infrared images. *Image and Vision Computing*, 18 (10):849–864, July 2000. 10

[91] H. Nanda and L. Davis. Probabilistic template based pedestrian detection in infrared videos. In *IEEE Intelligent Vehicle Symposium, Versailles, France*, 2002. 11

[92] J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965. 104

[93] E. A. Newman and P. H. Hartline. The infrared "vision" of snakes. *Scientific American*, 246(3):116–127, Mar 1982. 10

[94] W. Niblack. *An Introduction to Digital Image Processing*. Englewood Cliffs, N.J.: Prentice Hall,, 1986. 31

[95] C. Ó Conaire, E. Cooke, N. O'Connor, N. Murphy, and A. Smeaton. Background modelling in infrared and visible spectrum video for people tracking. In *2nd*

Joint IEEE International Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum (OTCBVS), San Diego, CA, USA, 2005. 27

[96] C. Ó Conaire, E. Cooke, N. O'Connor, N. Murphy, and A. F. Smeaton. Fusion of infrared and visible spectrum video for indoor surveillance. In *International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Montreux, Switzerland,*, April 2005. 27, 79

[97] C. Ó Conaire, N. O'Connor, E. Cooke, and A. Smeaton. Detection thresholding using mutual information. In *VISAPP: International Conference on Computer Vision Theory and Applications, Setúbal, Portugal*, Feb 2006. 45

[98] C. Ó Conaire, N. E. O'Connor, E. Cooke, and A. F. Smeaton. Comparison of fusion methods for thermo-visual surveillance tracking. In *International Conference on Information Fusion*, July 2006. 34, 152, 164

[99] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Transactions on System Man Cybernetics*, 9(1):62–66, 1979. 30, 66

[100] Z. Pan, G. Healey, M. Prasad, and B. Tromberg. Face recognition in hyperspectral images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 (12):1552–1560, Dec 2003. 11

[101] N. Paragios and R. Deriche. Unifying boundary and region-based information for geodesic activetracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 1999. 34

[102] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005. 41, 44, 223

[103] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proceedings of the IEEE*, 92(3):495–513, March 2004. 38

[104] J.P.W. Pluim, J.B.A. Maintz, and M.A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, Aug 2003. 13, 44

[105] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005. 33, 146, 147

[106] F. Porikli and O. Tuzel. Multi-kernel object tracking. In *IEEE International Conference on Multimedia & Expo*, July 2005. 146, 147, 156

[107] P. W. Power and J. A. Schoonees. Understanding background mixture models for foreground segmentation. In *Proc. Image Vision Comput*, pages 267–271, Nov 2002. 112, 113

[108] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, July 2003. 25, 84, 86, 103, 107

[109] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition edition, 1992. 79

[110] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14 (3):294–307, Mar 2005. 21

[111] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, Mar 2002. 73

[112] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama. Weighted voting-based robust image thresholding. In *Proc. International Conference on Image Processing (ICIP), Atlanta, GA, USA,*, Oct 2006. 30

[113] T.W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Trans. Systems, Man, and Cybernetics*, 8(8):630–632, 1978. 30

[114] O. Rockinger and T. Fechner. Pixel-level image fusion: The case of image sequences. *Proc. SPIE*, 1998. 15

[115] P. Rosin. Thresholding for change detection. In *IEEE International Conference on Computer Vision*, pages 274–279, 1998. 29, 43, 45

[116] P. L. Rosin. Unimodal thresholding. *Pattern Recognition*, 34(11):2083–2096, 2001. 29, 43, 66, 118, 128

[117] P. L. Rosin. Edges: saliency measures and automatic thresholding. *Machine Vision and Applications*, 9(4):139–159, 1997. 142

[118] P.L. Rosin and E. Ioannidis. Evaluation of global image thresholding for change detection. *Pattern Recognition Letters*, 24(14):2345–2356, 2003. 29, 30, 65, 98

[119] J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen. Adaptive document binarization. In *Proc. of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany*, pages 147–152, Aug 1997. 31

[120] D. A. Scribner, M. R. Kruer, and J. M. Killiany. Infrared focal plane array technology. *Proceedings of the IEEE*, 79(1):66–86, 1991. 8

[121] A. Selinger and D. Socolinsky. Appearance-based facial recognition using visible and thermal imagery: A comparative study. Technical report, Equinox Corportation, Feb 2002. 17

[122] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. In *2nd IEEE Int. Workshop on PETS, Kauai, Hawaii, USA*, Dec 2001. 34, 36

[123] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–165, Jan 2004. 29, 30, 42, 65, 66

[124] V. Sharma and J. W. Davis. Feature-level fusion for object segmentation using mutual information. In *IEEE Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum*, June 2006. 16

[125] K. She, G. Bebis, H. Gu, and R. Miller. Vehicle tracking using on-line fusion of color and shape features. In *IEEE International Conference on Intelligent Transportation Systems*, October 2004. 20, 38, 39, 191

[126] Y. Sheikh and M. Shah. Bayesian object detection in dynamic scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2005. 24

[127] L. Sigal, S. Sclaroff, and V. Athitsos. Skin color-based video segmentation under time-varying illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):862–877, July 2004. 136

[128] A. F. Smeaton and M. McHugh. Event detection in an audio-based sensor network. *Multimedia Systems*, 12(3):179–194, Dec 2006. 73

[129] P. Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):447–458, 1990. 19

[130] K. Smith, D. Gatica-Perez, and J.-M. Odobez. Using particles to track varying numbers of interacting people. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2005. 38

[131] P. Smits and A. Annoni. Toward specification-driven change detection. *IEEE Trans. on Geoscience and Remote Sensing*, 38(3):1484–1488, 2000. 45

[132] P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W H Freeman & Co, 1973. 70

[133] L. Snidaro and G. L. Foresti. Real-time thresholding with euler numbers. *Pattern Recognition Letters*, 24(9–10):1533–1544, June 2003. 29

[134] Infrared Solutions. Some historical facts. http://www.infraredsolutions.com/html/technology/historicalFactsF.shtml. 7

[135] M. Spengler and B. Schiele. Towards robust multi-cue integration for visual tracking. *Machine Vision and Applications*, 14(1):50–58, 2003. 38, 39, 191

[136] A. Srivastava and X. Liu. Statistical hypothesis pruning for identifying faces from infrared images. *Image and vision computing*, 21(7):651–661, July 2003. 11

[137] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of CVPR99*, pages II:246–252, 1999. 23, 111

[138] H. Stern and B. Efros. Adaptive color space switching for face tracking in multi-colored lighting environment,. In *Proc. IEEE Int'l Conf. on Automatic Face and Gester Recognition Washington DC, USA*, pages 249–254, 2002. 40

[139] C. Su and A. Amer. A real-time adaptive thresholding for video change detection. In *Proc. IEEE Int. Conference on Image Processing (ICIP), Atlanta, GA, USA*, Oct 2006. 31

[140] Y.-L. Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 1182–1187. IEEE Computer Society, 2005. 23, 25

[141] H. Torresan, B. Turgeon, C. Ibarra, P. Hébert, and X. Maldague. Advanced surveillance system: Combining video and thermal imagery for pedestrian detection. In *Proc. of SPIE, Thermosense XXVI*, volume 5405 of *SPIE*, pages 506–515, April 2004. 16, 38

[142] K. Toyama and G. D. Hager. Tracker fusion for robustness in visual feature tracking. In *SPIE Int'l Sym. Intel. Sys. and Adv. Manufacturing*, October 1995. 41

[143] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proceedings of the Seventh IEEE International Conference on Computer IEEE Comput. Soc.*, volume 1, pages 255–261, 1999. 23, 25, 84, 86

[144] J. Triesch and C. von der Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13(9):2049–2074, 2001. 20

[145] W. Tsai. Moment-preserving thresholding: A new approach. *Comput. Vision Graphics Image Process.*,, 29(3):377–393, Mar 1985. 30, 43

[146] K. Tumer and J. Ghosh. Estimating the Bayes error rate through classifier combining. In *In Proceedings of the International Conference on Pattern Recognition, Vienna*, pages 695–699, 1996. 170

[147] C. J. Van Rijsbergen. *Information Retrieval*. Dept. of Computer Science, University of Glasgow, Butterworths, London, 2nd edition edition, 1979. 69, 133

[148] A. Vetro, T. Haga, K. Sumi, and H. Sun. Object-based coding for long-term archive of surveillance video. In *IEEE International Conference on Multimedia and Expo (ICME)*, July 2003. 23, 26

[149] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 734–741, Oct 2003. 45, 47, 48, 105

[150] P. A. Viola. *Alignment by Maximization of Mutual Information*. Phd thesis, Massachusetts Institute of Technology, Massachusetts (MA), USA, June 1995. 13, 44

[151] M. Wan and J.-Y. Herve. Adaptive, region-based, layered background model for target tracking. In *International Conference on Pattern Recognition*, volume 1, pages 803–807, 2006. 26

[152] J. Wang, X. Chen, and W. Gao. Online selecting discriminative tracking features using particle filter. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1037–1042, 2005. 221, 230

[153] P. J. Withagen, F. C. A. Groen, and K. Schutte. Emswitch: A multi-hypothesis approach to em background modelling. In *Proceedings of Acivs 2003 (Advanced Concepts for Intelligent Vision Systems)*, Sept 2003. 24

[154] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 51–56, October 1996. 23

[155] C. R. Wren and F. Porikli. Waviz: Spectral similarity for object detection. In *IEEE International Workshop on Performance Evaluation of Tracking & Surveillance*, January 2005. 23

[156] F. Xu and K. Fujimura. Pedestrian detection and tracking with night vision. In *Procs. IEEE Intelligent Vehicles Symposium*, June 2002. 11

[157] C. Yang, R. Duraiswami, and L. Davis. Efficient mean-shift tracking via a new similarity measure. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 176–183, June 2005. 35, 39, 150

[158] Y. Yang and X. Liu. A re-examination of text categorization methods. In *22nd Annual International SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, Berkley, August 1999. 69

[159] K. Yasuda, T. Naemura, and H. Harashima. Thermo-key: human region segmentation from video. *Computer Graphics and Applications, IEEE*, 24(1), Jan/Feb 2004. 13

[160] Z. Zhang and R. S. Blum. A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application. *Proceedings of the IEEE*, 87(8):1315–1326, Aug 1999. 15

[161] L. Zhao and C. Thorpe. Stereo and neural network-based pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):148–154, September 2000. 27

[162] S. Zhou, R. Chellappa, and B. Moghaddam. Appearance tracking using adaptive models in a particle filter. In *Proc. of 6th Asian Conference on Computer Vision (ACCV)*, Jan 2004. 34, 35, 37, 179, 218, 222

[163] Z. Zivkovic and B. Krose. An em-like algorithm for color-histogram-based object tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 33, 146, 156