Coherent Segmentation of Video into Syntactic Regions

Alan F. Smeaton[1] Hervé Le Borgne[2] Noel E. O'Connor[1]
Tomasz Adamek[2] Oliver Smyth[2] Seán De Burca[2]

[1]Adaptive Information Cluster and Centre for Digital Video Processing,
Dublin City University, Glasnevin, Dublin 9, Ireland.

[2]Centre for Digital Video Processing,
Dublin City University, Glasnevin, Dublin 9, Ireland.

Alan.Smeaton@DCU.ie

Abstract

In this paper we report on our work in realising an approach to video shot matching which involves automatically segmenting video into abstract intertwinded shapes in such a way that there is temporal coherency. These shapes representing approximations of objects and background regions can then be matched giving fine-grained shot-shot matching. The main contributions of the paper are firstly the extension of our segmentation algorithm for still images to spatial segmentation in video, and secondly the introduction a measurement of temporal coherency of the spatial segmentation. This latter allows us to quantitatively demonstrate the effectiveness of our approach on real video data.

1 Introduction

Content-based access to video material is of huge importance as the issues of capture, compression, storage, transmission and presentation are solved enough to allow widespread, large-scale deployment of digital video systems. These digital video systems are used in applications as diverse as CCTV and surveillance [4], to TiVo, SKY+ and home access to broadcast TV archives [9] and the content-based applications we seek to develop include searching, browsing, summarisation and automatic linking between related video clips.

There are several different ways in which video material can be analysed in order to support content operations and the most straightforward, and widespread, is to use the spoken dialogue associated with a video as a key for content-based access. This can be shown as particularly useful in access to broadcast TV news, as the TRECVid benchmarking exercise has illustrated annually since 2001 [10]. However, spoken dialogue access, whether based on closed captions or speech recognition, is limited in that its access is not based on an anlysis of the actual visual content, only on what is spoken about. Thus it is useful for topical rather than content-based retrieval.

Keyframe image matching is an approach to video retrieval in which a single image from a video, a keyframe, is used as a surrogate representation for an entire shot. Keyframes can be used as an input for image-based retrieval using low-level features such as colour, texture and edge histograms and such approaches are popular in the TRECVid community [10]. For more advanced systems, objects appearing

in keyframes can be used to support object-object matching when such objects can be segmented from their backgrounds. This is more fine-grained than keyframe matching and there are a small number of systems which can demonstrate this [8], [7].

While these approaches are useful for high-level retrieval they do have limitations if trying to do exact shot matching, such as in the detection of shots which have similar composition, colours, textures, camera and object motions. This is useful in near-duplicate shot detection or in automatically creating links between very similar shots. If a shot contains object and/or camera motion then these, and their movements, are lost as a single keyframe cannot capture these. In general there has not been much work on matching an entire shot against another entire shot using visual features, though the work on temporal correlograms [6] is close.

To do full-scale shot-shot matching it make sense to aggregate or clump the visual features appearing in a frame into regions, and keep those regions across the frames of a shot, and then to index a shot by the regions and how they move during the shot. This would be equivalent to decomposing a shot into a series of moving and intertwining coloured "blobs", a bit like a lava lamp which we use in homes for decoration, and then matching shots based on the shapes, transformations and movements of these blobs.

In this paper we report on our work in realising this approach to video shot matching which involves automatically segmenting a video sequence into abstract intertwined shapes in such a way that there is temporal coherency. The rest of paper is organised as follows. In the next section we present our approach to syntactic segmentation of single video frames based on our previous work on segmentation of images and video keyframes. In section 3 we extend this to video, which we regard as a sequence of adjacent and related images. This requires us to re-evaluate syntactic segmentation to take account of previous and succeeding frames in the segmentation, so as to avoid visual "jitter". We introduce a measure of temporal coherency to quantify this and in section 4 we report an evaluation of our video segmentaiton on several video clips, presenting examples and their measures of temporal coherency. A concluding section includes discussion of our future work.

2 Syntactic Segmentation of Still Images

We summarise here our approach to syntactic segmentation of single video frames based on our previous work on segmentation of images [1]. It consists of an extension of the Recursive Shortest Spanning Tree (RSST) algorithm [5], improved by the use of syntactic features [2] leading to a more realistic segmentation of real-world objects.

The original RSST algorithm is a relatively simple and fast region-growing method. Regions are considered as the nodes of a graph, and the branches between two nodes are the merging cost of two adjacent regions. It starts from pixel level and iteratively merges regions (two regions per iteration) connected by the least cost branches. The cost was originally computed according to the average color of the regions and their sizes. The process is designed to stop when a desired number of regions are obtained or a minimum link cost is reached.

Our modifications of the original algorithm include adaptation to various region colour representations used during region grouping and adding new stopping criteria. The main contribution of this work is the introduction of a post-processing stage to avoid over-segmentation. In this stage, regions are grouped into bigger entities according to complex homogeneity criteria. This homogeneity criteria takes into account several syntactic visual features such as region complexity, contour jaggedness, geometric homogeneity and inclusion. The segmentation process itself consists of a list of ad-hoc rules that specify in which condition regions must be merged or not. This approach was proven to be efficient in practice (see figure 1 for examples). As an input parameter we can limit the maximum number of segments/regions, and the minimum number can also be fixed.

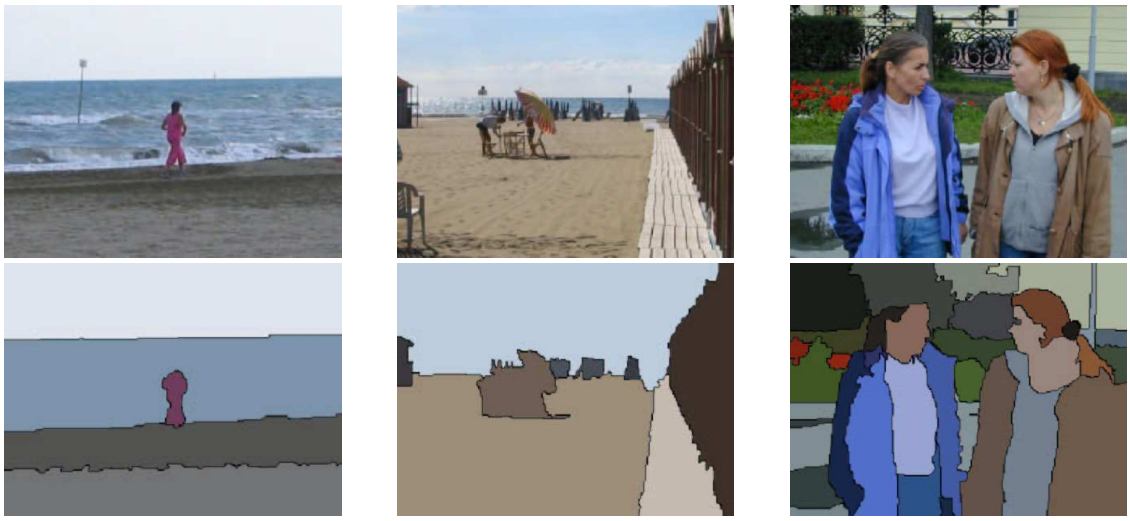

Figure 1: Example output of the arbitrary shape segmentation process. Top: original images. Bottom: result of the segmentation.

3 Syntactic Segmentation of Video

The syntactic segmentation of still images is useful but when applied to frames in a video sequence there is a lack of coherency across frames yielding a "flickering" effect as segments appear and disappear. To address this we have extended our segmentation algorithm to video as described below.

3.1 Algorithm for Video Segmentation

The general idea of the method for video region segmentation is to discover if the regions for a given frame are consistent with the regions in its neighboring (preceding and succeeding) frames. It thus require two processes: a region matching measure and consistency decision.

Region matching is a difficult and widely studied (e.g [3]) problem in general. However we perform the operation in a very particular case that allows us to make two strong assumptions which simplify it. Since we want to match regions in consecutive frames of a video shot, we assume that the regions don't change too much in terms of size, appearance or position, as long as we restrict the process to a small neighborhood for a given frame. This general assumption breaks down for the case of tracking over shot bounds or other scenes where regions will disappear such as when the camera is panning or when an object disappears from the frame and one way to address this would be to stop the tracking after some period but we ignore this for the moment. We define four boolean criteria which are true if the difference between two regions are below arbitrary thresholds. Let consider two regions R_1 and R_2, belonging to two different frames in a common neighborhood, and we want to know if they represent the same underlying object or background area which has been segmented into a region. Firstly, if their relative area sizes (number of pixels) have a difference less than 20% they are considered to have a similar area. Secondly, we check the RGB values of both regions and allow a difference of plus or minus 10 in each color component. Thirdly, we examine the actual pixels in each region and find out how many pixels are common to R_1 and R_2. If more than 85% of the pixels are common we consider the location of both region as similar. Lastly, we define the center of one region as the intersection of two lines: the first going from the leftmost point to the rightmost point of the region and the other line from the upper point to the bottom. We then compute the distance between R_1 and R_2 as the Euclidean distance between their centers and consider it is small if this value is less than 20 pixels. Finally the two regions are considered as similar when they have a similar size and color and at least one of the other features (whether a similar location or a small distance). All the thresholds were determined experimentally but the chosen values

are not very critical according to our tests.

We then must decide if a given region R must appear or not as part of the syntactic segmentation of a given frame j. Let us first examine the case of a particular region R that is not found in frame j. The decision whether or not to add R to j will be decided by a majority vote: if the region R is located in the majority of frames of the neighborhood of j then the region is added to it. In order for our hypothesis to hold, this neighborhood is no more than seven frames, centered around j. This process is repeated twice, first on the original segmented sequence and second on the sequence processed in the first pass giving us the "second iteration" in the experiments reported later. This second pass which uses the results of the first pass, does seem to increase the robustness of our method, but a third processing does not seem necessary based on our evaluations.

Finally, each frame is cleaned of any noise or once-off small errors, by calculating the best RGB value for each pixel in a frame by examining the same pixel in the frame's neighboring frame j. As a result any small discrepancies from frame to frame will be removed and the result will be a smoother output. For a given frame j and pixel coordinates (x, y) the function will look at frames $j - 3$ to $j + 3$ and for each frame note the RGB value of the pixel at position (x, y). Which ever RGB value is most common over the six frames is the new RGB value for the pixel (x, y) in j. Any once-off errors are easily handled in this manner and we found it works to very good effect and although it is an arbitrary choice, future work will probably deal with several refinements of this including developing simpler methods to determine the centre of gravity of regions. Future work could also examine using other colour spaces such as HSV or YIQ which might be better for addressing slight lighting changes or shadows emerging during a shot.

3.2 Measuring Temporal Coherency

In order to measure how much this algorithm contributes to a more temporally coherent video segment than treating frames independently, we have developed a method for actually computing a measure of temporal coherency. At present there is no widely accepted metric for measuring this and most work done in this area has focused on object-based segmentations (i.e. foreground/background) and not region-based segmentations like the algorithm presented here.

Our approach is to track a particular region through a sequence, to see how coherent it is throughout the entire sequence. In other words, for a particular region, find out what frames it does and doesn't appear in and measure the lengths of the strings or "lifelines". This gives us an idea of how each region behaves over the sequence and if we count the number of changes in the sequence i.e. when the result changes from *present* to *absent* or back again, we can use this count as a measure of temporal coherency. As a measure of temporal coherency of an entire video clip, we can apply this principle to each individual region that appears and the example in Table 1 illustrates this where a '1' indicates a region is present in a frame and a '0' indicates it is not.

Given the lifelines of each region we can calculate the temporal coherency of the sequence as the sum of the number of changes per region divided by the number of frames in the sequence of each region. normalised by the number of regions in the entire sequence:

$$T_{coh} = \frac{\sum \frac{numChanges}{numFrames}}{numRegions} \tag{1}$$

This will return a value between 0 and 1 and in the case of the data in Table 1 the value of T_{coh} will be 0.168. The closer the result is to 0 the fewer changes and less flickering detected and hence the more coherent the sequence will be. Realistically however a segmented sequence will probably never approach a value of 1 unless it is hugely incoherent and is like white noise. A value of close to 1 would only be the case where the consecutive frames in a sequence had absolutely nothing in common. On the other hand a video sequence will need to be almost absolutely still with little movement and no change or occlusion of objects in order to have a value if 0. A low, non-zero value is probably correct for most video sequences.

Table 1: Example of Calculating Temporal Coherency

	Frames									
Region	0	1	2	3	4	5	6	7	8	9
A	1	1	1	1	1	1	1	1	1	1
B	1	1	1	1	1	1	1	1	1	1
C	1	1	1	1	1	0	1	1	0	1
D	1	1	1	1	0	0	1	1	0	0
E	1	1	1	1	1	1	1	1	1	1
F	1	1	1	1	1	1	1	1	1	1
G	1	1	1	1	1	0	1	1	1	1
H	1	1	1	0	1	0	0	0	1	0
I	1	1	1	1	1	1	1	1	1	1
J	0	0	0	0	0	0	0	0	0	0
K	0	0	0	1	0	1	1	1	0	1
L	0	0	0	1	0	1	1	1	0	1
M	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0
O	0	0	0	1	0	1	1	1	0	1
P	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	1	0
S	0	0	0	0	0	0	0	0	0	1

Having a more coherent segmentation will make the video sequence more appealing to a viewer and it is an intuition that this in turn would lead to better video processing in applications like shot cut detection. In particular, an application which would use one of the frames segmented more coherently as a representative keyframe for content-based retrieval should perform better if the segmentation of the frame is in the context of the whole shot and this provides the justification for trying to remove flicker.

4 Evaluation

In our experiments we investigated different options regarding the size of the comparison set for each frame, i.e. the number of frames before and after which are used to compare against the current frame. The larger the comparison set is, the longer it takes for the program to execute and the further we look either side of frame j the more changes occur to the regions due to the natural progression of the sequence and so regions may become less relevant to the frame j anyway. This is especially the case for outdoor sequences or sequences with a lot of shot changes. We found that using three frames behind and three frames ahead was a good approach as its excution time was accectable and its results proved effective.

To evaluate our proposed methods for video segmentation with temporal coherency we used 4 well-known test sequences, each of over 300 frames. A still image from each of these is shown in Figure 2. A value for Temporal Coherence (T_{coh}) for each sequence was computed based on segmenting each frame of each of the four sequences independtly of neighbouring frames and this gives the values for the "Independent frames" in Table 2. A coherent regional segmentation for the sequences was estimated twice, firstly based on an estimation of frame coherency against frames in the original sequence only ("first iteration") and secondly based on an estimation of frame coherency against frames newly generated in lieu of the original frames ("second iteration"). This latter approach uses the coherent sequence generated in the first pass to improve again on the original. This could in theory be run again and again

Figure 2: Sample frames from the test sequences (left to right, top to bottom), Foreman, Table Tennis, Mother and Daughter, Coastguard.

Table 2: Evaluation of Region Segmentation on Standard Clips

	Temporal Coherence (T_{coh})		
Video Sequence	Independent frames	First iteration	Second iteration
Foreman	0.145	0.097	0.098
Table Tennis	0.174	0.107	0.096
Mother and Daughter	0.126	0.083	0.074
Coastguard	0.136	0.087	0.087

to try to improve the sequence however we have found that the value of this this really depends on the content of the segmented video sequence and in general two iterations of this smoothing seems to yield good results. The *foreman* sequence contains both indoor and outdoor characteristics. Later on in the sequence the shot pans down a street. The original segmented frames from this shot contain many regions unique to only 1 frame and there is a lot of flicker. The coherent sequence greatly reduces the number of rogue regions and visually the flickering effect is greatly reduced. The *table tennis* sequence is an indoor sequence with a zoom out shot and shot change. Again the flickering effect is greatly reduced in certain areas of the frame like the gray poster on the back wall for example. The *mother and daughter* sequence is a very straightforward sequence — no shot changes or camera moves. Our algorithm performs very well on this sequence as the figures show. The *coastguard* sequence is an outdoor sequence of a boat on a river. The original sequence is extremely incoherent with many regions appearing at random in each frame. Our algorithm removes a lot of the flicker but also removes a lot of detail from the sequence.

5 Conclusions and Future Work

The overall aim of the work reported here is to enable us to do fine-grained shot-shot matching which incorporates camera and object motion, composition, colour, and regions. This is important for applications like shot retrieval, automatic shot linking and near-duplicate detection and is at a level of granularity which is above retrieval based on spoken dialogue, keyframe matching based on colour histograms, or

even based on segmented objects, but has all of these, plus the temporal aspect as well. In effect we want to do shot matching in 3 dimensions. Our syntactic segmentation of moving video into regions or blobs, described in this paper is an enabler for this work and it follows that we now need to index shots by some representation of these temporal segments, and try to do shot-shot matching.

Before we can do this we have some further work which we need to do on the region-based segmentation and that includes mapping or normalising the colours used in the segmented output to some reduced palette of colours instead of allowing any colour from the original video sequence to appear. This will greatly improve the chances of shot matching if a reduced colour space is used. Even more importantly we need to re-develop our implementation to eliminate the redundant processing which makes it presently so inefficient. Our current implementation was developed with flexibility in mind, allowing us to adjust parameters and other algorithm refinements as needed but we are confident that an efficient implementation can be re-engineered.

On the algorithm side, to improve the temporal coherency of our resulting images, would require a more robust method of tracking the dynamics of each region in the frame as it moves across frames and addressing the real issues of measuring temporal coherency across a shot bound and in cases where regions enter and leave the frame during a shot. Using Markov Random Field Models (MRF) [11] could be one way of classifying the motion vectors for every region which would be based on information gathered from previous frames. There would also be scope to experiment with such a technique by processing the video in reverse order and maybe combining the segmentation results from both directions into one. MRF models can be further employed to define whether a region is dynamic or whether it is simply part of an image's background. By defining regions as part of the foreground, we could exclude irrelevant changes in partially covered background objects that lead to unwanted redefinitions of a region's boundaries.

Finally, the Temporal Coherence measure (T_{coh}) itself has some limitations in measuring coherence. To illustrate this we tested the T_{coh} of a video sequence generated from a series of totally random images and found T_{coh}= 0.1823. This was unexpected as in this case the consecutive frames will be as incoherent as it can get yet the reason for the low value was because of the number of different regions that were calculated for the sequence. Our random sequence of 20 images produced 195 different regions, compared to 18 in the mother and child sequence and in the formula for T_{coh} the sum of the number of errors per region divided by the number of frames is then normalised by the total number of regions - hence the small result for the random sequence.

The T_{coh} measure is therefore very much dependant of the length of the sequence and the number of regions generated by the algorithm. For this reason the absolute values for T_{coh} for one video sequence cannot be compared to T_{coh} values from different sequences and the appropriate application of the measure is to compare the temporal coherence of different segmentation algorithms on the same sequence. Thus the values for T_{coh} in Table 2 should be compared for segmentations within each sequence and not across different sequences.

Acknowledgments

This work is partly supported by Science Foundation Ireland under grant 03/IN.3/I361, by the European Commission under contract FP6-001765 (aceMedia) and by Enterprise Ireland and Egide under the ReSEND project (FR/2005/56).

References

[1] Adamek T., O'Connor N. and Murphy N. Region-Based Segmentation of Images Using Syntactic Visual Features. In Proceedings of WIAMIS 2005 - 6th International Workshop on Image Analysis for Multimedia Interactive Services, Montreux, Switzerland, 13-15 April 2005.

[2] Bennstrom C.F. and Casas J.R., Binary-partition-tree creation using a quasi-inclusion criterion". Proc. of the 8^{th} international conference on information visualization (ICIV), London, UK, 2004.

[3] Del Bimbo A. and Pala P.,"Image Retrieval by Elastic Matching of User Sketches", Proc. of the 8th International Conference on Image Analysis and Processing (ICIAP'95), C. Braccini and L. DeFloriani and G. Vernazza editors, Springer, pp 185-190, Berlin, 1995

[4] Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., Merkl, H. and Pankanti, S. "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking". IEEE Signal Processing Magazine, Vol. 22, No. 2, pp. 38-51, 2005.

[5] Morris O., Lee M. and Constandinides A., Graph theory for image analysis: an approach based on the shortest spanning tree. In Proceedings of the IEE. Vol 133, April 1986, pp.146-152.

[6] Rautiainen, M. and Doermann, D.,"Temporal color correlograms for video retrieval". In Proceedings of the 16th International Conference on Pattern Recognition Volume: 1, pp. 267- 270, 2002.

[7] Sav S, O'Connor N, Smeaton A.F and Murphy N, "Associating Low-level Features with Semantic Concepts using Video Objects and Relevance Feedback". 6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005), Montreux, Switzerland, 13-15 April 2005

[8] Sivic, J. and Zisserman, A. "Video Google: A Text Retrieval Approach to Object Matching in Videos". In Proceedings of the International Conference on Computer Vision, Oct 2003.

[9] Smeaton A.F., Lee H. and McDonald K., "Experiences of Creating Four Video Library Collections with the Físchlár System". Journal of Digital Libraries, Vol. 4, No. 1, pp. 42-44, 2004.

[10] Smeaton A.F., Kraaij W. and Over P. "The TREC Video Retrieval Evaluation (TRECVID): A Case Study and Status Report". In Proceedings of RIAO 2004 - Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval, Avignon, France, 26-28 April 2004.

[11] Tsaig Y., Averbuch A., "Region-based MRF Model for Unsupervised Segmentation of Moving Objects in Image Sequences", http://www.stanford.edu/ tsaig/Papers/cvpr01.pdf