

Dublin City University
School of Electronic Engineering

**Organising a photograph
collection based on human
appearance**

by

Bartłomiej Uscilowski

submitted for the qualification of MEng

supervised by Dr. Noel Murphy

September 2007

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of MEng 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: (Candidate) ID No.: 51173735 Date:

Abstract

This thesis describes a complete framework for organising digital photographs in an unsupervised manner, based on the appearance of people captured in the photographs. Organising a collection of photographs manually, especially providing the identities of people captured in photographs, is a time consuming task. Unsupervised grouping of images containing similar persons makes annotating names easier (as a group of images can be named at once) and enables quick search based on query by example.

The full process of unsupervised clustering is discussed in this thesis. Methods for locating facial components are discussed and a technique based on colour image segmentation is proposed and tested. Additionally a method based on the Principal Component Analysis template is tested, too. These provide eye locations required for acquiring a normalised facial image. This image is then pre-processed by a histogram equalisation and feathering, and the features of MPEG-7 face recognition descriptor are extracted. A distance measure proposed in the MPEG-7 standard is used as a similarity measure.

Three approaches to grouping that use only face recognition features for clustering are analysed. These are modified k-means, single-link and a method based on a nearest neighbour classifier. The nearest neighbour-based technique is chosen for further experiments with fusing information from several sources. These sources are context-based such as events (party, trip, holidays), the ownership of photographs, and content-based such as information about the colour and texture of the bodies of humans appearing in photographs. Two techniques are proposed for fusing event and ownership (user) information with the face recognition features: a Transferable Belief Model (TBM) and three level clustering. The three level clustering is carried out at “event” level, “user” level and “collection” level. The latter technique proves to be most efficient.

For combining body information with the face recognition features, three probabilistic fusion methods are tested. These are the average sum, the generalised product and the maximum rule. Combinations are tested within events and within user collections. This work concludes with a brief discussion on extraction of key images for a representation of each cluster.

Contents

1	Introduction	12
1.1	Digital photograph collections	12
1.1.1	Organising a collection	12
1.1.2	A tool for organising a collection	14
1.2	Clustering	15
1.2.1	Feature space	16
1.2.2	Similarity measure	17
1.2.3	Grouping	18
1.2.4	Data abstraction	18
1.2.5	Evaluation	19
1.3	Major contributions	19
1.4	Structure of the thesis	20
2	Facial component localisation and extraction techniques	23
2.1	Introduction	23
2.2	Geometric methods	24
2.2.1	Deformable templates	24
2.2.2	Active Shape Model	25
2.2.3	Symmetry operator	26
2.2.4	Hough transform	26
2.2.5	AI algorithms	26
2.2.6	Genetic algorithms	28
2.3	Subspace techniques	29
2.3.1	Principal Component Analysis	29
2.3.2	Independent Component Analysis	29
2.4	MPEG-7 Face Location	29
2.5	Other techniques	30
2.5.1	Gabor functions extensions	30
2.5.2	AdaBoost	30
2.5.3	Fuzzy logic	30
2.5.4	Corner detection	31
2.5.5	Colour segmentation	31

2.6	Summary	31
3	Face Component Localisation	32
3.1	Introduction	32
3.2	Facial components extraction via image segmentation	32
3.2.1	Algorithm overview	32
3.2.2	Initial segmentation — the RSST algorithm	34
3.2.3	Face localisation — skin colour segmentation	35
3.2.4	Facial components extraction with the EM algorithm	37
3.3	Experiments	45
3.3.1	Facial components extraction with user interaction	45
3.3.2	Facial components extraction with automated face localisation	49
3.3.3	Facial components extraction with the regions classification based on heuristics	51
3.4	PCA template	52
3.4.1	Assumptions	52
3.4.2	Locating eyes	53
3.4.3	Results	56
3.5	Conclusion	57
4	Face recognition	59
4.1	Introduction	59
4.2	Face recognition techniques	59
4.2.1	Subspace methods	59
4.2.2	Probabilistic approaches	63
4.2.3	Biometric-based methods	64
4.2.4	3-dimensional techniques	65
4.2.5	Other techniques	66
4.3	Recognition with MPEG-7 descriptor	66
4.3.1	MPEG-7 Face Recognition	66
4.3.2	Pre-processing	67
4.3.3	Experiments	68
4.4	Conclusion	69
5	Clustering	71
5.1	Introduction	71
5.2	Known techniques	72
5.2.1	Hierarchical algorithms	72
5.2.2	Partitional algorithms	73
5.2.3	Fuzzy algorithms	76
5.2.4	Artificial intelligence	76

5.2.5	Modern approaches	78
5.2.6	Conflicting information	79
5.3	Clustering of similar faces	80
5.3.1	Approach I: Outlier based divisive algorithm	80
5.3.2	Approach II: Single-link	85
5.3.3	Approach III: nearest neighbour	88
5.3.4	Conclusion	90
5.4	Duplicates detection	94
5.5	Conclusion	95
6	Information fusion	96
6.1	Introduction	96
6.2	Existing techniques	96
6.2.1	Simple probabilistic techniques	96
6.2.2	Transferable Belief Model	100
6.2.3	Bayesian inference	103
6.3	Available sources	105
6.3.1	Events	106
6.3.2	User collection	106
6.3.3	Body patch	106
6.4	Transferable Belief Model I	107
6.4.1	TBM for clustering	107
6.4.2	Implementation	108
6.4.3	Experiments	110
6.4.4	Conclusions on Transferable Belief Model	110
6.5	Three level clustering	111
6.5.1	Data set levels	111
6.5.2	Event level	111
6.5.3	Merging clusters	113
6.5.4	Evaluation	114
6.6	Combination of body patch with facial analysis	121
6.6.1	Body patch	121
6.6.2	Combination	124
6.6.3	Experiments	126
6.7	Conclusion	136
7	Presentation of clusters	137
7.1	Introduction	137
7.2	Presentation of clusters	137
7.3	Key-frames extraction	139
7.3.1	Non-query-based key-frames	139

7.3.2	Query-based key-frames	143
7.4	Key-frame evaluation	149
7.4.1	Objective measurements	149
7.4.2	User study	149
7.5	Conclusion	150
8	Summary	152
8.1	A brief review	152
8.2	Future work	154
8.2.1	Transferable Belief Model on similarity measure	154
8.2.2	User interaction	156
	Bibliography	159
A	Derivations required for Expectation-Maximisation algorithm	A-1
A.1	Maximum Likelihood estimation	A-1
A.2	Maximum Likelihood estimation for the mixture of Gaussian distributions	A-2
A.2.1	The likelihood maximisation of the unimodal multivariate Gaussian PDF	A-3
A.2.2	The likelihood estimation of the multimodal PDF	A-4
A.2.3	The likelihood maximisation of the multimodal PDF	A-5
B	MPEG-7 FaceRecognition Descriptor	B-1
C	Evaluation	C-1
C.1	Analysis of clusters	C-1
C.2	Dataset	C-3
C.3	Precision and recall	C-6
C.4	Definition of valid clusters	C-6
D	Detailed results — clustering	D-1
D.1	K-means approach	D-1
D.1.1	Classic similarity measure	D-1
D.1.2	Normalised similarity measure	D-10
D.2	Single-link approach	D-19

List of Figures

1.1	Sample shapes of clusters with different distance measures.	17
2.1	Eye (a) and mouth-open (b) templates proposed by Yuille <i>et al</i> [1].	24
2.2	An ASM of a human face with 58 points and norms shown. (IMM database [2]).	25
2.3	Block diagram of the genetic algorithm adopted for facial features extraction [3].	28
3.1	The RSST algorithm diagram.	34
3.2	The points of the skin coloured pixels on the UV plane and the lines bounding the skin colour area.	36
3.3	The sample facial images with the skin colour regions extracted using thresholding. The upper images are originals while the bottom ones show the extracted skin regions and the best fitted ellipses. . .	36
3.4	The histogram (a) and the Gaussian model (b) of the colour components of the region representing the eyes.	38
3.5	An example of multimodal distribution as the model of the facial object.	39
3.6	The object extraction via the EM algorithm — a system diagram. . .	41
3.7	Examples of scribbles used for facial components extraction; colours of the lines drawn by the user have been changed for better visibility.	45
3.8	Results of facial features extraction from the frame of “Foreman” sequence.	46
3.9	Results of facial features extraction from the frame of “Claire” sequence.	47
3.10	Results of eye extraction and localisation. The parameters used for extraction are presented at the bottom of each picture.	48
3.11	Results of the mouth extraction.	48
3.12	Images with lowest error rates. Numbers of RSST regions and foreground modes are given at the bottom of each image.	50
3.13	The square root error of the eye locations obtained using an approach with an automated skin colour segmentation as the function of the number of RSST regions and the number of foreground modes.	51

3.14	The square root error of the eye locations obtained using an improved approach with an automated skin colour segmentation as function of the number of RSST regions and the number of foreground modes.	52
3.15	The images with the lowest error rates. The numbers of RSST regions and the foreground modes are given at the bottom of each image.	53
3.16	The graph of the average location error when the heuristics are applied.	54
3.17	Sample human faces extracted from a photograph collection.	55
3.18	Positions of eyes in bounding box for non-rotated face.	55
3.19	Sample eye regions, (a) before and (b) after histogram equalisation.	55
3.20	Average error of location of left and right eye normalised by the distance between eyes.	57
4.1	Eigensignatures of three identities in a three-dimensional eigenspace.	63
4.2	1D HMM states for face representation [4] (a_{ij} are coefficients of the state transition matrix \mathbf{A}).	64
4.3	Examples of normalised facial image; (a) before any pre-processing, (b) after colour histogram equalisation, (c) after colour histogram equalisation and feathering with $\sigma = 0.2$	68
4.4	Recognition rates for different scenarios.	70
5.1	Sample clusters created with the nearest neighbour technique.	75
5.2	Summary of results obtained with modified k-means algorithm for scenarios presented in Table 5.1.	83
5.3	Results of clustering in scenarios outlined in Table 5.1 obtained with a use of Mahalanobis-like distance measure.	86
5.4	Summary of results obtained with modified single-link algorithm for scenarios presented in Table 5.1.	89
5.5	Results of clustering with nearest neighbour technique.	91
5.6	Precision as a function of recall for three clustering methods: k-means, simple-link and nearest neighbour based.	93
6.1	Two fusion models: (a) fusion of features, (b) fusion of decisions.	97
6.2	A sample Bayesian network for medical diagnosis.	104
6.3	Three levels at which photograph dataset can be divided.	112
6.4	Merging at user level.	112
6.5	Diagram of the system for clustering similar faces within events.	115
6.6	Diagram of the system for clustering similar faces within users collections.	116

6.7	Diagram of the system for two-level clustering of similar faces within events and users collections.	118
6.8	Precision and recall (a) and number of clusters (b) for different values of threshold for merging clusters and collection level; manual eye locations.	119
6.9	Precision and recall (a) and number of clusters (b) for different values of threshold for merging clusters and collection level; automated eye locations.	120
6.10	Diagram of the system for two-level clustering of similar faces within events and the whole collection.	121
6.11	Precision and recall and numbers of clusters produced at two level clustering on event and collection level, as a function of the merging threshold (for manually located eyes).	122
6.12	Precision and recall and numbers of clusters produced at two level clustering on event and collection level, as a function of the merging threshold (for automatically located eyes).	123
6.13	Diagram of the system for three-level clustering of similar faces: within events and at user and collection levels.	124
6.14	Results of three level clustering for different values of thresholds (manual eye locations): (a) mean precision, (b) mean recall, (c) number of created clusters, (d) number of valid clusters.	125
6.15	Results of three level clustering for different values of thresholds (automated eye locations): (a) mean precision, (b) mean recall, (c) number of created clusters, (d) number of valid clusters.	126
6.16	Diagram of the system for combining body patch with facial features and events.	128
6.17	Diagram of the system for combining body patch with facial features and user information.	129
6.18	Results for combining facial and body features within events using average distance value $p = \alpha p_f + (1 - \alpha)p_b$	130
6.19	Results for combining facial and body features within events using product of probabilities $p = p_f^\alpha \cdot p_b^{1/\alpha}$	131
6.20	Results for combining facial and body features within events using maximum $p = \max(\alpha p_f, (1 - \alpha)p_b)$	132
6.21	Results for combining using average distance value $p = \alpha p_f + (1 - \alpha)p_b$	133
6.22	Results for combining using product of probabilities $p = p_f^\alpha \cdot p_b^{1/\alpha}$	134
6.23	Results for combining using maximum $p = \max(\alpha p_f, (1 - \alpha)p_b)$	135

7.1	An example of a single personal cluster placed in the two-dimensional facial space. The images 1,4,6 and 9 were captured at event e_1 , and the images 2,3,5,7 and 8 were captured at event e_2 . The black dot denotes the centre of gravity of the cluster.	140
7.2	(a) the flow chart of the non-query-based Method 1; (b) sample choice of key images from the sample cluster using Method 1. . . .	141
7.3	(a) the flow chart of the non-query-based Method 2; (b) sample choice of key images from the sample cluster using Method 2. . . .	142
7.4	(a) the flow chart of the non-query-based Method 3; (b) sample choice of key images from the sample cluster using Method 3. . . .	144
7.5	(a) the flow chart of the non-query-based Method 4; (b) sample choice of key images from the sample cluster, large ellipses enclose subclusters.	145
7.6	(a) the flow chart of the query-based Method 1; (b) sample choice of key images for the query using Method 1.	146
7.7	(a) the flow chart of the query-based Method 2; (b) sample choice of key images for the query using Method 2.	147
7.8	(a) the flow chart of the query-based Method 3; (b) sample choice of key images for the query using Method 3.	147
7.9	(a) the flow chart of the query-based Method 4; (b) sample choice of key images for the query using Method 4.	148
8.1	BBAs as functions of a distance d	156
B.1	(a) An example of the normalised facial image and (b) an example of the description scheme containing an extracted descriptor represented in the XML format.	B-1
C.1	Sample clusters — the example for analysing results	C-2
C.2	The histogram of the occurrences of identities in the large dataset .	C-4
C.3	Histogram of occurrences of identities in the small dataset	C-5
D.1	Precision and recall for clustering with modified k-means algorithm using data set of 1127 faces with outliers; (a) precision and recall, (b) number of clusters.	D-2
D.2	Precision and recall for clustering with modified k-means algorithm using small data set of 68 faces with outliers; (a) precision and recall, (b) number of clusters.	D-3
D.3	Precision and recall for clustering with modified k-means algorithm using data set of 478 faces without outliers; (a) precision and recall, (b) number of clusters.	D-4

D.4	Precision and recall for clustering with modified k-means algorithm using small data set of 55 faces without outliers; (a) precision and recall, (b) number of clusters.	D-5
D.5	Precision and recall for clustering with modified k-means algorithm and automatically located eyes using data set of 1127 faces with outliers; (a) precision and recall, (b) number of clusters.	D-6
D.6	Precision and recall for clustering with modified k-means algorithm and automatically located eyes using small data set of 68 faces with outliers; (a) precision and recall, (b) number of clusters.	D-7
D.7	Precision and recall for clustering with modified k-means algorithm and automatically located eyes using data set of 478 faces without outliers; (a) precision and recall, (b) number of clusters.	D-8
D.8	Precision and recall for clustering with modified k-means algorithm and automatically located eyes using small data set of 55 faces without outliers; (a) precision and recall, (b) number of clusters.	D-9
D.9	Clustering with modified k-means algorithm and normalised distance using large data set with outliers and manual eye locations; (a) precision and recall, (b) number of clusters.	D-11
D.10	Clustering with modified k-means algorithm and normalised distance using small data set with outliers and manual eye locations; (a) precision and recall, (b) number of clusters.	D-12
D.11	Clustering with modified k-means algorithm and normalised distance using large data set without outliers and manual eye locations; (a) precision and recall, (b) number of clusters.	D-13
D.12	Clustering with modified k-means algorithm and normalised distance using small data set without outliers and manual eye locations; (a) precision and recall, (b) number of clusters.	D-14
D.13	Clustering with modified k-means algorithm and normalised distance using large data set with outliers and automated eye locations; (a) precision and recall, (b) number of clusters.	D-15
D.14	Clustering with modified k-means algorithm and normalised distance using small data set with outliers and automated eye locations; (a) precision and recall, (b) number of clusters.	D-16
D.15	Clustering with modified k-means algorithm and normalised distance using large data set without outliers and automated eye locations; (a) precision and recall, (b) number of clusters.	D-17
D.16	Clustering with modified k-means algorithm and normalised distance using small data set without outliers and automated eye locations; (a) precision and recall, (b) number of clusters.	D-18

D.17 Clustering with modified single-link algorithm using large data set containing outliers, manually located eyes; (a) precision and recall, (b) number of clusters.	D-20
D.18 Clustering with modified single-link algorithm using small data set containing outliers, manually located eyes; (a) precision and recall, (b) number of clusters.	D-21
D.19 Clustering with modified single-link algorithm using large data set without outliers with manually located eyes; (a) precision and recall, (b) number of clusters.	D-22
D.20 Clustering with modified single-link algorithm using small data set without outliers with manually located eyes; (a) precision and recall, (b) number of clusters.	D-23
D.21 Clustering with modified single-link algorithm using large data set containing outliers, automatically located eyes; (a) precision and recall, (b) number of clusters.	D-24
D.22 Clustering with modified single-link algorithm using small data set containing outliers, automatically located eyes; (a) precision and recall, (b) number of clusters.	D-25
D.23 Clustering with modified single-link algorithm using large data set without outliers, automatically located eyes; (a) precision and recall, (b) number of clusters.	D-26
D.24 Clustering with modified single-link algorithm using small data set without outliers, automatically located eyes; (a) precision and recall, (b) number of clusters.	D-27

List of Tables

3.1	The percentage of the correct mode classification when searching the for eyes region.	52
3.2	Error of eye localisation.	57
4.1	Scenarios for experiments with recognition.	69
5.1	Scenarios of experiments with clustering of similar faces.	84
5.2	Precision and recall of clustering using nearest neighbour based algorithm.	92
6.1	Sample table of masses associated with evidences and clusters; assuming three clusters in a set.	108
6.2	Example of generating subsets of a power set 2^3	110
6.3	Results of clustering at single levels.	117
8.1	Thresholds for MPEG-7 FR features (1127 faces, including Unknowns).	155
8.2	Components of sums for Dempster's rule of combination.	156
C.1	Annotated features	C-3
C.2	Statistics on the large dataset	C-4
C.3	Statistics on a small private dataset	C-5

Chapter 1

Introduction

1.1 Digital photograph collections

1.1.1 Organising a collection

With the rapid increase in the use of digital cameras, the storage and management of digital images is becoming an increasing problem. Hard drive capacities now available enable the storage of huge digital photograph collections. It is important to digital camera users that such collections are organised in the most efficient and automated way. For example, browsing through the unsorted collection of thousands image files named e.g. DSC00034.JPG when searching for one particular photograph of one's child at last year's holiday can be very time consuming and stressful if done in front of waiting family members.

A distinction must be noted between personal photograph collections and business collections. This is especially true when one considers the different approaches that a home user and a business user take when searching through the photographs. It is cost effective for the business user to annotate the collection in a manual or semi-automated way in order to obtain accurate representations and classes. Such organisation results in very effective and accurate search system. In contrast to this, the home user is not keen on manually annotating her/his photographs. As the home collections are usually much smaller than business collections (e.g. Irish Times collection of photographs vs. MediAssist collection [5, 6]) it is sometimes easier and more straight forward for the home user to browse the collection instead of searching by a textual query. Therefore, automated organisation of photographs allowing the user to browse the specified subset of photos seems to better match the home user's needs. This thesis concentrates on a system for organising a personal photograph collection, not a corporate one.

Some metadata associated with photographs can be used for organising photograph collections. The metadata can be categorised into two groups:

- content information: this is information on what is in an image; it can consist

of low-level features such as colour or texture, or high-level features such as identities of people captured, buildings, etc.

- context information: this is all information associated with the environment of a photograph such as creation time, location, event at which the photograph was captured, ownership, etc.

The metadata can be added through a manual annotation process or extracted in an automated way, such as colour or texture descriptors. The creation time can be also automatically obtained from an EXIF (Exchangeable Image File Format) header and location may be acquired from a GPS (Global Positioning System) device.

Research work on organising collections of digital images has been undertaken for a number of years. The early systems such as Fotofile [7] were followed by other research systems such as AutoAlbum [8], MiAlbum [9], PhotoTOC [10] or PhotoFinder [11] and techniques as presented in [12, 13, 14] and [15].

The early approaches to automated organisation of photograph collections employed some low-level content features of images such as colour, colour histogram and texture for example-based browsing and retrieval [7] or semi-automated annotating [9]. These systems require some amount of user feedback for the confirmation of validity of indexing. The content features are used in AutoAlbum [8] for automatically organising photographs into albums, allowing the user to browse through images contained in created albums. Some context information associated with photographs is also utilised in this system [8] and in others. These early approaches obtain additional information from manual annotation. Later approaches such as PhotoTOC [10] extract metadata from EXIF headers or from the parameters of the image file (e.g. creation time). The grouping of images into albums is usually based on temporal clustering in connection with colour (histogram) clustering [10]. Digital cameras are equipped with more and more tools such as integrated GPS devices (e.g. Kodak Digital Science 420 camera [13]). These are exploited by researchers for enhancing event detection and organisation in photograph collections by using both the location and creation time [13, 16, 17].

Commercial products are also available. These provide some means of image manipulation for enhancing photographs. They allow users to browse through thumbnails or lists of files. However, many file managers nowadays also provide the functionality of the thumbnail view of image files. The commercial software does not or rarely does go beyond organising photographs in albums resembling directories on a filesystem.

Modern commercial tools for managing home collections of photographs are usually Internet-oriented. They provide convenient tools for creation of personalised web pages containing photograph albums, allowing easy sharing of photographs. Some systems go even further providing only a web interface, allowing

the user to upload their photographs and then to browse them over the Internet. The examples of such systems are Ryjia 1.0 [18] or the MediAssist project [19] (although MediAssist is not a commercial system).

Apart from Ryjia 1.0 [18], which employs face recognition technology for label propagation, commercial software does not fully exploit the potential of human faces in photographs. The removal of red eye effect, brightening, sharpening or smoothing of facial region (e.g. Canon PhotoPrint software) are usually the only means of facial analysis used. However, this software does not provide tools for automatically organising photographs according to humans captured in photographs.

Modern photograph management systems provide some automated ways for creating albums. However, these are limited only to grouping photographs from the same event, time or location [13, 19]. In the case of the identities of people presented in photographs all these systems require prior, at least partial, annotation of faces, providing ways of automated propagation of labels to unlabelled faces. Also, they rely heavily on face recognition techniques, which give excellent results on pictures captured in constrained environment such as laboratories, but fail in photographs taken by the user in various conditions.

A fully automated and unsupervised method for finding occurrences of the same people is proposed in [20]. This work, however, concentrates on organising the cast list in a video. Working with a video sequence is different than working with still images. In the video sequence, the occurrences of the same person can be tracked within a scene using object tracking techniques based on the colour and texture of an object. In that way an initial grouping can be carried out and images within such groups can be used for training a classifier. It has the additional advantage of the multiple views of the same face.

This thesis proposes a system for automated, unsupervised grouping of similar people in a collection of still images. The system is based on face recognition, but also employs other techniques such as matching of clothing using the technique introduced in [21] and contextual information such as event and ownership information.

1.1.2 A tool for organising a collection

A tool for organising a collection of digital photographs based on the appearance of humans in images consists of two applications:

- an indexing tool, for organising the collection;
- a browsing tool, which enables an efficient browsing the collection.

As an example of the organisation tool one can imagine the following scenario:

Barry has uploaded his recent photographs from his digital camera onto a PC. The system goes through the photographs, finds faces in some pictures (faces indicate that humans are present), extracts them and groups similar faces. The interaction between the system and Barry might look like this:

S: I've found new photos and those faces in them; these 3 faces seem to be similar, are they of the same person? [Yes/No]

B: Yes

S: What is this person's name?

B: Deanne

S: What about these 5 faces, are they of the same person? [Yes/No]

⋮

The system creates, in an unsupervised manner, the groups of similar faces. In an ideal scenario, these faces would be of the same person captured in different photographs. The groups (clusters) of similar faces do not necessarily have to have a label (the name of a person) assigned to them. Some people from a crowd might have been captured in photographs and the user does not necessarily know their identities and/or might want to regard them as outliers. It is also possible that the system recognises such outliers and deals with them, as outliers represent the noise in a data set and might affect in a negative way the effectiveness of a clustering process.

Let us consider the scenario of Barry browsing his photo collection:

Barry browses the photographs from last holidays. He spent his holidays with Deanne, Sean and Siobhan. He wants to see more photographs of Deanne, so he clicks on Deanne's face in one of already opened images. The system responds by showing all images that contain faces from the same group as the clicked face of Deanne. Or, in the case of the large number of images and the limited screen area, only key images extracted from Deanne's cluster.

It can be seen in the second scenario that browsing photographs and searching through a collection for the photographs of a particular person does not require labels (names) associated with clusters. The images of the same person can be retrieved by getting all photographs in a cluster. If there is a limited area on which the retrieved images are shown then the key images are chosen and presented.

1.2 Clustering

There are two distinct ways of grouping data:

- supervised (called supervised clustering or classification) which assigns data to groups created using labelled training data. This method is called “classification” in this thesis.
- unsupervised (called unsupervised clustering or simply clustering) which groups unlabelled data. This method is called “clustering” in this thesis.

In the case of classification the number of groups can be learnt from the training data. In contrast to that, the number of groups (clusters) in clustering must be either given beforehand or estimated during the process.

Clustering of any data consists of at least 3 elements [22]:

1. the definition of a feature space and feature extraction
2. the definition of a similarity measure
3. grouping (classification)

Two further elements can be considered:

4. the creation of cluster prototypes (data abstraction)
5. evaluation

For the purpose of managing and organising a digital photograph collection, the first four points are most important. The fifth point is vital for comparison between different clustering techniques.

1.2.1 Feature space

A feature space gives a representation of a data set, ideally allowing one to discover patterns or a structure within the data set. This pattern representation may be more convenient for processing than the raw data, such as in the case of dimensionality reduction techniques, e.g. Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) or Independent Component Analysis (ICA). Another possibility is that features are constructed using parameters of a data model, as in the case with some probabilistic representations. In the case of selecting a subset of features it is essential to choose the most discriminative features. This leads to dimensionality reduction but also improves the accuracy of the representation.

When one considers organising a photograph collection based on the people captured in the photographs, face recognition technology seems to be an obvious choice. This follows the observation that the human recognition is based mostly on recognising human faces. Today’s state-of-the-art face recognition techniques give excellent results in controlled environments such as laboratories. However, in uncontrolled conditions (e.g. in personal photograph collections) the results of recognition are far from satisfactory [23].

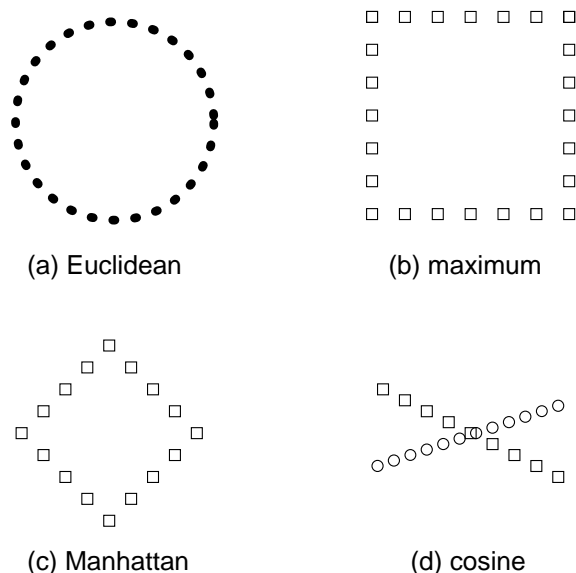


Figure 1.1: Sample shapes of clusters with different distance measures.

Therefore, in order to improve the organisation of the collection, information additional to face recognition should be used. This information may be based on the context of the photograph, such as creation time and location, the event at which the photograph was taken, or ownership. Other, content-based features may be taken into consideration such as the body patch technique introduced in [21]. These features, when combined in an efficient way, should provide highly acceptable clustering.

1.2.2 Similarity measure

A similarity measure defines a degree of similarity between two points (faces) in a data set. It is usually defined as a distance (proximity) between two points in the feature space. Therefore, it is closely related to the feature extraction or selection processes. The shapes of clusters are defined by the proximity measure and the feature space. The boundaries of the shapes of clusters for various similarity measures in two-dimensional linear feature space are presented in Figure 1.1.

The similarity measure is often chosen according to the feature space characteristics, e.g. MPEG7 Face Recognition features are known to work best with the weighted L_1 norm [24]. The similarity measure is sometimes not a metric, i.e. does not satisfy the triangle inequality [22]. It may happen when two faces (let say face A and face B) of two women are in close proximity, and one of these faces (B) is similar to her brother's face C resulting in small distance between faces B and C . Then the value of the distance between faces A and C may be higher than the sum of distances between A and B , and B and C , because A and C are the faces of people not similar at all (not related to each other and of different genders). Some distance measures might also contain conceptual or contextual information

employing so-called syntactic features [25]. The choice of similarity measure, however, is often driven by examining sample data and the patterns that the sample data forms.

1.2.3 Grouping

Grouping is an essential part of the clustering process and can be carried out in various ways [22]. The hierarchical approach produces different clusters at different hierarchy levels. The techniques based on square error minimisation (k-means), graph theory or mixture modelling usually result in hard grouping, i.e. any given data point is assigned to only one cluster. Fuzzy algorithms provide soft partitions, i.e. a membership function defines the degree of the assignment of data points to clusters. Some grouping techniques are based on artificial intelligence such as neural networks or genetic algorithms.

Several characteristics of grouping need to be analysed. The most important are the order-dependency and the computational complexity. The results of a good algorithm should not depend on the order of grouping the data i.e. grouping should be order-independent. Computational complexity defines how quickly the time of computations grows with the increase size of the data. The clustering of large data sets can be very time consuming for very complex algorithms.

1.2.4 Data abstraction

Once the grouping of data is carried out, it is vital to represent clusters in an efficient and convenient way. Therefore, the choice of cluster parameters for representation of these groups is crucial. Usually such representation emerges from the grouping algorithm, e.g. hierarchical grouping results in a hierarchical representation, which might allow a user to browse through a hierarchy.

In organising a photograph collection, the representation of clustering results can be seen from a special point of view. Clusters are to be presented to the user on a limited area available on a computer screen. Therefore, as a trade-off between the size of an image and the number of images that can be shown, the key photographs that best represent the clusters can be presented to the user.

The choice of key photographs (let us call them key-frames, similarly to key-frames in video) is not a trivial task. This choice can be based on several criteria such as time spread, event spread, scenery spread etc. However, these criteria, although they might be objective, do not necessarily reflect users' preferences. Additionally, the choice might depend on whether the presentation of clusters results from a user query or is independent of the query, or it might depend on how specific the query is. The non-query presentation of clusters can be viewed as the query-based one, with a very unspecific (general) query.

1.2.5 Evaluation

Although it is not that important for the user of a photograph collection to numerically evaluate the accuracy of clustering, it is vital for comparing various clustering approaches. The user can visibly compare the results of clustering with different algorithms and even might be able to correct the results manually. However, for objective comparison of clustering methods, the numerical assessment of clustering results is necessary.

The results of clustering can be assessed using various measures such as the spread of image features inside clusters or the ratio of the variance of the features inside clusters to the variance between clusters, etc. The most commonly used measures are precision and recall. The precision measure inspects the internal structure of clusters by examining how many images in a cluster are of the given person, i.e. it gives an insight to how many false positives are captured. The recall measure tells about how many photographs of all the photographs of the given person are captured in a single cluster, i.e. it indicates the number of missed positives.

The data set which is used for evaluation in this work consists of photographs taken by people at various occasions in different conditions. These photographs have been annotated both automatically (e.g. creation time is stored by digital cameras in the EXIF format) and manually (e.g. identities of people in photographs). Detailed description of the data set used by the author for experiments is available in Appendix C.2.

1.3 Major contributions

The major contribution described in this thesis is the application of a region-based colour image segmentation to the problem of locating facial components such as eyes, eyebrows or lips (see Section 3.2 and [26]). This is based on the observation that facial components differ in colour from other, skin coloured, areas of a human face. Therefore, these colour differences are utilised for extracting regions of the components and the locations are found as the centres of components' regions. The RSST and EM algorithms are employed for colour segmentation. They are modified to suit the purpose of the facial components extraction.

Earlier approaches to locating facial components deal mostly with intensity images, stripped from colour information. This simplifies an image representation, but removes colour information that is vital for the colour segmentation. Therefore, the colour image segmentation could not be used in earlier approaches. However, similar techniques were developed, e.g. deformable templates are fitted to an image using edge information together with valley and peak information in an intensity image [1]. Another similar work was carried out in [27], where an ob-

ervation that facial components have lower intensity values than other parts of a human face was exploited. However, they utilised only grey-level information, using colour information only for finding a full facial region in an image. The work described in Section 3.2 and in [26] can be seen as a large extension to this work, which adds colour information and uses more sophisticated algorithms (RSST and EM).

A different flavour of colour segmentation with the modified RSST algorithm was later used in [28] for locating eyes and lips. This is further used for finding the position of a human face in an image. However, this approach did not exist before the one described in [26].

The second major contribution in this thesis is the application and evaluation of methods for unsupervised clustering of human faces, that are similar in appearance (see Section 5). In the literature, one can find attempts to the unsupervised creation of cast lists in movies [29], where additional temporal information is available. In the case of still images, face recognition algorithms were applied to the problem of classification, not to unsupervised clustering. Three simple algorithms for unsupervised clustering are tested in order to find out how well they perform when dealing with the face recognition problem.

Another major contribution in this thesis is the application of the fusion of different sources of information for clustering of human faces. Context-based features — the ownership of an image and an event at which a photograph was captured, and content-based — the features of a part of human body, are fused with the face recognition features. Those additional features enhance the clustering results as the face recognition features provide limited, noisy information. Similar work is presented in [30]. They use even more sources of contextual information, but they perform only the classification and the label propagation, not unsupervised clustering.

A novel approach is proposed - a three level algorithm (see Section 5.5). This is used for combining ownership and event information with the face recognition features. It is based on the observation that clustering algorithms perform better on smaller datasets and certain people appear in photographs captured only at some events or by some users. An important conclusion drawn from the experiments is that the appearance of human faces is event specific i.e. changes between events are large enough to prevent merging the clusters that represent the same person.

1.4 Structure of the thesis

The structure of this thesis follows the steps of an unsupervised clustering process. Firstly, in Chapters 2 and 3 methods for locating eyes in images with human faces are described. These are used for extracting the face recognition features that

are described in Chapter 4. The methods for the extraction of these features are analysed. This is the first step in the unsupervised clustering process — the definition and extraction of features. The second step — the definition of similarity (or distance) function is analysed in Chapter 4 together with the definition of features for certain features work better with some similarity functions.

Chapter 2 provides the overview of state-of-the-art techniques for locating facial components such as eyes, mouth, eyebrows, etc. It concentrates on finding eye locations, because these locations are used by both component-based and holistic face recognition algorithms. A novel approach to locating eyes in a facial region, based on colour image segmentation, is introduced by the author in Chapter 3. This method exploits the observation that components such as eyes, lips or eyebrows differ significantly in colour from the skin coloured facial region. Experiments on the MPEG-7 HHI facial database and results are presented in that chapter. Another method, based on PCA is investigated for comparison. This method is quicker and more convenient when the precise position of a human head is known.

In Chapter 4, the most successful algorithms for face recognition are analysed. The analysis concentrates on investigating the suitability of the recognition algorithms to provide features for unsupervised clustering. The well known techniques such as Principal Component Analysis (PCA, eigenfaces), Linear Discriminant Analysis (LDA, Fisherfaces), probabilistic subspaces and Elastic Bunch Graph Matching are described. The MPEG-7 Face Recognition descriptor is chosen to provide face recognition features for experimentation with clustering algorithms. Two simple techniques for pre-processing the normalised facial image are investigated: the histogram equalisation and image feathering. The MPEG-7 standard does not recommend any pre-processing on normalised facial images, but it is shown that the proposed pre-processing methods, although simple, greatly enhance the recognition ratio.

In Chapter 5 methods and algorithms for the third step of clustering process — grouping — are investigated. A brief survey on techniques is presented. Then three algorithms are chosen for further analysis, these are k-means, single-link and a nearest neighbour-based one. Modifications applied to these approaches are described. The modifications were made in order to deal with outliers and estimate the number of clusters in data. Experiments conducted with these three algorithms are described and results are presented and discussed.

The next chapter (Chapter 6) presents methods for enhancing the clustering by using some sources of information additional to the face recognition features. Additional information that is used consists of information about events, the ownership of photographs and the body part of detected persons. A survey on probabilistic methods for information fusion is presented. It is followed by the description of the implementation of the Transferable Belief Model (TBM) for clustering of iden-

tities based on several sources of information. After that an alternative method is proposed for combining events and ownership information with facial features. It analyses the collection on 3 levels: event level, user (owner) level and the whole collection level. This method is based on the observations made in Chapter 5. Fusion of body information with facial features is investigated at the end of Chapter 6. Three probabilistic approaches are analysed: the average sum, the generalised product and the maximum rule. The results of experiments with combining those additional sources of information are presented and discussed.

Chapter 7 presents a discussion on choosing key images for representing clusters in a photograph collection. Two approaches are described: browsing (non-query-based) and searching (query-based). Some methods for evaluation are also presented.

A summary of this thesis is presented in Chapter 8. Two directions that future work can take are also presented in the same chapter. Another application of Transferable Belief Model (TBM) is proposed as one possible direction. This TBM is constructed solely on the similarity measure between data points unlike the TBM proposed in Chapter 6, which is constructed on similarity measure between data points and clusters. The second proposed direction would try to exploit an input given by the user. This input is collected by means of corrections made by the user to the clustering obtained in an automated way.

The thesis ends with four appendices. Appendix A contains derivations of formulae for obtaining Maximum Likelihood in the case of Gaussian distribution. These are required for deriving the Expectation-Maximisation algorithm. Appendix B presents the MPEG-7 Face Recognition descriptor in detail. Appendix C describes the methodology of experiments carried out by the author, data sets used and the definitions of measures and valid clusters. Appendix D contains detailed results of experiments described in Chapter 5.

Chapter 2

Facial component localisation and extraction techniques

2.1 Introduction

The very first step of a clustering process is defining and extracting features from the data. These features are then used for grouping data points (see Section 1.2). The extraction of features for face recognition requires the location of facial components, most importantly the eyes. Even the holistic methods such as holistic subspace algorithms need a normalised facial image, which is best defined by the positions of eyes. This is the case for example with the MPEG-7 face recognition descriptor, which defines precisely the normalised facial image (see Appendix B).

Other face recognition methods also require positions of facial components to be known. This is vital for component based algorithms, but also for biometric methods such as the Elastic Bunch Graph.

There has been a large amount of research in the area of facial feature extraction and localisation and a number of different methods have been proposed. However, those methods tend to work under specific conditions, e.g. particular pose, light conditions, constant resolution, high quality etc. The most popular approaches are based on deformable templates [31, 32, 1], symmetrical layout of facial features [31] and the Gabor Wavelet Transform (GWT) [31]. Other techniques seem to extend those methods listed above, i.e. neural networks are combined with GWT [33] or genetic algorithms are used to match deformable templates [34]. This chapter provides a description of facial feature localisation using such techniques as deformable templates, Active Shape Model, Hough transform, neural networks, genetic algorithms, Principal Component Analysis and fuzzy logic. This gives the context for the novel method proposed by the author in Chapter 3

2.2 Geometric methods

2.2.1 Deformable templates

Deformable templates were introduced by Yuille *et al* [1]. These templates can translate, rotate, change in size and change by a small controlled amount in shape to best fit to the data. Matching templates is based on minimisation of various energy functions. The parameters of templates are iteratively updated using the steepest descent method. Due to the ability of the templates to deform, the method should be flexible enough to work on images in various resolutions and size and to take into account some rotation of the head. Figure 2.1 presents the eye and mouth-open templates of archetypal features.

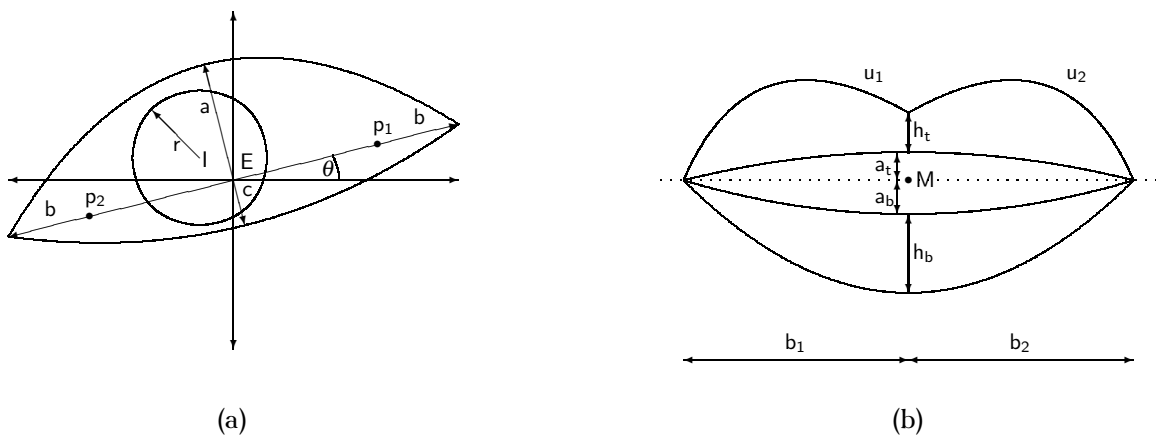


Figure 2.1: Eye (a) and mouth-open (b) templates proposed by Yuille *et al* [1]. The eye template's parameters are the centre of the iris I , its radius r , the centre of white E , parameters of the parabolas bounding white area a, b and c , rotation angle θ and centres of whites p_1 and p_2 . The mouth-open template is parameterised by the mouth centre M , rotation angle θ , parameters of the parabolas bounding lips b_1, b_2, a_b, a_t, h_b and h_t ; additionally there are two parameters u_1 and u_2 needed for description of top edge of upper lip.

The success of matching the template depends heavily on the starting points, since if the starting point is located above the eyes, the template can converge either towards an eye or towards an eyebrow, causing errors [1]. The method is computationally expensive [31, 1].

Another approach to eye templates was developed by Hallinan [31]. He models an ideal eye as two regions with uniform distribution of intensity, the regions correspond to an iris and the whites. All the differences between the ideal and the actual eye in the image are seen as the noise. Hallinan uses an α -trimmed distribution for template modelling rather than a uniform distribution. The best overall precision rate¹ of this method is at the level of 80% when using the α -

¹The precision is defined as the percentage of the correctly localised features among all the features localised in the testing dataset.

trimmed distribution [31].

Kampmann and Zhang [32] use simplified templates and cost functions for eyes extraction from video sequences. The eye template consists of the iris and two parabolas parameterised by the iris centre point, the corners and the height of the eyelid, while the radius of the iris is estimated from the distance between the corner points of the eye.

2.2.2 Active Shape Model

The active shape model (ASM) is a statistical model of the shape of any object, in particular facial components such as eyes, eyebrows, nose, mouth and the curvature of a jaw. This technique was proposed by Cootes [35] and can be viewed as a generalisation of deformable templates. It is based on PCA projection of coordinates of the shape points. The shapes are modified by changing the PCA features.

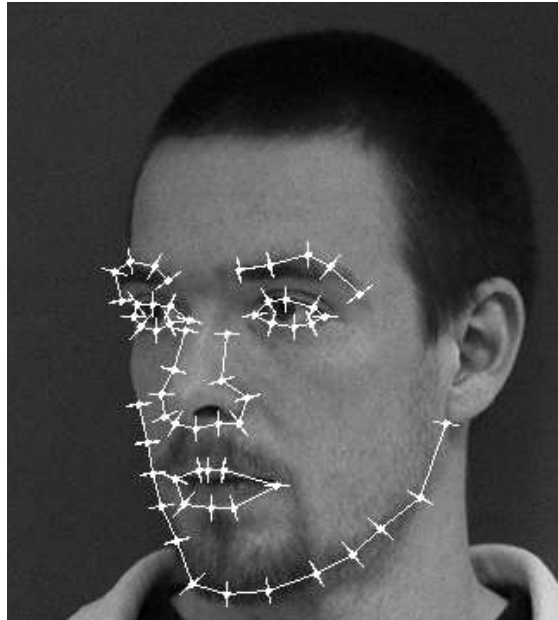


Figure 2.2: An ASM of a human face with 58 points and norms shown. (IMM database [2]).

Figure 2.2 shows an example of ASM applied to a human face. This particular model consists of 58 points.

The model was extended by Cootes to employ a texture within the object, in addition to a shape, for more complete representation. This new model is called an active appearance model (AAM). A comparison of ASM and AAM can be found in [36].

2.2.3 Symmetry operator

Facial features such as eyes, eyebrows and mouth have shapes and locations symmetrical about a vertical line through the centre of the face. Thus Reisfeld and Yeshurun [31], Saber and Tekalp [37] and Jacquin and Eleftheriadis [38] employ symmetry operators for features localisation. Reisfeld and Yeshurun locate features in points with the highest values of symmetry measure. They report that their method is robust to changes in scale and rotation. However, whenever the location of the face area is unknown, it is highly computationally expensive. A precision of 95% was reported in experiments with the database containing facial images with facial area occupying 15% – 60% of the image area [31].

A symmetry functional is used for eye localisation within an elliptical face blob by Jacquin and Eleftheriadis [38]. They use a rectangular window such as the eyes-nose-mouth model and locate features by scanning the rectangular search area with that window, followed by matching using a symmetry functional. Jacquin and Eleftheriadis [38] report 95% accuracy of tracking facial features and the method's robustness to conditions such as the presence of eye-glasses or beard.

2.2.4 Hough transform

The Hough transform is a transform used to find analytically described shapes [31, 39]. The transform was used for eye extraction by Nixon [39]. He uses magnitude and directional information of the intensity gradient for localisation of analytically described shapes. There are two shapes defined for an eye region: the iris and the perimeter of eye sclera. The iris is represented by a circle and the sclera is defined as an ellipse. The ellipse is deformed with an exponential function to more precisely fit eye shape in the corners further from the centre of the face. A Sobel operator is used to obtain gradient magnitudes and directions. The accuracy of localisation of the centres of irises reported in [39] is on average 0.33 pixels. The mean value of error of sclera localisation is 1.33 pixels.

2.2.5 AI algorithms

Artificial neural networks

A very simple approach to facial components extraction using artificial neural networks (NN) is based on a multilayer perceptron (MLP). Vincent *et al* [33] have described experiments with the MLP for eyes extraction from greyscale facial images. The images they use contain frontal views of faces and shoulders. The localisation process is performed by scanning the image or a part of the image with a rectangular window. The pixels covered by the window are inputs to the MLP. The searching area does not cover the whole image but is restricted to a

rectangular area in the neighbourhood of eye regions when the higher resolution images are processed. This speeds up the processing time and increases training efficiency since the number of examples containing an eye region appears more often when the scanning area is reduced [33].

The MLP was trained (using the backpropagation algorithm) to give response 1 when the window was centred over an eye and 0 otherwise in the case of low resolution images. For higher resolution images the response was set between 0 and 1 when the searching window was centred within the six pixels distance from the centre of the eye, increasing towards the eye centre. Vincent *et al* [40] report the results of between 86-91% success. However, they also report a problem with spurious detections.

Several methods are proposed for increasing the efficiency of the MLP eye locator. Hand *et al* [41] present a fast method of projecting weights from the neural network of one level (i.e. lower resolution) to another level network. Since the lower resolution images were obtained by subsampling original images by the factor of 2 (height and width of the image each time are reduced to half of their values), the Quad Tree method is used for projecting weights to another level. This increases the processing speed using the multiresolution approach and weights projection, as well as decreasing the number of spurious localisations.

Vincent *et al* [40] improve neural network performance by using second degree neurons. Such neurons can produce more complex decision surfaces unlike simple neurons, which have a decision surface of a single hyperplane. Good results have been obtained for verification of eyes localisation using second degree neurons for lower resolution images.

Debenham *et al* [42] propose to use radial basis functions (RBF) as the activation functions for neurons in the hidden layer. The neurons with radial basis functions (called RBF neurons) are capable of producing a finite (closed) decision surface while neurons with conventional activation functions give an infinite single hyperplane. The multiresolution experiments with RBF neurons presented by Debenham *et al* [42] gave good results in the localisation of eyes.

Reinders *et al* [43] propose a few modifications to the simple MLP (multilayer perceptron) extractor. They reduce the search area by prediction of the most probable feature location from the previous frames. However, the location of eyes in the first frame has to be done manually or by a computationally expensive search in the whole image. In order to solve the problem of wide variations of eyes appearance in the images of different people, Reinders *et al* [43] introduce the idea of micro-features. They propose four micro-features: left and right hand side corners and top and bottom eyelids. Additionally they introduce a probabilistic approach for post-processing the results produced by the neural network. The experiments made by Reinders *et al* show an average of 96% success in micro-

feature location using neural networks with probabilistic post-processing.

2.2.6 Genetic algorithms

Lin and Wu [3] use a genetic algorithm for fitting templates to facial components in an image. They construct simple templates for every feature, like eyes, eyebrows, mouth or nosetip. The templates are matched to the features in particular areas of the face. Figure 2.3 shows a block diagram of the genetic algorithm applied for facial feature template matching [3].

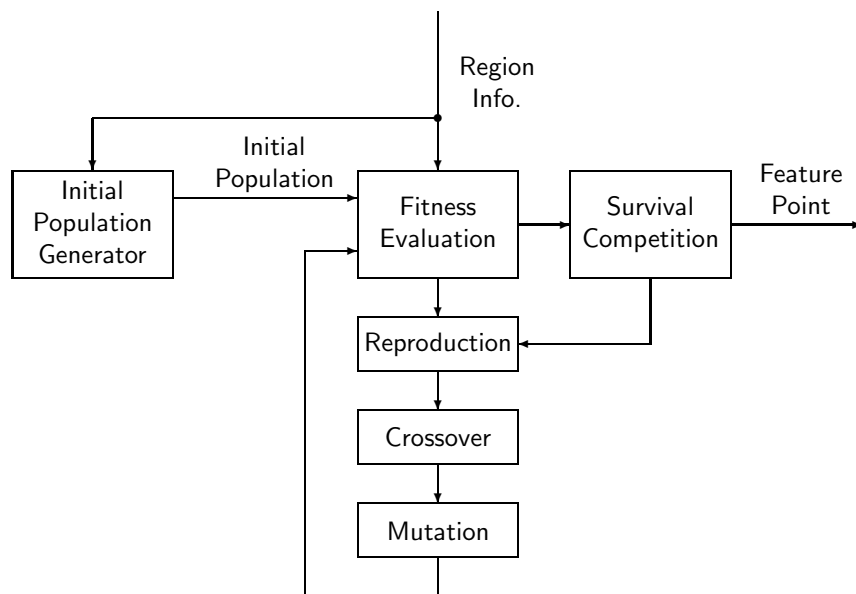


Figure 2.3: Block diagram of the genetic algorithm adopted for facial features extraction [3].

Huang and Wechsler [44] use genetic algorithms to evaluate finite state automata (FSA) and Decision Trees (DT) in order to locate eyes. First they create a saliency map of potential eye locations using FSA. The FSA moves through the image using information derived from the present location and feature map. The feature map consists of mean, standard deviation and entropy of the intensity values of the image. Once the saliency map is obtained it is examined to see if it contains the eyes. The eye recognition is based on Decision Trees combined with the genetic algorithm. To avoid redundant eye locations the winner-take-all algorithm is applied to post-processing during the eye recognition stage [44]. Huang and Wechsler report a success rate of 95% eye localisation without any false classifications. They also report that removing the saliency map creation stage results in increasing processing time and decreasing success rate to 87.5% [44].

2.3 Subspace techniques

2.3.1 Principal Component Analysis

Principal Component Analysis (PCA) was proposed by Turk and Pentland as a technique of face recognition [45]. They propose the projection of facial images into a space of eigenvectors called “eigenfaces”. The faces can be then classified by comparing their position in the eigenfaces space.

Spors *et al* [46] and Talmi and Liu [47] adopted this technique of eigenvectors to eye localisation purposes. They call the eigenvectors of eyes the “eigeneyes” in the same manner as facial eigenvectors are called “eigenfaces”. The PCA technique was also used for extraction of other facial components, so eigennooses, eigeneyes and eigenmouths were created [47].

2.3.2 Independent Component Analysis

Independent Component Analysis (ICA) is a technique based on higher order statistics. The main applications of this technique are Blind Signal Separation (BSS) and feature extraction. Takaya and Choi [48] adopted ICA for eyes and mouth extraction from facial images.

Once Takaya and Choi compute the basis vectors, they use simple heuristics to find facial features within basis images. The technique based on ICA requires knowledge of the position of the face and that can be achieved by colour segmentation. However, a method based on wavelet subband filtration is proposed in [48] to remove the components occupying large areas, leaving details only. Then the facial localisation is no longer necessary.

2.4 MPEG-7 Face Location

The Motion Picture Expert Group (MPEG) has proposed an algorithm for localisation of facial areas within colour and monochrome pictures for the purpose of extraction and creation of the MPEG7 Face Recognition descriptor [24]. This algorithm is based on such techniques as skin colour segmentation, Gaussian derivative filtering, heuristics and template matching. The algorithm for face detection confirmation has been proposed as well. The correlation between the facial images and the face-valley and face-ridge filters is used for confirmation of the face candidates.

2.5 Other techniques

2.5.1 Gabor functions extensions

Gabor functions are broadly used to improve the performance of other techniques of facial features extraction [31, 33, 48, 34]. These functions locate the high curvature points, thus they can be used for finding eye and mouth corners, nostrils, etc. [31, 34].

Lisboa [49] proposes to use wavelet coefficients as the input to an artificial neural network instead of the intensity values. This decreases the input size from a 16×16 pixel matrix to 72 wavelet coefficients. In spite of reducing the number of inputs, the results obtained with this method are similar to results obtained with the normal neural networks.

Feris *et al* [34] use a two-level hierarchical wavelet networks for localisation of corner points of eyes, mouth and nostrils. The first level is used for face matching, the second one for feature localisation. The Gabor wavelet transform in conjunction with a bijective geometric transformation is used in the first level wavelet network. Feris *et al* use brute-force search within the facial area in order to find a location that gives the minimum difference between the wavelet network and the target feature [34].

2.5.2 AdaBoost

The “boosting” technique was introduced for face localisation by Viola *et al* [50]. However they also apply the technique to find the eye region within the facial region in an image.

The AdaBoost method uses a cascade of “weak” classifiers. This structure, with appropriate training using boosting technique gives an efficient “strong” classifier. By using simple and quick weak classifiers the technique can be fast and very efficient.

Due to characteristics of the detector and its simplicity, this method gives excellent results in the case of frontal views of faces, without in-plane rotations. Detecting faces and their components in the case of in-plane rotated faces requires running the algorithm at different rotation angles. This, due to the simplicity and speed of the algorithm is not difficult to do.

2.5.3 Fuzzy logic

Johnson *et al* [51] present a neural technique for feature extraction based on fuzzy logic. The fuzzy logic is applied to find the corners within the image, which are used to form critical points. Five critical points are introduced: the eye, the tip of the nose, the point under the nose, the lips and the chin. The feature vectors

are formed based on relative distances and angles between critical points. The recognition is based on the feature vectors and is robust to translation rotation and scale [51]. The authors report a significant reduction in the computational cost of their approach in comparison with traditional neural networks with 98% success rate in feature extraction.

2.5.4 Corner detection

The SUSAN corner detection operator was proposed for facial feature extraction by Wu *et al* [52]. They obtain face area and facial features regions using colour information in conjunction with information of edge strength and orientation. Then the SUSAN operator is used to locate corner points of features to detect features points. They report that the system is invariant to changes in translation, slight rotations and facial expressions [52].

2.5.5 Colour segmentation

A use of full colour representation for the extraction of facial components was first proposed by the author in [26]. This technique, based on colour segmentation, is described in detail in Chapter 3.2.

Cooray *et al* [28] propose a similar technique based on colour segmentation and probability distribution. Firstly they find the facial region using skin colour information. Then they obtain feature candidate regions by applying the RSST algorithm to the chrominances of a facial image. A region which is inside the facial area and has a dominant red value is assumed to represent the mouth. The eyes are found as the regions with the largest values of luminance variations. The authors report good results for good quality frontal view images.

2.6 Summary

This chapter presents techniques and algorithms developed over the years for the facial components localisation problem. There are very precise methods such as e.g. deformable templates which give very accurate locations of the components. However, this comes with the cost of computational complexity and initialisation issues.

In the next chapter, the author proposes a novel approach to facial components localisation that is based on colour segmentation. This is followed by a description and results of the experiments carried out by the author in order to evaluate the proposed algorithm. Another approach based on a PCA template is presented for comparison.

Chapter 3

Face Component Localisation

3.1 Introduction

In the previous chapter techniques and algorithms for facial components localisation developed over the years by many researchers are described. These algorithms usually deal with gray level facial images in a fixed scale environment (although there are exceptions such as deformable templates).

In this chapter, a method for locating eye positions is proposed by the author and analysed. It is based on colour segmentation techniques and can be used for locating any face components that differ in colour from the human skin, such as eyes, lips, and eyebrows. The colour segmentation is realised with Recursive Shortest Spanning Tree (RSST) and Expectation-Maximisation (EM) algorithms, and components are classified using simple heuristics. Another technique presented in this chapter locates eye positions within a bounding box outlining the facial area. It is based on PCA projection of the candidate eye regions. This method is presented here for comparison with the one proposed by the author.

3.2 Facial components extraction via image segmentation

3.2.1 Algorithm overview

The algorithm proposed by the author for facial components extraction and localisation consists of three steps:

- initial segmentation
- face localisation
- facial components extraction.

The algorithm uses initial segmentation to obtain an initial set of regions that are used for the final object extraction. The object of interest (in this case a human face with facial components) needs to be specified by the user. Two approaches are investigated: the first one uses lines (scribble) drawn by a user to mark the human face and the background; the second one employs an automated face localisation algorithm. Once the facial object in an image is specified, the final extraction of components from the object is carried out.

The initial segmentation (presented in detail later in this chapter) is carried out using the RSST algorithm similarly to Cooray's *et al* [53] approach. Other automated segmentation techniques are available as well, including the watershed used for initial segmentation by O'Connor [54], pyramidal region growing or colour clustering [55]. The RSST was chosen because it produces connected regions due to the fact that it merges neighbouring regions and it thus provides a good initial segmentation for facial components extraction. This is an efficient algorithm with easy control of the segmentation granularity [53, 55]. Section 3.2.2 provides more details about the RSST.

The second step, face localisation, is provided with simple skin colour segmentation. However, initially, simple user interaction was used for localising the facial region, requiring the user to draw a simple line over the facial object and another one over the background. This user interaction is the same as used by O'Connor [54] who draws lines (called *scribbles*) over the object of interest and the background. This approach is different from the Cooray approach [53] where the user clicks on the regions of interest within the object.

The face localisation algorithm is used in order to remove the user interaction. The algorithm is similar to the method proposed by Sobottka *et al* [56, 27], however, the skin colour segmentation is performed on the region level delivered by the initial segmentation instead of on the pixel level (as it is in Sobottka's approach). For each separated skin-coloured region the best matching ellipse is calculated. The ellipse approximates the head blob [56] and is used for marking the facial object and the background. These objects do not necessarily have to be marked very precisely since the final step of extraction provides accurate boundaries. However, better skin segmentation provides more accurate information for further processing. Details about the face localisation are presented in Section 3.2.3.

The last stage of the algorithm is the final extraction of the facial components. The EM algorithm described in detail in Section 3.2.4 is employed at this step. Since the face is represented by a multimodal Gaussian distribution, the facial components may be represented by the separate modes. Thus, when the facial object is extracted from the image, the modes are searched for components. In the next three sections, the algorithms are described in more detail.

3.2.2 Initial segmentation — the RSST algorithm

Recursive Shortest Spanning Tree (RSST) is based on graph theory which was successfully used for the image segmentation [57]. This hierarchical automatic algorithm segments the image iteratively from a huge number of small regions (on the pixel level) to several regions with relatively large area (which is the final result in this process).

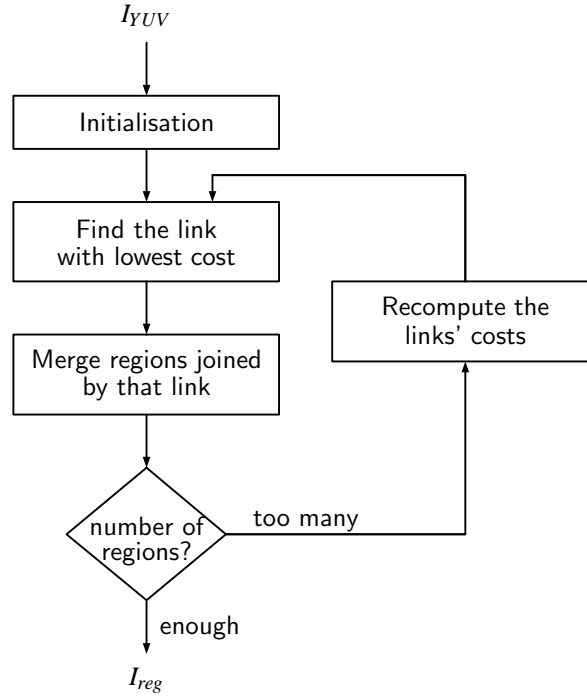


Figure 3.1: The RSST algorithm diagram.

A block diagram of the RSST algorithm is shown in Figure 3.1. The process begins with an initialisation step, where each pixel is mapped as a separate region and the links between adjacent pixels are created. Then, the cost of each link is calculated using the equation [53, 58]:

$$d(R_i, R_j) = [(Y_{R_i} - Y_{R_j})^2 + (C_{r_{R_i}} - C_{r_{R_j}})^2 + (C_{b_{R_i}} - C_{b_{R_j}})^2] \times \frac{N_{R_i} \times N_{R_j}}{N_{R_i} + N_{R_j}} \quad (3.1)$$

where R_i and R_j denote the regions joined by the link with the cost (distance) $d(R_i, R_j)$, Y_R is the mean value of the luminance of the region R , C_{r_R} and C_{b_R} are the mean values of chrominances of the region R , and N_R denotes the number of pixels in the region R (the size of this region).

Once the costs for all links are obtained, the regions with the lowest link cost between them are found. The next step is to merge these regions, which gives a new region with new mean values of luminance and chrominances. These values must be recalculated. The size of the new region is the sum of sizes of merged regions. Every link belonging to one of the regions being merged is attached to the newly created region. Then, all the duplicated links are removed, since they

are not needed any longer. The number of regions is decreased at each iteration.

Next, the number of regions is examined. If there are more regions than specified by the user, the link costs must be recalculated for the new region. Then the next iteration is carried out.

When the specified number of regions is reached, the merging process is stopped and the regions are mapped to the image, giving the final segmentation. Because the final number of regions depends on the characteristics of the image, Cooray *et al* [53] introduced the PSNR (Peak Signal to Noise Ratio) as the criterion of the convergence of the RSST algorithm [53]. However, this is achieved at the cost of increased processing time due to the calculation of the PSNR at every iteration.

3.2.3 Face localisation — skin colour segmentation

The initial face localisation is introduced in order to remove the user interaction from the algorithm. The facial areas are found by examining the RSST regions in order to find skin-coloured regions. Then the adjacent regions are merged and the separate disconnected skin coloured regions are obtained. Since this chapter is not focused on face localisation, it is assumed that the facial image contains a single face and the face occupies a biggest region with skin colour within the image. The face localisation technique described in the remainder of this section is based on the algorithm developed by Sobottka and Pitas [56, 27]. It differs from the original approach in that the colour of regions is analysed, not the colour of pixels as it is in the original algorithm.

Firstly the search for the skin coloured regions is carried out. The RSST regions are represented by mean colour so that colour information with thresholding is used for finding potential facial regions. The thresholds are defined in the YUV colour space and work on the chrominances values. Sobottka and Pitas [56, 27] use the HSV colour space, but since the RSST and EM algorithms use the YUV representation and there is no need to make additional conversion to HSV space. The thresholds are set to bound the area of skin colour extracted from the sample facial images. The sample points and the bounding lines are presented in Figure 3.2.

The area in the colour space corresponding to skin tones, based on the points presented in Figure 3.2 is defined with the following equations:

$$Cr > 80 \tag{3.2}$$

$$Cb > 140 \tag{3.3}$$

$$Cb - 0.67Cr < 107 \tag{3.4}$$

$$Cb + 1.6Cr < 356 \tag{3.5}$$

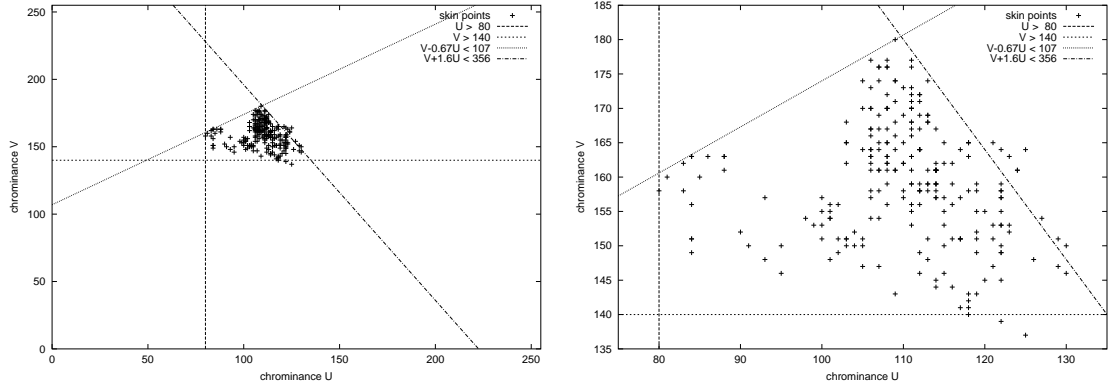


Figure 3.2: The points of the skin coloured pixels on the UV plane and the lines bounding the skin colour area.

These equations were obtained by the author and they are different from the ones presented by Sobottka and Pitas, because a different representation of colour is used and partially because of a different dataset. It can be seen in Figure 3.2 that some of the skin coloured points are outside the area defined by the above equations. However, this does not affect the overall process of facial components extraction since only a rough idea about the facial regions is required. If the definition of the skin colour area is too wide, it can result in classifying the background regions as the facial regions, which affects the extraction results. Additionally the dark areas, which usually represent the hair, are removed by excluding all the regions with the mean value of luminance below 100. Figure 3.3 presents examples of skin regions that were found using the thresholds defined above.

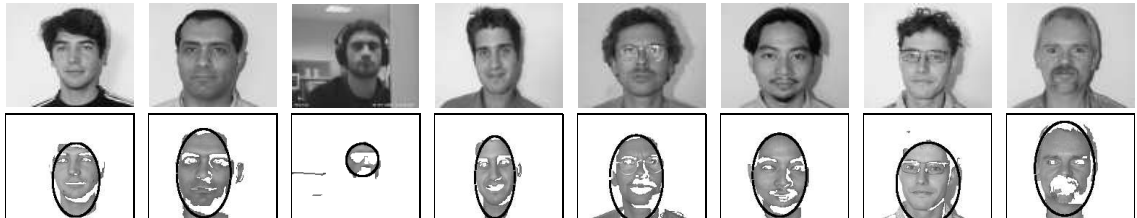


Figure 3.3: The sample facial images with the skin colour regions extracted using thresholding. The upper images are originals while the bottom ones show the extracted skin regions and the best fitted ellipses.

The neighbouring skin colour regions are merged in the next step, thus connected larger regions are created (these regions are separated from each other). The ellipses that fit best to these regions are calculated. The ellipse computation is based on the work of Sobottka and Pitas [27]. The ellipse is defined with its centre point x_m, y_m , the lengths of two axes a and b and the orientation θ [27]. The centre point of the ellipse is calculated as the centre of gravity of the region:

$$x_m = \sum_{i=0}^N x_i, \quad y_m = \sum_{i=0}^N y_i, \quad (3.6)$$

where N denotes the number of pixels within the region.

The orientation of the ellipse and the lengths of the axes are computed using moments, which lead to the following equations [27]:

$$\theta = 0.5 \arctan \left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right) \quad (3.7)$$

$$a = \left(\frac{4}{\pi} \right)^{1/4} \left[\frac{(I_{max})^3}{I_{min}} \right]^{1/8} \quad (3.8)$$

$$b = \left(\frac{4}{\pi} \right)^{1/4} \left[\frac{(I_{min})^3}{I_{max}} \right]^{1/8}, \quad (3.9)$$

where

$$\mu_{p,q} = \sum_{i=0}^N (x_i - x_m)^p (y_i - y_m)^q \quad (3.10)$$

$$I_{max} = \sum_{i=0}^N [(x_i - x_m) \cos \theta - (y_i - y_m) \sin \theta]^2 \quad (3.11)$$

$$I_{min} = \sum_{i=0}^N [(x_i - x_m) \sin \theta - (y_i - y_m) \cos \theta]^2. \quad (3.12)$$

The ellipse of the biggest region in the image (it is assumed that the image contains no more than one face and that this face occupies the biggest of the skin coloured regions) is used to produce two stamps to mark the facial and background areas.

3.2.4 Facial components extraction with the EM algorithm

An image as the set of objects and regions

An image, unless this is a pattern image such as a chess board, usually contains meaningful regions called objects. Thus the image can be considered as a set of objects, at least two: a foreground and a background. The background particularly is not necessarily a meaningful object.

The object can be seen as the set of regions, e.g. a human body contains the regions such as a head, shoulders, arms, legs etc. The facial object is the object of the special interest in this chapter. The face can be considered as consisting of regions such as eyes, lips, nose, eyebrows etc.

Regions consist of pixels characterised by relatively high homogeneity and usually their structure is not very complicated. On the other hand, the complexity of objects might be very high and they might contain regions of very different characteristics. Among the features describing a region there are low-level characteristics such as both vertical and horizontal coordinates and the mean colour of the region. The number of these features depends on the representation of the

colour, usually there are three colour coordinates. This gives five features in total — two spatial coordinates and three coordinates of mean colour.

The Gaussian distribution is a very common model for real signals. This convenient model is used for modelling the regions within the objects and particularly the face. The multivariate (multivariate since there are several variables (features) describing the region) Gaussian Probability Distribution Function (PDF) is defined as [54, 59]:

$$\text{pdf}(x, \theta) = \frac{1}{(2\pi)^{\frac{k}{2}} |\theta_2|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\theta_1)^T \theta_2^{-1} (x-\theta_1)}, \quad (3.13)$$

where $\theta = [\theta_1 \ \theta_2]^T$ is the set of parameters, $\theta_1 = m_x$ denotes the vector of the mean values of the features and θ_2 is the covariance matrix. The value of k equals to the number of features, which determines the dimension of the θ_1 vector (in our case $k = 5$). When the features (the components of the parameter vector) are independent, the covariance matrix consists only of variances of the features and is denoted by [54]:

$$\theta_2 = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k^2 \end{bmatrix} \quad (3.14)$$

Figure 3.4 (a) presents the histogram of the colour of the eye regions in the YCrCb colour space. The Gaussian distributions modelling the eye features are shown in Figure 3.4 (b). The model obtained with the Gaussian PDF gives good representation of the regions [54], such as eye or lips regions and is used for modelling of facial image.

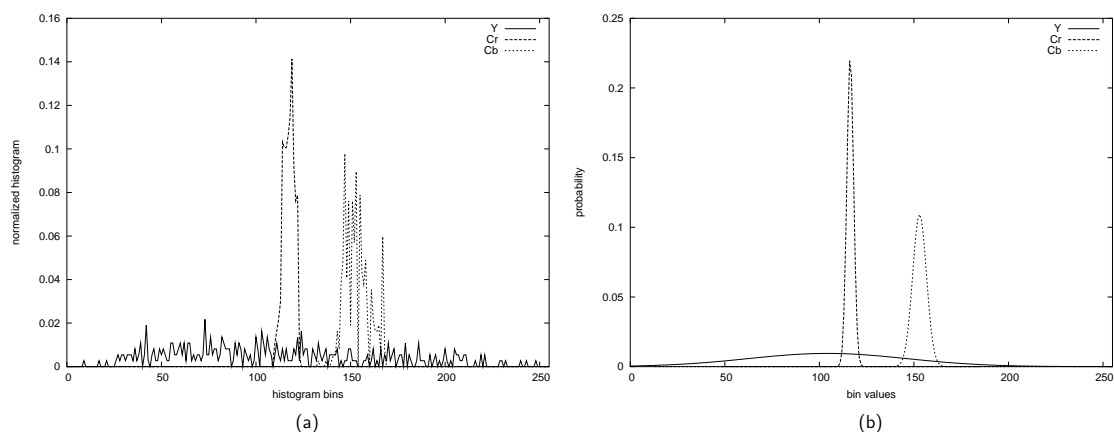


Figure 3.4: The histogram (a) and the Gaussian model (b) of the colour components of the region representing the eyes.

The full facial object requires more complex representation in order to reflect its more complicated structure. Such an object can be viewed as a set of regions, therefore can be modelled with the superposition of the regions PDFs. The Gaus-

sian model of each region within the object is called a *mode* and the PDF of the object is then named a multimodal Gaussian distribution. This multimodal object PDF can be obtained as the weighted sum of each region PDF [54]:

$$\text{pdf}(x, \theta) = \sum_{g=1}^G \pi_g \text{pdf}_g(x, \theta_g), \quad (3.15)$$

where $\text{pdf}(x, \theta)$ denotes the object PDF, G is the number of modes (regions within the object) and $\text{pdf}_g(x, \theta_g)$ is the PDF of the g th region. The symbol π_g denotes the weight at which the g th PDF is summed. If the condition $\sum_{g=1}^G \pi_{gi} = 1$ is met then π_g is considered as the given (*prior*) probability that the x th pixel belongs to the region g .

An example of a multimodal distribution modelling the facial object is presented in Figure 3.4. Only the luminance component is shown for better visualisation. The object consists of four regions — thus there are four modes which form the distribution. To show clearly how the multimodal distribution is created, the sum of modes is not weighted and is not normalised. It can be seen, by examining the shape of the distribution function, that in this case just two modes would be enough to represent the facial object sufficiently. However, if the facial components are to be found, the number of modes has to be increased, so a separate mode is applied to each component. By doing so the object may be over-represented, however this over-representation is justified when the facial components, rather than the facial object, are to be extracted.

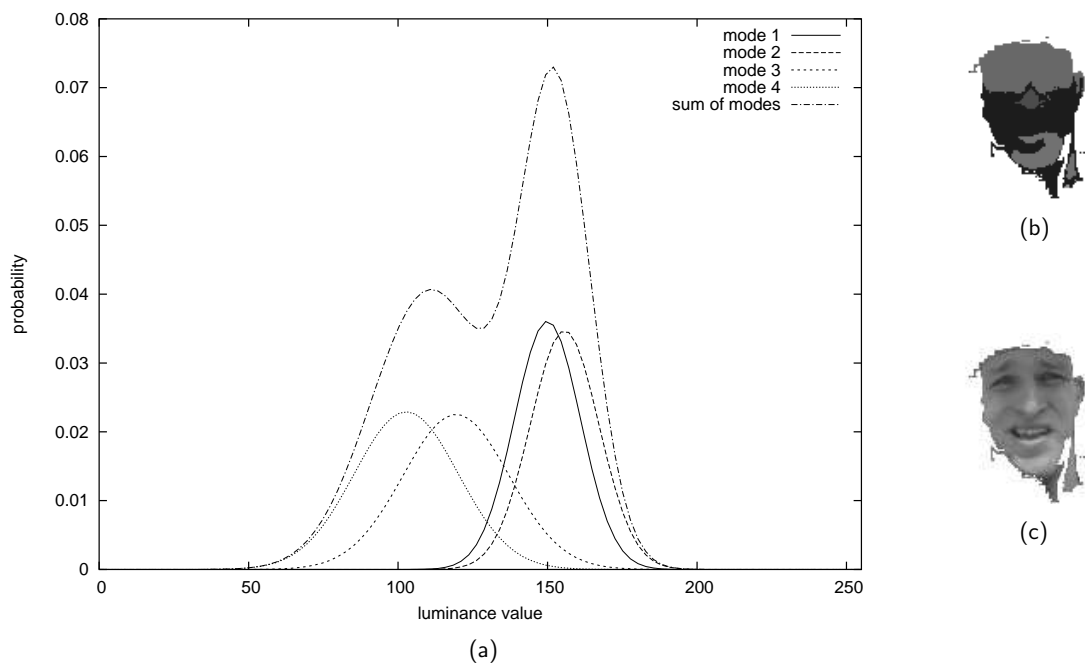


Figure 3.5: An example of multimodal distribution as the model of the facial object.

Since the image is considered as a set of objects (i.e. faces and the background)

and each object is modelled with a multimodal PDF, the image can be seen as a mixture of the multimodal PDFs. The PDF does not have to be a Gaussian distribution in general. However, this distribution is very similar to the distribution of colour samples of the facial features (see Figure 3.4) and thus was chosen for modelling the facial components.

Facial component extraction with the EM algorithm

This section describes the EM algorithm applied for extraction of the facial object and its components. It is assumed that the image of the face is a frontal view shot of the face and both eyes are visible. The head is not in-plane rotated or else is rotated with a very small angle, so it is assumed that the components are horizontally aligned. In Section 3.3 it is assumed that there is only one face in an image. In the case of multiple faces in an image, the image can be seen as a set of M objects: $M - 1$ faces and the background. Alternatively the image can be split into $M - 1$ subimages, each containing only one face, thus consisting of two objects: the face and the background. In the description of the algorithm the latter approach is taken, with M objects (faces and the background). The flow chart in Figure 3.6 shows the structure of the EM algorithm used for the object extraction. The very first step is the calculation of the initial parameters of the multimodal PDFs modelling the objects. This requires some information about the object. Then the iterative loop consisting of two steps is run. These steps are the E-Step (Expectation Step), where the estimations of the posterior probabilities are calculated, and the M-Step (Maximisation Step), where the updates for the parameters are computed in order to maximise the expectation of the likelihood function.

The algorithm is run on two levels — the object level, where the objects (faces and the background) are represented with the multimodal PDFs, and the region level where the regions (facial components or background components) are modelled with the unimodal Gaussian PDFs and the set of regions forms the object PDF.

The initialisation step

The initialisation step provides the initial values for the parameters of the objects and the regions. Some information about the objects has to be known giving some data at the start of algorithm. This data does not have to be complete. However, the more data available the better the objects are represented. The information about the face and the background is provided by the user, who draws lines (scribbles [60, 54]) on the objects or is derived by the face localisation algorithm presented in Section 3.2.3.

Since the image is pre-segmented with the RSST algorithm, lines (scribbles

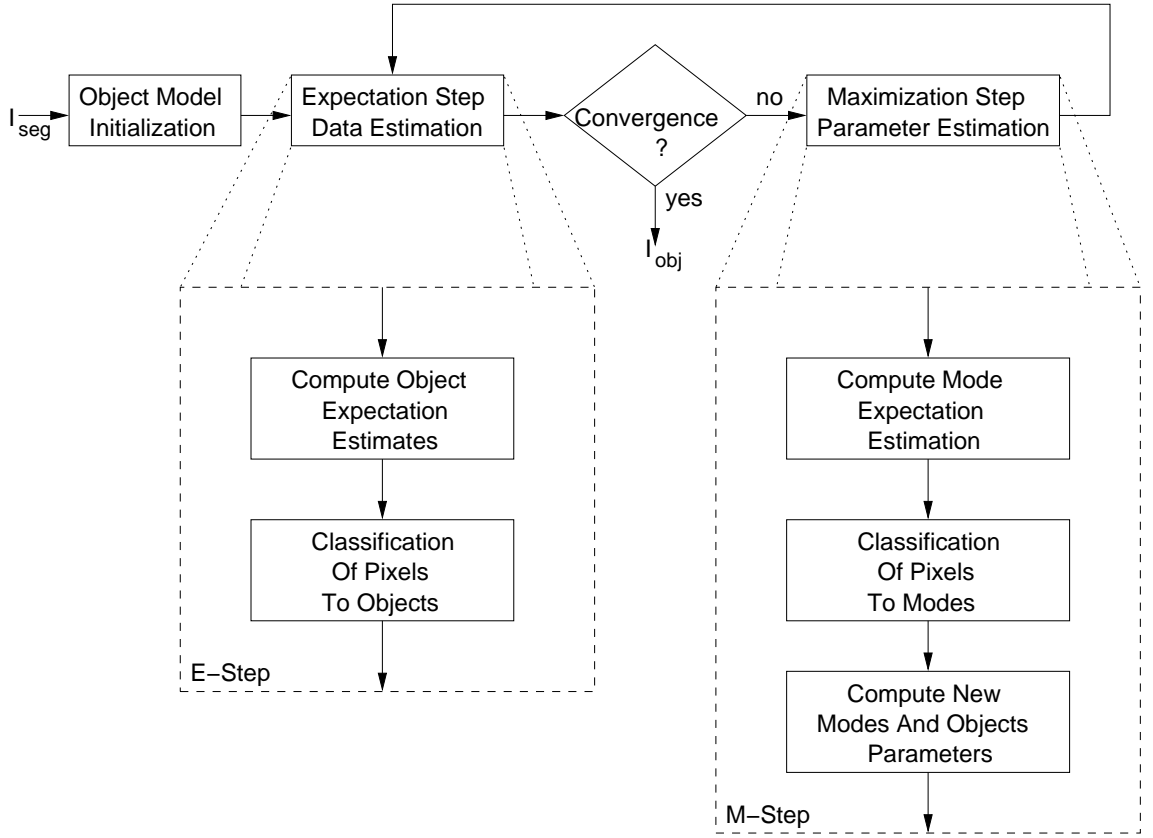


Figure 3.6: The object extraction via the EM algorithm — a system diagram.

[60, 54]) are drawn over the regions. The regions marked with these lines are merged until the number of the regions within each object reaches the specified value. It is assumed that the number of marked regions is higher than the number of modes. If this is not true then some of the modes represent no region and should be removed. Then, either the number of modes should be decreased, or the number of regions of the initial segmentation should be increased.

As can be seen in Figure 3.5, two regions representing the facial object should be sufficient for the facial object extraction. However, these regions would not contain the separated facial components. Thus the number of regions has to be as high as the number of components or higher. Then each facial component is represented with a separate region what provides the components extraction.

The clustering algorithm used for the regions merging is similar to the RSST algorithm in the way that it iteratively merges the scribbled regions spanned with the lowest link cost. The cost of the link is:

$$d(R_i, R_j) = [\mathbf{f}_{mR_i} - \mathbf{f}_{mR_j}]^T [\mathbf{f}_{mR_i} - \mathbf{f}_{mR_j}], \quad (3.16)$$

where the feature vector \mathbf{f}_{mR} contains the features of the R th region. Since it is assumed that the components are horizontally aligned, the vectors \mathbf{f}_R contain the vertical coordinate value and colour components $\mathbf{f}_R^T = [Y_R \ Cr_R \ Cb_R \ y_R]$ where y_R denotes the vertical coordinate of the centre of weight of the region R . Additionally,

it is worth mentioning that disjoint components like eyes, eyebrows or nostrils are represented with the same region, which means that for example one region contains both eyes and there is not a separate region to represent each of the eyes.

Once the required number of regions within each object is obtained, the initial values of the parameters are calculated. The mean and variance values of the G regions (modes) PDFs are calculated using equations:

$$m_{gl} = \frac{1}{N} \sum_{n=1}^N x_{gnl} \quad (3.17)$$

$$\sigma_{gl}^2 = \frac{1}{N} \sum_{n=1}^N (x_{gnl} - m_{gl})^2 \quad (3.18)$$

where $g = 1, \dots, G$ denotes g -th region and $n = 1, \dots, N$ denotes n -th pixel. The parameters l are colour components in the YCrCb space and the spatial coordinates, which provides the five dimensional PDFs. The prior probability must also be assigned to each region. Since the knowledge at the beginning is not complete, an equal value of the prior probability is assigned to every pixel which is not scribbled. Thus the probability becomes $\pi_{gn} = 1/G$ for $g = 1, 2, \dots, G$ and $n = 1, 2, \dots, N$ where N is the number of pixels in the image and the G is the number of modes within the object. The prior probability of the pixels belonging to the marked regions are assigned $\pi_{gn} = 1$ if they belong to the g th region, and $\pi_{gn} = 0$ otherwise.

Once the PDFs and prior probabilities for the regions (modes) are obtained, the initial object PDFs $\text{pdf}_m(x_n, \theta_m)$, $m = 1, 2, \dots, M$, where M denotes number of objects, are computed using the Equation 3.15. If the n th pixel does not belong to any object (its region has not been marked) the prior probability π_{mn} , $m = 1, 2, \dots, M$ that the pixel belongs to m th object is set to $1/M$. If the n th pixel was marked as belonging to the p th object its prior probability $\pi_{pn} = 1$ and $\pi_{mn} = 0$ for all $m \neq p$.

The expectation step

The Expectation Step (E-Step) is the first step in the iteration loop. During the first iteration, all the information and the parameters required for the calculations are provided with the initialisation step. For the subsequent iterations, the PDFs are obtained from the M-Step of the previous iteration and the posterior probabilities from the previous iteration become the prior probabilities in the current one. The actual posterior probability τ_{mn} , $m = 1, 2, \dots, M$ for each object is calculated using Bayes rule [54]:

$$\tau_{gn} = \frac{\pi_{gn} \text{pdf}_g(x_n, \theta_g)}{\sum_{g=1}^G \pi_{gn} \text{pdf}_g(x_n, \theta_g)}. \quad (3.19)$$

The calculation of the posterior probabilities gives the soft segmentation of the image into objects. A hard segmentation assigns pixels to the regions and objects

by examining the posterior probability. The n th pixel becomes the member of this object which has the highest probability τ_{mn} .

There might be some pixels which do not belong to any object. These pixels are called *outliers*. A particular pixel becomes an outlier when the posterior probabilities that this pixel belongs to any object have a very low value. The outliers are found by thresholding the posterior probabilities and are removed from the following iterations since they provide no information about objects [54].

The maximisation step

The Maximisation Step (M-Step) updates the parameters of the objects PDFs in order to maximise the likelihood expectation with the current pixel's membership of the objects. This is carried out by updating the parameters of every region within each object. Firstly, the posterior probabilities that the given pixel n belongs to the g th region within the object this pixel belongs to (from the hard segmentation of the E-Step on the object level) is calculated. Bayes rule (Equation 3.19) is used for these calculations assuming that there are G regions within the object. These calculations can be seen as the E-Step and a soft segmentation at the region level.

Once the posterior probabilities are available, the hard segmentation is carried out. The pixels are assigned to this region within the object, which has the highest posterior probability that this particular pixel belongs to this region. This classification results in the updated regions which can be seen as the estimation of the likelihood function for the current parameters [54].

The parameters of each region PDF are updated in order to maximise the likelihood estimation. The updated parameters are calculated using the equations:

$$m_{gl} = \frac{\sum_{n=1}^N \tau_{gn} x_{nl}}{\sum_{n=1}^N \tau_{gn}}. \quad (3.20)$$

$$\sigma_{gl}^2 = \frac{\sum_{n=1}^N \tau_{gn} (x_{nl} - m_{gl})^2}{\sum_{n=1}^N \tau_{gn}}, \quad (3.21)$$

These parameters form the updated region PDFs: $\text{pdf}_g(x_n, \theta_g)$, $g = 1, 2, \dots, G$ where G denotes the number of regions in the given object. The m th object PDF pdf_m is calculated using the region's PDFs with Equation 3.15.

If the convergence condition is not met, the updated PDFs and the posterior probabilities are used in the next iteration of the algorithm, and the current posterior probabilities become the prior probabilities in the following iteration.

Convergence

Convergence is achieved when the estimation of the Maximum Likelihood reaches the Maximum Likelihood point or at least gets very close to that point. One of the ways to find out when the convergence is met is to examine the changes of

the parameters θ . The EM algorithm is stopped when the PDF parameters do not change more than a given threshold. In the case of the object extraction from the digital image, the algorithm can be stopped when only very small number of pixels change their membership to the objects. In the experiments described in the Chapter 3.3, the latter condition is used and it is assumed that the convergence is reached when less than 5% of all pixels in the image change their membership in an iteration.

Post-processing

Post-processing is applied to the final facial object extraction. When the automatic face localisation algorithm presented in Section 3.2.3 is used for marking the facial object and the background, the ellipse created as the facial blob is used in post-processing. All the pixels placed inside the blob ellipse are considered as the pixels belonging to the facial object and its components. All the others are removed from the components regions giving more accurate boundaries of the facial features.

Components classification with heuristics

The algorithm described produces a set of regions which contain facial features. However, this does not determine which region contains any particular component. Simple heuristics are hereby proposed by the author to assign regions with the components. Some geometric dependences and the colour characteristics of the features are used for the classification.

It is usually the case that the components such as eyes and eyebrows are placed above the centre point of the face and the mouth is localised below this point. The eyebrows and the eyes usually have a lower mean intensity value than the facial skin. Due to the symmetrical nature of the facial components to the vertical line going through the centre of the face, the horizontal centre of gravity of the component region should remain close to the centre of the face. It is important that dual features such as eyes or eyebrows are represented with a single region so their horizontal centre of gravity should be located near the centre of the face.

The observations presented above lead to a set of heuristics rules. First of all, the mean values of the intensity of the regions are examined and their horizontal distance from the centre of the face is calculated. The Euclidean distance is calculated

$$D_E = Y_R^2 + (x_R - x_F)^2, \quad (3.22)$$

where D_E denotes the distance measure, Y_R is the mean luminance value of the region, which is in the range of $(0, 255)$, so the value of the distance measure is lower for darker regions. The symbol x_R denotes the horizontal coordinate of the centre of weight of the region and x_F is the horizontal coordinate of the centre of

the facial object. Once the distances for each region within the facial object are calculated, three regions are chosen: the region with the lowest distance measure among the regions placed below the centre of the face and two regions with the lowest distance measure above the centre of the face. The region which is below the facial centre is regarded as the mouth region. The bottom region from the two upper regions is seen as the region containing the eyes and the most upper one should contain the eyebrows.

3.3 Experiments

3.3.1 Facial components extraction with user interaction

The extraction of facial components requires somewhat different conditions than object segmentation. The scribbles that mark the objects now have to be more complicated, especially the line drawn over the face object. That line should pass through all features that are to be extracted. The background line should cover also the hair area, since it is vital that hair is removed from the facial object to avoid inclusion of the hair regions in the regions of other components. Figure 3.7 shows examples of the scribbles used for the experiments.

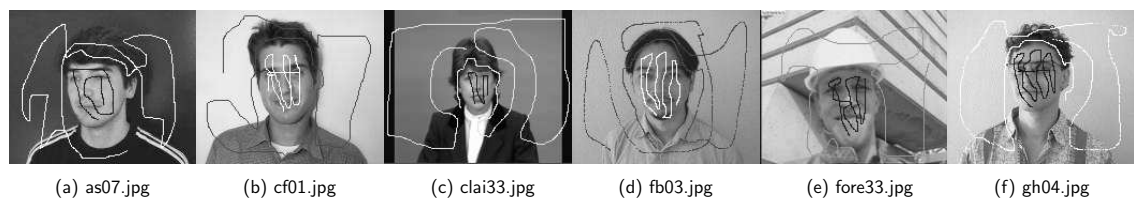


Figure 3.7: Examples of scribbles used for facial components extraction; colours of the lines drawn by the user have been changed for better visibility.

The number of modes within the facial object has to be increased due to the more complex structure of such an object. When the number of modes is too low the algorithm tends to gather surrounding image parts into component regions. The experiments were carried out for three cases: 10, 15 and 20 modes applied to the facial object. The number of modes higher than 20 provides many modes containing no pixels at all, thus experiments were not carried out for more than 20 modes for the facial object.

Figures 3.8 and 3.9 present results of eye and mouth extraction. The initial segmentation was carried out with 150, 200, 250, 300 and 600 RSST regions. The number of modes used for facial object representation was set to 10, 15 and 20. It can be seen that there is a particular number of modes and RSST regions that gives good results. In many cases the modes cover regions which do not belong to the facial components. However, it is possible to find parameters giving good results for each image.

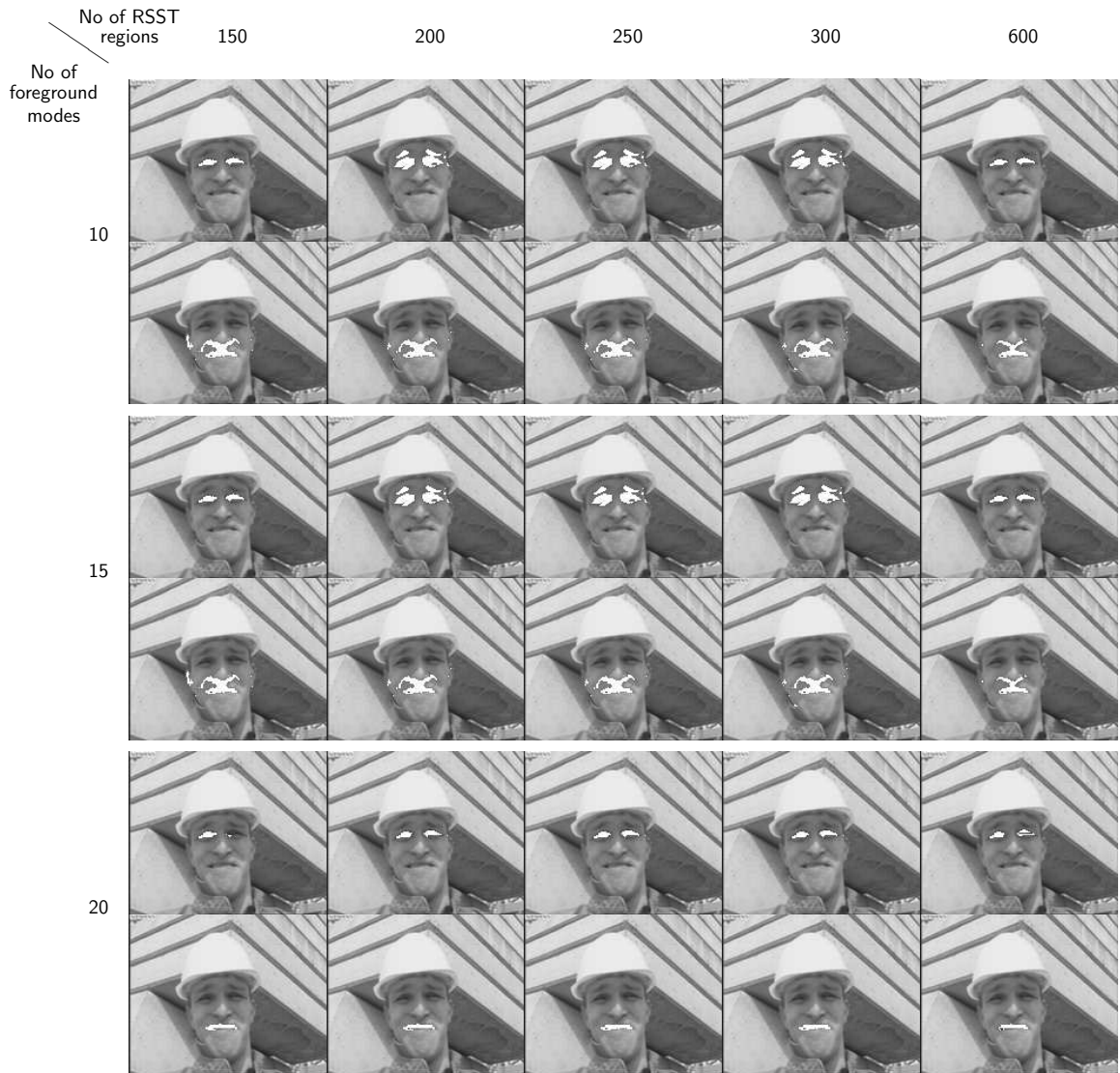


Figure 3.8: Results of facial features extraction from the frame of “Foreman” sequence.

In the first case illustrated, the “Foreman” video sequence, very good eye extraction was achieved with 20 modes applied when the number of regions of initial segmentation ranges from 200 to 300. A lower number of facial modes results in a merging of the eyebrows with the eyes. The results show that mouth extraction requires a high number of modes, the best results were obtained using 20 modes. The shadows below the nose and the nostrils are included in the lips region when the number of modes is too low.

Results obtained for the “Claire” image are less accurate. The eye extraction encounters problems caused by the hair that is regarded, by the algorithm, as the part of the eye region. Additionally the extraction of both eyebrows is impossible to achieve since one of the eyebrows is covered by hair. Some parts of hair are

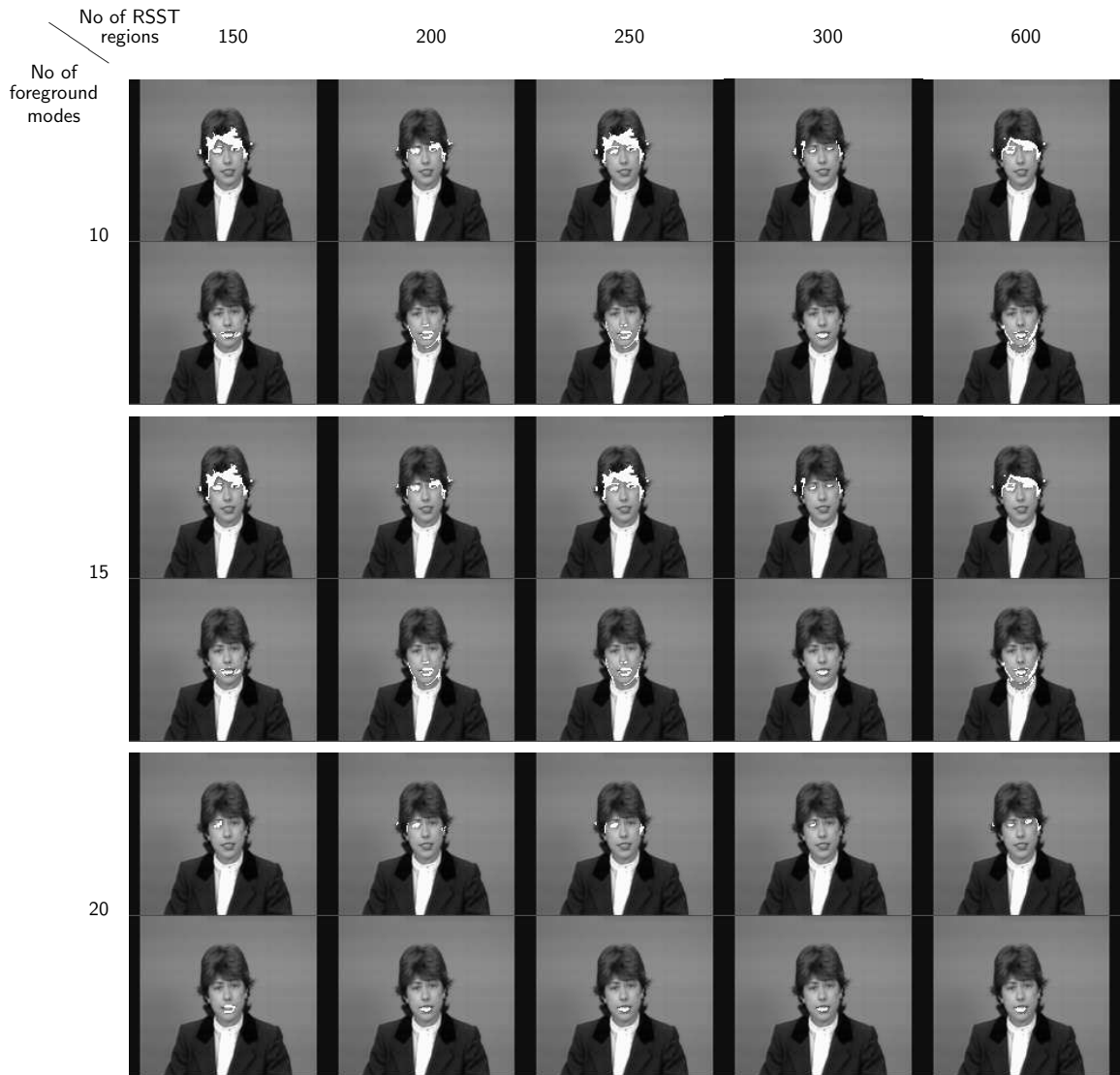


Figure 3.9: Results of facial features extraction from the frame of “Claire” sequence.

included in the eye regions. A smaller number of facial modes results in more hair being included in the eye regions. In this sequence the number of RSST regions is very important. Setting this number to 300 gives the best results according to the number of non-eye pixels included in the eye region. Even the best results require post-processing for removing hair pixels. This can be achieved i.e. by considering only these pixels that are placed within the skin coloured area of the face.

The results obtained with 20 modes applied to the facial object are an example of too many modes being used for modelling the object as shown in Figure 3.9. One of the eyes is completely excluded from the eye region except for the very fine initial segmentation level (600 RSST regions). It is desired in the presented approach (as outlined in Section 3.2.4) that both eyes are represented with single

mode, thus a mode containing only one eye is considered as an error.

The best results of mouth-lips extraction were obtained with the parameters in the middle range of their values: 200 – 250 RSST regions combined with 10 – 15 modes. In these cases both upper and lower lips were extracted giving precise vertical location of the mouth. However, the nostrils are included in the lips region. When only the lower lip is extracted it is separated from nostrils, i.e. when 20 modes are applied to facial region.

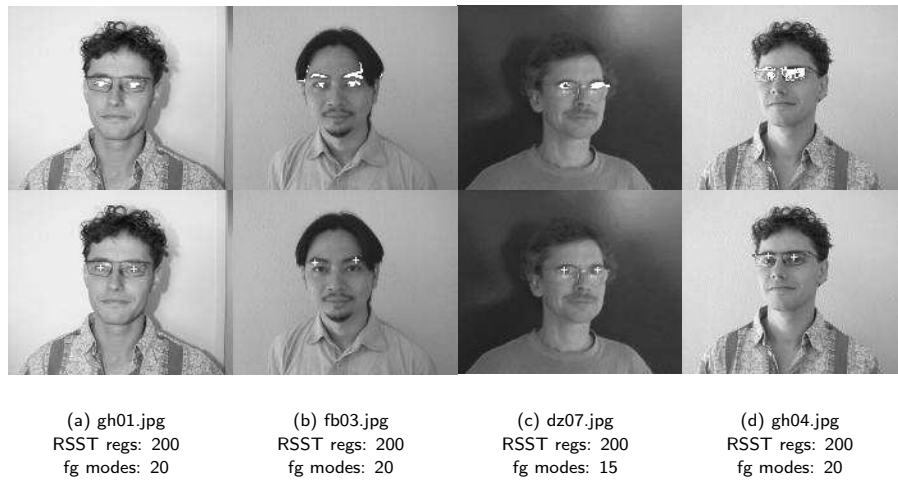


Figure 3.10: Results of eye extraction and localisation. The parameters used for extraction are presented at the bottom of each picture.

Figure 3.10 presents results of the eye extraction and localisation obtained using sample images from the HHI database. The image presented in Figure 3.10(a) shows perfect extraction and localisation of eyes in spite of the appearance of eyeglasses. Figure 3.10(b) presents an example of a false localisation, when the eyebrows are included in the eye regions. The locations of the eyes are then shifted up towards the eyebrows. The localisation errors caused by the reflections on the eyeglasses and eyeglasses frames together with the eyebrows are presented in Figure 3.10(c) and (d). In both cases additional pixels belonging to the eyeglasses add an error to the eye locations.

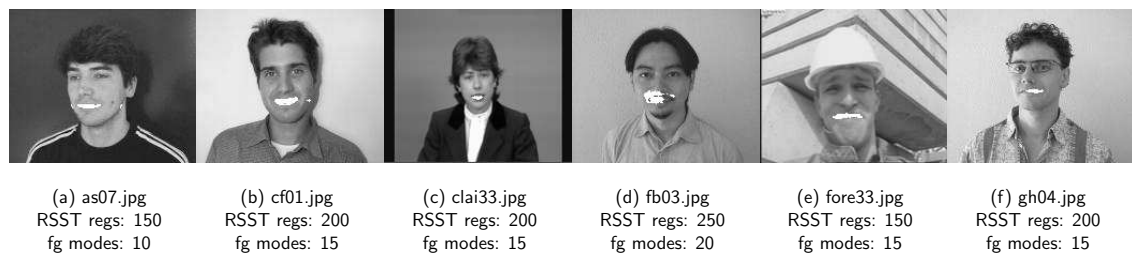


Figure 3.11: Results of the mouth extraction.

Additional results with the mouth extraction are presented in Figure 3.11. Images 3.11(a), (b) and (e) present very good results, however, some pixels from the

surrounding background or shadows are extracted as well. The algorithm gets confused by the moustache as shown in Figure 3.11(d). The lips together with the moustache and the nostrils are considered as a single mode providing false mouth extraction. Since the algorithm extracts horizontally aligned features, it is sensitive to vertical shadows. Figure 3.11(d) and (f) provide sample images with such shadow. This results in partial extraction of the mouth feature, as the remaining part differs in brightness from the extracted part.

The experiments carried out show that the presented technique can be used for facial components extraction under certain conditions. The number of RSST regions and the number of facial modes are critical for the extraction results. It is possible to find the values of these parameters that give good results. However, these values are different for each image. The lighting conditions have a great impact on the algorithm performance. This is particularly true when the vertically aligned shadows cause the algorithm to fail (see Figure 3.11 (d) and (f)).

3.3.2 Facial components extraction with automated face localisation

In order to remove the user interaction, the automated face localisation algorithm was proposed by the author as described in Section 3.2.3. The experiments that were carried out used two marking stamps created from the facial ellipse: the smaller ellipse and its interior marking the facial object, the larger one with its exterior marking the background. The gap between these two marks gives the algorithm an opportunity to extract facial boundaries accurately. However, some difficulties are encountered when the pixels in the gap are falsely classified as facial components. The set of 48 images taken from the HHI database was used for carrying out the experiments. The extraction and localisation were obtained for 150, 200, 250, 400 and 600 RSST regions and 10, 15 and 20 facial modes were used for each image.

Figure 3.12 presents some results of the eye localisation using the algorithm with the face area marked automatically. The locations of the eyes are compared with the reference location points obtained from the images containing eye masks. These mask images were created with the manually segmented eye regions. Thus they can be viewed as the “ideal” eye regions serving as a reference. The localisation errors presented were computed as the Euclidean distance between the locations obtained with the algorithm and the reference locations. The error in eye locations ranges from 0 up to 3 pixels. However, there are different algorithm parameters used for each image. Most of the eyes were located using 20 modes for the facial object. The number of RSST regions was usually set to 400. However, the lower values often give equally good results.

The graph in Figure 3.13 presents the average error of the eye localisation for

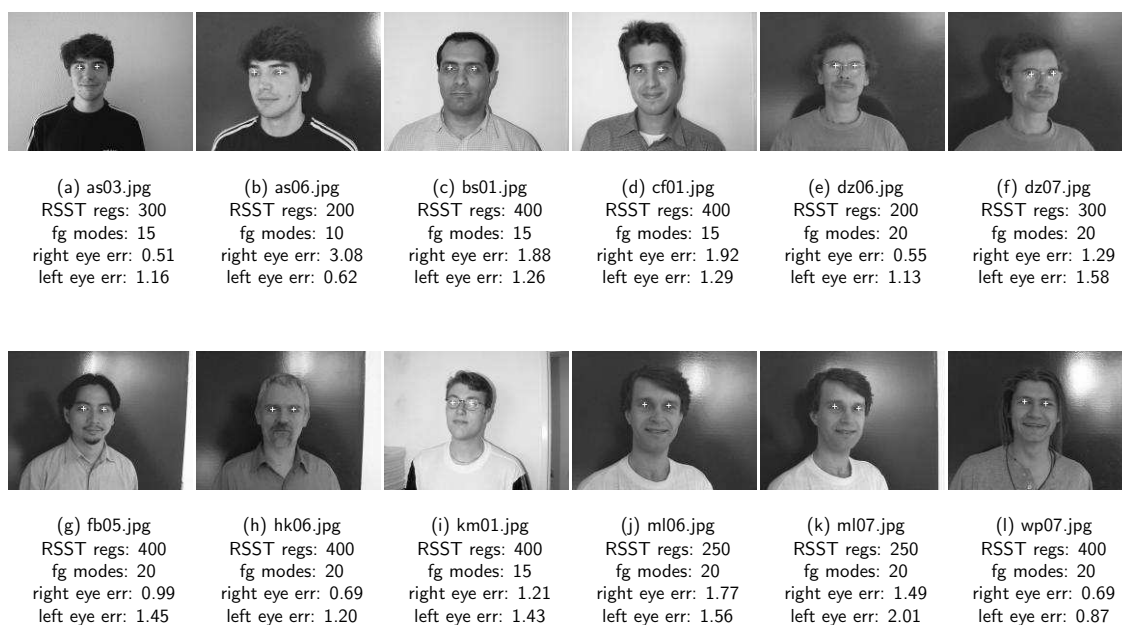


Figure 3.12: Images with lowest error rates. Numbers of RSST regions and foreground modes are given at the bottom of each image.

48 images from the HHI database. The error decreases when the number of RSST regions is increased. However, if that number rises above 400, the error decrease is not that significant and in the case of 10 modes for the facial object the error even increases. The smallest average error was obtained for 20 modes applied to the facial object. A further increase of the number of modes results in the presence of modes containing no pixels and does not improve the error rate while being computationally more expensive.

The average error rates are not small enough to use this technique to successfully locate the eyes. In order to improve the error rate some post-processing is necessary. During the post-processing, all the pixels that are outside the ellipse computed from the skin region, are removed from the facial modes. Thus all the regions of a dark background should be removed from the facial feature regions, thus improving the precision of the localisation.

Figure 3.15 presents images with the eyes located using the algorithm with the post-processing stage applied. The errors in localisation of each eye are shown as well as the parameters for extraction. The error values have decreased in the presented cases. There are less variations in the number of facial modes giving best results. However, the number of RSST regions varies substantially between images.

The graph presented in Figure 3.14 shows an average error obtained with the post-processing applied. The values of the error have decreased and are less dependent on the number of RSST regions, the graph is more even. The errors obtained with 15 and 20 facial modes are approximately equal for both

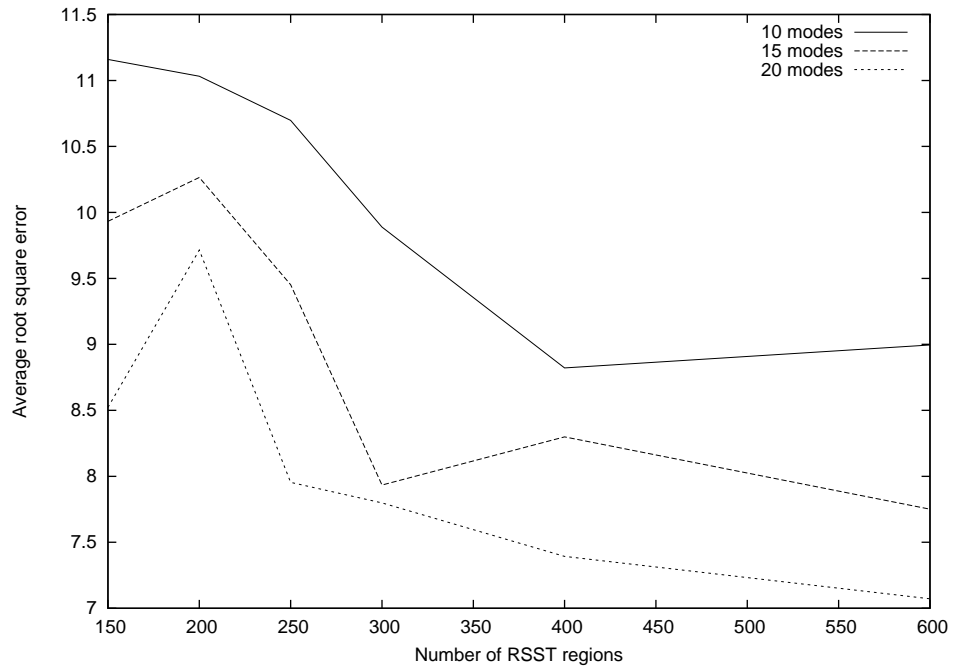


Figure 3.13: The square root error of the eye locations obtained using an approach with an automated skin colour segmentation as the function of the number of RSST regions and the number of foreground modes.

cases in the range of 200 to 250 RSST regions. The overall error rate has been decreased by applying post-processing. However, the values of average error are still unacceptably high. Although in the best cases the location errors for the eyes are acceptable.

3.3.3 Facial components extraction with the regions classification based on heuristics

The simple heuristics described in the Section 3.2.4 were applied for the classification of the modes. The heuristics determine which of the facial regions contain eye information and could be used for the location computing.

Table 3.1 presents the percentage of correct classifications. The percentages are very low. However, this statistic was created by comparing the results of heuristics with one single mode viewed as the reference. Often two modes can give good locations of the eyes, i.e. it may happen that one of the modes contains a pupil region only and another one white areas of the eyes. Thus both should be considered as correct regions.

The overall performance of the algorithm deteriorates when the heuristics are used. The graph presented in Figure 3.16 shows an average error obtained using the algorithm with the heuristics applied. These errors are unacceptably high, however, they show characteristics similar to the previously presented results. The error rates decrease when the number of modes and the number of RSST

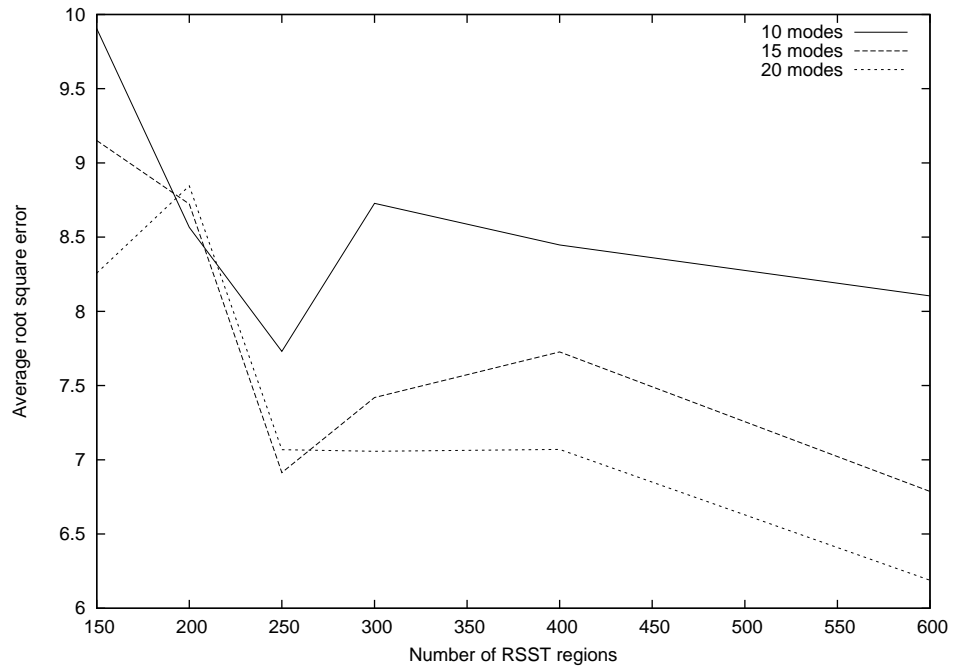


Figure 3.14: The square root error of the eye locations obtained using an improved approach with an automated skin colour segmentation as function of the number of RSST regions and the number of foreground modes.

No of modes	No of RSST regs					
	150	200	250	300	400	600
10	40	38	38	35	42	29
15	29	35	33	27	27	29
20	25	21	31	23	27	31

Table 3.1: The percentage of the correct mode classification when searching the for eyes region.

regions increase.

3.4 PCA template

3.4.1 Assumptions

The method described in previous sections extracts eye regions, which can lead to accurate eye locations, however segmentation of an image is time consuming and may yield different results depending on background complexity and other environmental conditions in a photograph. Considering that a bounding box outlining the position of a head is given, the approach to eye localisation can be simplified. The bounding box roughly surrounds both eyes and possibly eyebrows and at the bottom it is restricted by position of mouth. Sample bounding boxes of faces are presented in Figure 3.17.



Figure 3.15: The images with the lowest error rates. The numbers of RSST regions and the foreground modes are given at the bottom of each image.

It can be seen in Figure 3.17 that the size of the head region can be estimated by a size of the bounding box. Thus, having given the bounding box, only the in plane rotation of a head needs to be estimated. For non-rotated faces, eyes are centred roughly in the places shown in the template scheme presented in Figure 3.18.

3.4.2 Locating eyes

Estimation of in-plane rotation angle

The angle of in-plane rotation of the given face in an image is found by rotating the facial region at several angles and extracting the eye region with eye positions defined as in Figure 3.18. The angles are between -45° and 45° at every 5° distance, which gives 19 sampling positions. The sampling of every 5° was arbitrarily chosen as a trade-off between precision and processing time.

At every sample rotation the eye region is extracted and projected onto a PCA eigeneyes subspace. The PCA projection is described in the following section. Once the PCA projections are obtained the distance of the projections to the centre of eigeneyes subspace is calculated for every projection. Then the angle at which the projection is closest to the centre of eigeneyes subspace is regarded as the angle of in-plane rotation of the face.

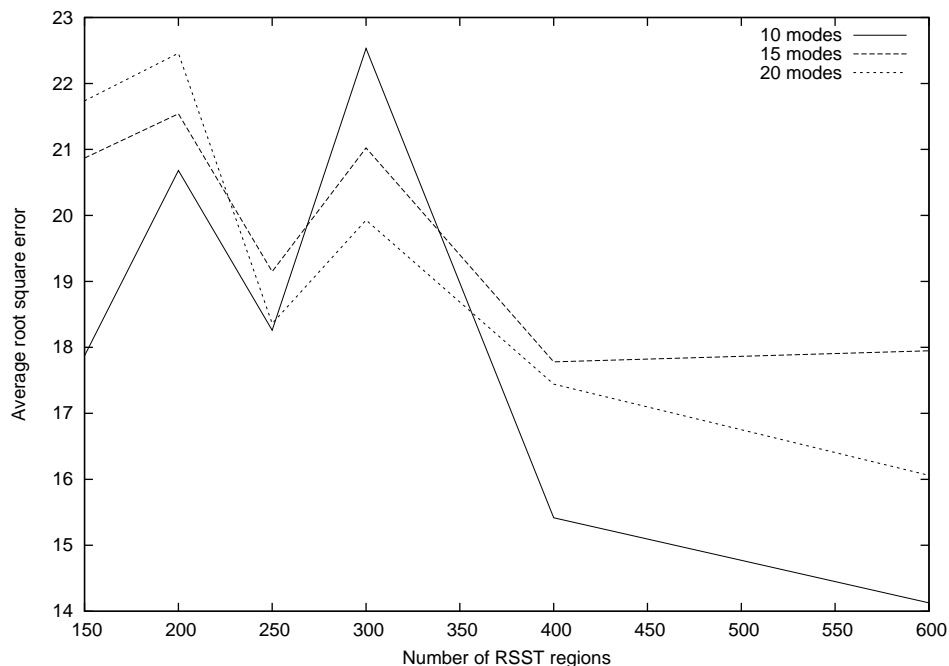


Figure 3.16: The graph of the average location error when the heuristics are applied.

PCA eye template

The decision whether the given region contains eyes is based on a PCA template. An eye region of a face is projected onto the PCA eye subspace and the distance of this projection from the centre of the subspace is calculated. The region which contains eyes would be projected closer to the subspace centre than non-eye regions.

The eye region containing both eyes is normalised before the projection to the size of 33×11 , with eyes centred in 6th row and 9th and 24th columns. This region is then converted into a vector \mathbf{x} of length 363 by row-wise scanning. For extracting the projection, firstly the mean eye region $\bar{\mathbf{x}}$ is subtracted from the eye vector and then the resulting zero-mean vector is projected onto the matrix of basis vectors $\mathbf{W} = \{\mathbf{w}_1 | \mathbf{w}_2 | \dots | \mathbf{w}_N\}^T$:

$$\mathbf{y} = \mathbf{W}(\mathbf{x} - \bar{\mathbf{x}}). \quad (3.23)$$

Pre-processing

The normalised eye region of size 33×11 consists of intensity values only. This removes dependency on colour bias, or white balance of the region. However there is still dependency on illumination values, whether the given face is in a shadow or in strong light. Therefore the eye region is pre-processed with histogram equalisation in order to bring up details and equalise illumination conditions in each region.



Figure 3.17: Sample human faces extracted from a photograph collection.

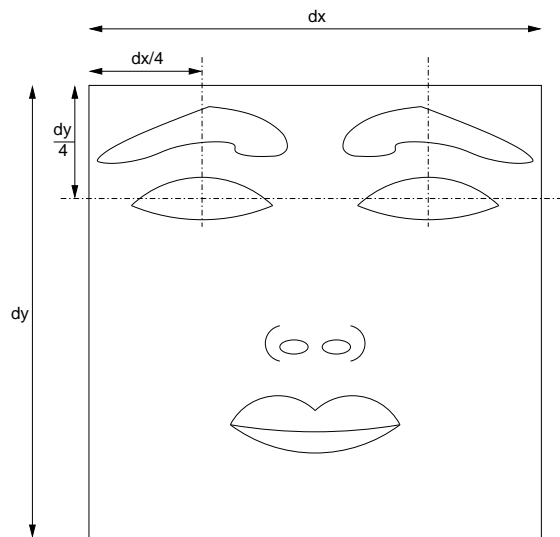


Figure 3.18: Positions of eyes in bounding box for non-rotated face.

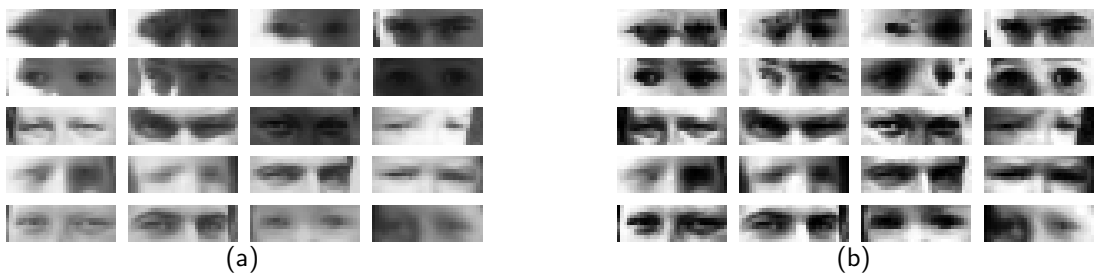


Figure 3.19: Sample eye regions, (a) before and (b) after histogram equalisation.

Algorithm

The algorithm for finding eye positions, or rather the in-plane rotation of the face within the bounding box using eye PCA templates is presented below.

Algorithm:

1. Get facial bounding box
 2. Choose rotation angle
 3. Rotate facial region
 4. Extract eye region
 5. Calculate PCA projection of eye region
 6. Calculate and store the distance of the projection to the eigeneye space
 7. if all rotation angles considered go to step 8, else go to step 2
 8. choose the angle at which the distance of the projection to the centre of eigeneye space is lowest
-

3.4.3 Results

The algorithm presented above was tested by the author using the test dataset described in Appendix C.2. As human faces in the test set are of different sizes, the error rate must be normalised in order to give meaningful values. Therefore the distances between ground truths and automatically found locations are normalised by the distance between ground truth locations of left and right eyes d_t . The mean normalised error is the sum of individual errors for left and right eyes calculated for all $K = 1127$ images in the dataset:

$$\bar{e} = \frac{1}{K} \sum_{k=1}^K \frac{\|\mathbf{p}_{ak} - \mathbf{p}_{tk}\|}{d_{tk}} \quad (3.24)$$

where \mathbf{p}_a denotes vector of coordinates of location of an eye found using the PCA template method, \mathbf{p}_t is a ground truth position of the eye and d_{tk} is the distance between both eyes in k -th facial image calculated using ground truth locations of these eyes.

The average error of 0.218 of the distance between left and right eye means that if there are faces of such a size that distance between eyes is 32 pixels, then the average error of locations from the centre of an eye would be 7 pixels. If such

Table 3.2: Error of eye localisation.

mean error left eye	0.208
mean error right eye	0.228
mean error both eyes	0.218
maximum error left eye	1.58
maximum error right eye	2.26
minimum error left eye	0.0
minimum error right eye	0.0

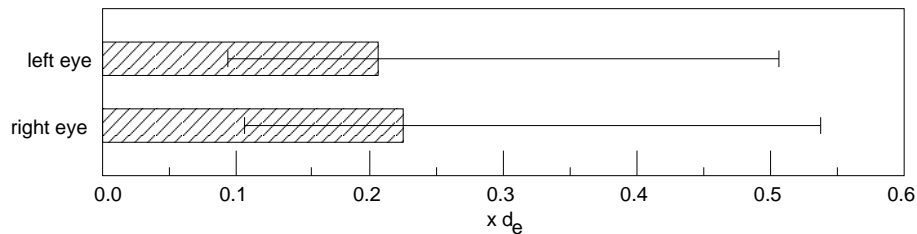


Figure 3.20: Average error of location of left and right eye normalised by the distance between eyes.

an error is in the horizontal direction, then the depicted location would still remain within the boundaries of an eye.

However, in some situations the error values become large, as the maximum error values shown in Table 3.2 and the spread of the error shown in Figure 3.20. This happens when a region that is far from the true eye region, produces the projection closest to the centre of the eigeneye space. Then the rotation angle can be very different from the true one resulting in large localisation error. This can also happen if the required in-plane rotation angle is outside the sampling range.

The technique based on PCA template is efficient. However, its accuracy depends heavily on accuracy of the bounding box. In some situations a small change in bounding box results in large errors in eye locations.

3.5 Conclusion

In this chapter techniques for localisation of facial components, with emphasis on locating eyes, are discussed. A novel localisation method based on colour segmentation of a facial image is proposed here and analysed. However this method is computationally expensive, which results in long computation times. In consumer applications the user is not eager to wait too long especially if the time exceeds the time needed for manual localisation. Therefore a quicker method based on PCA projection of candidate eye regions within a facial bounding box is analysed as a replacement. Both methods locate eyes with similar error, however the second method is faster and able to locate eye in wider range of face sizes and

in-plane rotations.

The next chapter presents face recognition techniques and examines the MPEG-7 face recognition descriptor (Appendix B) from the perspective of some enhancements to normalised facial image. The normalised facial image is based on eye positions and the influence of automated eye localisation realised with the PCA based method on face recognition results is analysed.

Chapter 4

Face recognition

4.1 Introduction

The major features used for clustering of similar persons in a photograph collection presented in Chapter 5 are face recognition features. It is essential to know what features are available and how suitable they are for unsupervised clustering. Therefore, in this chapter, an overview of face recognition techniques is presented. It is highlighted how they can be employed for unsupervised clustering of persons similar in appearance. The face recognition techniques suitable for unsupervised clustering should provide good generalisation. Also, it should be possible to obtain parameters for feature extraction (e.g. subspace basis vectors) without prior knowledge about classes (identities) in a data set. Then, the parameters for extraction can be obtained using a publicly available facial image database such as the FERET database [61] or MPEG-7 HHI face database.

Methods for locating eyes and other components of a human face are presented in the previous chapters. Accurate locations of facial components are required by many face recognition algorithms, especially the ones that are based on biometric data of a human face. However, the holistic methods also depend on the locations of eyes for normalisation of the facial image.

4.2 Face recognition techniques

4.2.1 Subspace methods

Subspace methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Independent Component Analysis (ICA) became very popular recently, due to fast and yet reliable processing. These methods analyse the structure of data and reduce the data size and redundancy. The similarity of faces is measured by computing the distance between points in multidimensional space obtained by projecting face images onto the basis vectors.

The subspace methods are characterised by a good generalisation. Additionally, such subspaces as PCA, Kernel PCA (KPCA) or ICA can be easily created using an arbitrary training set and another data set can be easily projected onto these subspaces. Similarities between data points from a projected data set can be easily found, despite the fact that a given subspace is not optimised for a given data set. An exception is the LDA method, which is optimised for a given training set, thus generalisation is expected to be poorer than of other subspace methods.

Principal Component Analysis

The PCA and ICA methods are presented in Chapter 2 in context of facial component extraction. Here their application to face recognition is discussed.

The PCA method is based on eigenvector decomposition of data and was first proposed for face representation by Sirovitch and Kirby [62] and used for recognition by Turk and Pentland [63]. The method provides orthogonal vectors with the principal direction pointing at the direction of the highest variance of the data. As eigenvectors of face images are face-like in appearance, they are called eigenfaces. This method gives good results in controlled environment with little changes to view points, i.e. usually is used for frontal views of faces. Many other techniques perform PCA decomposition as a pre-processing step for reducing data dimensionality.

Independent Component Analysis

ICA introduces a stronger condition than PCA as it assumes statistical mutual independency of basis vectors or input data, in other words that the random vectors are drawn from two or more statistically independent random sources. ICA reduces not only first and second order statistics as PCA does, but also reduces to zero higher order statistics. In that way more redundancy in the data can be removed and the representation can be more compact. Bartlett *et al* [64] employed ICA for face recognition in two scenarios: the first having independent basis vectors, and the second for independent coefficients used for image representations. She reports similar results for both scenarios with the latter giving better results and both outperforming the PCA method. However, the comparison of PCA and ICA given in [65] finds no difference in performance between PCA and ICA.

Linear Discriminant Analysis

The Linear Discriminant Analysis (LDA) method is based on Fisher's Linear Discriminant (FLD). This method searches for a subspace where the ratio of between-class scatter matrix and within-class scatter matrix is maximised. Thus the projection of the data is optimal for linear separation of classes. Kucharski *et al* [66]

project face vectors onto the space of LDA basis vectors for matching face images. Gross *et al* [23] report improved performance when using LDA in comparison to PCA.

LDA maximises the ratio [66]:

$$\left| \frac{\Phi^T S_b \Phi}{\Phi^T S_w \Phi} \right| \quad (4.1)$$

where S_b is a between-class scatter matrix:

$$S_b = \sum_{m=1}^M N_m (\bar{x}_m - \bar{x})(\bar{x}_m - \bar{x}) \quad (4.2)$$

and S_w denotes within-class scatter matrix:

$$S_w = \sum_{m=1}^M \sum_{i=1}^{N_m} (x_i - \bar{x}_m)(x_i - \bar{x}_m) \quad (4.3)$$

where M denotes the number of classes and N_m is the number of instances in m th class. By maximising the expression in Equation 4.1 LDA searches for the subspace in which classes are most separable [66]. The dimension of such a subspace is at most $M - 1$.

Locality preserving subspace

Recent approaches exploit nonlinear characteristics of the appearance of human faces in digital images. The next section presents kernel methods which are one way for utilising nonlinearities in facial appearances. Another way is trying to find the nonlinear submanifold on which facial images might reside. One of the latter methods for representing human faces is locality-preserving subspace (LPS) based on locality-preserving projections (LPP) [67], called Laplacianfaces [68]. Other methods for nonlinear subspaces are also available such as Isomaps [69], local linear embedding (LLE) [70] and Laplacian eigenmaps [71]. However, these methods, define maps only for training data [68], thus they have problems dealing with new classes. They are not easy to use in unsupervised clustering, unless only a closed-world scenario is considered.

The locality-preserving projection preserves the local structure of the data, unlike PCA which preserves the global structure. In the case of nearest neighbour classification, which is most commonly used in face recognition, the local structure is more important than the global structure [68]. Additionally the locality-preserving projection produces a nonlinear manifold for the construction of the locality-preserving subspace, which gives superiority over linear PCA. The manifold is constructed as an adjacency graph [68].

Kernel methods

Techniques presented in previous sections provide dimensionality reduction of the data by trying to find any linear structure within the data. A different approach is proposed in kernel methods, where firstly additional dimensions are created with a kernel function and then dimensionality reduction is carried out using PCA or LDA. Kernels are usually made of nonlinear functions that project n -dimensional data onto a f -dimensional space, $f \gg n$. Because kernel functions are nonlinear, kernel methods are often referred as nonlinear subspace methods. The Gaussian kernel is used in most applications due to Gaussian characteristics of data. However, other mappings can be used such as polynomial or sigmoidal functions. The kernel subspace methods such as Kernel PCA (KPCA) or Kernel LDA (KLDA) can be viewed as generalisations of respectively PCA and LDA, as the linear subspace can be easily obtained with kernel methods by using polynomial kernels of first order [72].

Shakhnarovich *et al* [65] show that the kernel method outperforms PCA and ICA methods. Experiments presented in [72] confirm these findings. However, the dimensionality of subspaces created with kernel methods is higher than their linear counterparts.

View-invariant subspace methods

Pentland *et al* [73] propose a view-invariant face recognition algorithm in subspaces created for each view angle separately. For M individuals, each captured from C view points, C eigenspaces are calculated. The first step in this approach is to determine the pose of the head. This is done by selecting the eigenspace which describes the facial image in the best way e.g. by examining the likelihood estimate [73].

Another approach to the view-invariant eigenspace is presented in [74]. A single eigenspace is extracted from all MC images. This eigenspace describes both identity and pose of the face in an input image. Matching in such a eigenspace is based on the observation that the points representing the images of a given individual can be linked, creating a path (eigensignature) in the eigenspace (see Figure 4.1). The eigensignatures are located in different places for different individuals. However, the shapes of eigensignature curves are similar. Therefore, there exists one general shape of a facial eigensignature that can be used for estimating a specific eigensignature based on the projection of a single facial image [74]. The eigensignatures are generated using the Radial Basis Function Network (see Section 2.2.5) and the matching process is carried out by matching a facial image to a virtual eigensignature using the nearest neighbour classifier with the Euclidean distance.

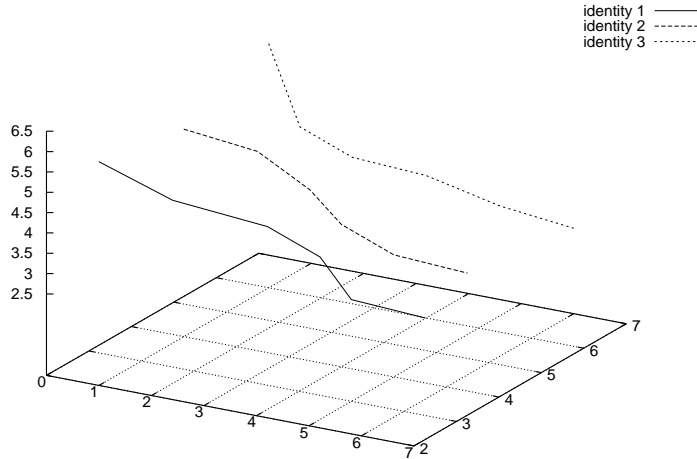


Figure 4.1: Eigensignatures of three identities in a three-dimensional eigenspace.

4.2.2 Probabilistic approaches

The probabilistic approach (similarly to LDA) defines two classes of variations of facial images: intrapersonal and extrapersonal [75] and (unlike LDA) models them as probability distributions. The former one corresponds to changes in face appearance of the same person, due to e.g. changes in expression, illumination or pose. The latter one corresponds to differences in face appearance between various people. The distance measure $S(\Omega)$ is calculated as the intrapersonal posterior probability $P(\Omega_I|\Delta)$ that the given facial image Δ belongs to a certain class:

$$S(\Delta) = P(\Omega_I|\Delta) = \frac{P(\Delta|\Omega_I)P(\Omega_I)}{P(\Delta|\Omega_I)P(\Omega_I) + P(\Delta|\Omega_E)P(\Omega_E)}, \quad (4.4)$$

where Ω_I denotes intrapersonal variations and Ω_E denotes extrapersonal variations. The probabilities are modelled as multivariate Gaussians within the PCA representation of parameter space, which reduces the number of parameters. It also allows such modelling in environments with the limited number of data. The face that is matched is projected onto two subspaces: the first one consists of eigenvalues of intrapersonal Gaussians, the second one contains eigenvalues of extrapersonal Gaussians. These projections are used for calculating the posterior probability with the Bayesian rule. The resulting probability is seen as a similarity measure and can be used for simple Euclidean distance matching, or for calculating Maximum Likelihood measure. This method outperforms the techniques presented previously, according to experiments carried out by Shakhnarovich *et al* [65].

It is not clear, however, how the Bayesian method could be used for unsupervised clustering. The Bayesian method as proposed in [65] requires training

of each class, as intrapersonal and extrapersonal probabilities must be computed. Therefore, when a new class is to be added, these probabilities must be recalculated. The probabilities cannot be obtained from a training set that would contain classes different than the data set. This method could be used for unsupervised clustering as a kind of post-processing or refining tool. However, there is a danger of carrying the erroneous classification from the initial clustering.

Another attempt to exploit the probabilistic framework for face recognition is the use of Hidden Markov Model (HMM). The HMM is a statistical model for representing an unobservable Markov chain of finite number of states. The probabilities of states are approximated with density functions, which are estimated from data. The HMM can be trained using a Viterbi algorithm, which is a special case of the EM algorithm (for information on the EM algorithm see Section 3.2.4).

A 1-dimensional HMM for recognition is proposed in [4]. Five states of HMM are defined, each representing different parts of a human face: hair, forehead, eyes, nose and mouth. The face is analysed in the same order i.e. from the top to the bottom. The HMM is fed with 2D DCT (2-dimensional discrete cosine transform) vectors extracted from each part of the face (regions covered by these parts might overlap). Figure 4.2 presents the states of the HMM representing a human face.

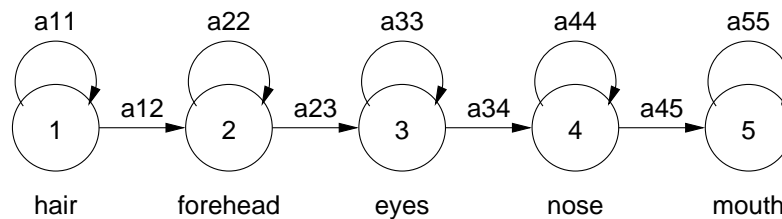


Figure 4.2: 1D HMM states for face representation [4] (a_{ij} are coefficients of the state transition matrix \mathbf{A}).

As HMM is built and trained for each class, it is not easy to adopt this method for unsupervised clustering. Therefore, it is probably best to use it as a second stage, in two-stage clustering, for refining initial clustering results. Then, the initial results might be used for training the HMM and if no new identity is expected to occur in new data, these HMMs can be used.

4.2.3 Biometric-based methods

The biometric-based methods exploit the structure of a human face. Early approaches, such as the pioneering system of Kanade [76], use just geometric information. Modern techniques utilise graph matching methods.

Elastic Bunch Graph Matching (EBGM) is a modern method based on graph matching and wavelet theory. It is an extension of a Dynamic Link Architecture (DLA) [77]. The Dynamic Link Architecture consists of a rectangular graph with

*jets*¹ in nodes. The local wavelet coefficients of jets are invariant to changes in lighting, rotation, scaling and translation.

The EBGGM extends the dynamic link architecture by using an elastic graph with jets placed in fiducial points² on a facial image. Additionally, in order to speed up calculations, each node contains several jets. Then the architecture is called the *bunch graph representation* [78] and the full graph representing a human face is named the *face bunch graph* (FBG) [77]. This provides better generalisation of the model, and therefore, an unknown face can be modelled more accurately.

For the recognition process the pose of a person's head is estimated firstly using graphs adapted to different poses. These graphs are learnt on a training set. Once the pose is estimated an elastic FBG is matched to the face — the fiducial points are found and jets placed onto them. Then, the similarity between faces is found by means of similarity matching between the FBGs of the faces in the same pose.

The biometric methods depend heavily on accurate locations of fiducial points. Therefore, they require facial images with a certain minimal size (say 128x128 pixels [78]), so that details of facial components are well visible. This restricts the use of biometric methods for analysing a photograph collection as in many cases the facial area is smaller than required. If images have sufficient details, the EBGGM is suitable for unsupervised clustering.

4.2.4 3-dimensional techniques

The methods and techniques presented above utilise only 2D information. Those methods have difficulties matching faces at various view-points, as differences in appearance between various views of the same face are larger than differences between two various faces at the same view [74]. A couple of approaches have been proposed for face recognition using 3D techniques in order to solve the problem of multi-view recognition.

A 3D model of a human head provides enough information to produce 2D representations of the face from unknown view-points. A representation of a face surface based on polyhedral approximation and multidimensional scaling (MDS) is proposed in [79]. This way a representation that is invariant to isometric deformations is obtained. This representation allows the mapping of 2D facial texture images into 3D facial geometry images. These are then decomposed with PCA.

A morphable 3D model for face recognition is proposed in [80]. Common computer graphics algorithms are used for creating a 3D model, and estimating the 3D shape and textures by fitting the morphable 3D model to the facial image. Faces are represented as parameters of 3D shape and texture. Good results were

¹Jets are Gabor wavelet coefficients extracted locally at points of graphs nodes placed on an image

²Fiducial points are points on a human face which strongly determine shape of a face and facial components such as e.g. corners of lips or eyes, high curvature points on the outline of nose, etc.

obtained in experiments carried out on FERET and CMU_PIE databases. However, they are similar to the results of the Bayesian approach [80].

4.2.5 Other techniques

There are many more face recognition techniques developed over the 30 years of research on computer face recognition. The methods presented above are the most popular and best known.

The subspace methods such as PCA, ICA and LDA presented above, are holistic methods, i.e. they analyse a human face as a one region. There are variations of each of these subspace methods that work with facial components instead of the full face. For example, the authors of [73] create PCA projections of facial components named eigenfeatures. Eigenfeatures give higher recognition rate than eigenfaces for low dimensionality, and a small increase in recognition rates when combined with eigenfaces.

The combination of different techniques leads to hybrid methods. It is proposed in [81] that PCA should be combined with local feature analysis (LFA). They argue that PCA should be used for estimating eigenmodes of the highest corresponding eigenvalues, while the LFA for other (higher-order) eigenmodes. The reason for this is that the PCA eigenmodes represent the global structure of a facial image and the first eigenmodes correspond to low frequency images. On the other hand, the eigenmodes of lower eigenvalues correspond to higher frequency images, which are better estimated with the LFA since the LFA preserves local topological information.

The algorithms of artificial intelligence (AI) such as neural networks or genetic algorithms are very often used for classification and combined with other face recognition methods. The Evolution Pursuit (EP), which is a type of a genetic algorithm, is used for face recognition in [82]. This technique is based on eigenspaces and Evolution Pursuit is used for searching for basis vectors, which minimise a fitness function. Support vector machines (SVMs) are also very often used for classification. The SVM is a two class classifier, therefore, several SVMs must be combined for recognising several identities (there are as many SVMs required as there are identities) [83].

4.3 Recognition with MPEG-7 descriptor

4.3.1 MPEG-7 Face Recognition

The MPEG-7 Face Recognition descriptor (MPEG-7 FRD) is a visual descriptor defined in the MPEG-7 standard [84] that aims at providing a face recognition functionality for content based image retrieval (CBIR). This descriptor employs a

holistic linear subspace method for face recognition. The mean human face and 48 basis vectors are defined by the standard. Also, the normalised facial image containing only illumination values is defined. The MPEG-7 FRD features are extracted by projecting the normalised facial image onto the subspace defined by the basis vectors. The final step of extraction is the normalisation of features. The MPEG-7 standard also proposes the weighted L_1 norm as a similarity measure for matching faces [24]. Detailed information on the MPEG-7 FRD is presented in Appendix B.

The MPEG-7 FRD is well suited for unsupervised clustering of similar faces. Because the basis vectors are already defined, there is no need for a training set. There is also some flexibility in choosing a distance measure, as the one proposed by the standard [24] can be replaced by Euclidean, cosine or any other suitable distance measure. The descriptor was designed with high generalisation ability in mind as its primary usage is matching faces of individuals that were not captured in the training set. Similarly to other holistic subspace methods, MPEG-7 FRD depends heavily on the accuracy of creation of a normalised facial image. This image is defined using locations of eyes and hence the performance of the descriptor depends heavily on the accuracy of the eye localisation process [85].

4.3.2 Pre-processing

The MPEG-7 standard does not propose any pre-processing of the facial image other than normalisation. It is prone to strong lighting changes and variations of background, as the area covered by the normalised facial image contains some background around the human face.

Histogram equalisation is carried out in this section on the normalised facial image, in order to remove lighting variations. The equalisation is not capable of removing strong shadows. This is especially true when the direction of strong light changes significantly. However, in most images, the appearance of faces of the same individual is equalised to some extent. Additionally, the histogram equalisation greatly enhances the contrast of the normalised facial image, which helps in outlining facial features. Figure 4.3 presents some examples of normalised facial images before (a) and after (b) histogram equalisation.

The best solution for removing the background influence on recognition is extracting a face using a colour segmentation of the normalised facial image. However, this could be time consuming and not possible when working on intensity values only.

So-called *feathering* is used here in order to simplify the background removal. When an image is processed by feathering the values of intensity are multiplied by a 2-dimensional Gaussian function centred over the object (in this case a human face). As a human face is expected to be in the centre of the normalised facial



Figure 4.3: Examples of normalised facial image; (a) before any pre-processing, (b) after colour histogram equalisation, (c) after colour histogram equalisation and feathering with $\sigma = 0.2$.

image the Gaussian is centred in the middle of the image. If intensity values in an image are in range $0 - 255$, feathering can be denoted with the Equation 4.5:

$$I_f(x,y) = G(x,y,\sigma) * (I(x,y) - 127) + 127, \quad (4.5)$$

where

$$G = \exp\left(-\frac{(x-x_c)^2}{(w\sigma)^2} - \frac{(y-y_c)^2}{(h\sigma)^2}\right)$$

and $I(x,y)$ is the intensity value at the point (x,y) , $I_f(x,y)$ is the intensity value after feathering, x_c and y_c denote the coordinates of the centre of an image, and w and h are respectively the width and height of an image. The value of σ determines the width of the Gaussian.

Figure 4.3(c) presents sample images after feathering. For comparison, they are shown alongside images that were not pre-processed in any way and images processed with histogram equalisation.

4.3.3 Experiments

Some experiments were carried out by the author for assessment of the influence of pre-processing on recognition with MPEG-7 FRD. Two data sets were used for assessment: a small dataset containing photograph from one personal collection and a set containing 1127 facial images from several collections. For details on the data sets see Appendix C.2.

The experiments were carried out in 12 scenarios summarised in Table 4.1. For each data set, MPEG-7 FRD features were extracted using facial images normalised using manually marked locations of eyes and, for comparison, locations automatically found with the PCA technique described in Section 3.4.

The experiments were carried out by matching each of the faces in the database to the other faces using the nearest neighbour classifier. The nearest neighbour

Table 4.1: Scenarios for experiments with recognition.

experiment	scenario
1	small data set, manual eye locations, no preprocessing
2	small data set, manual eye locations, histogram equalisation
3	small data set, manual eye locations, histogram equalisation + feathering
4	small data set, auto eye locations, no preprocessing
5	small data set, auto eye locations, histogram equalisation
6	small data set, auto eye locations, histogram equalisation + feathering
7	large data set, manual eye locations, no preprocessing
8	large data set, manual eye locations, histogram equalisation
9	large data set, manual eye locations, histogram equalisation + feathering
10	large data set, auto eye locations, no preprocessing
11	large data set, auto eye locations, histogram equalisation
12	large data set, auto eye locations, histogram equalisation + feathering

classifier used in these experiments is based on the distance measure defined for the FR descriptor in the MPEG-7 standard [24]. If the label of the queried face is the same as the matched nearest neighbour, then it is said that this face was recognised correctly, otherwise it is said that the face was not recognised. Correctly recognised faces are counted and their number is divided by the number of all faces in the database giving the recognition ratio.

As it can be seen in Figure 4.4, where recognition rates for scenarios from Table 4.1 are shown, the pre-processing does indeed improve recognition rates in most scenarios. It is not the case for a small data set with automatically located eyes. The reason for this can be the data set itself, as it is small and a few errors in eye localisation can cause significant errors in recognition. As feathering removes information placed near the edges of an image, any misplacement of a face outside the centre of an image reduces the number of facial details, thus reducing the recognition effectiveness.

An important observation must be made on the relation between the size of the data set and the recognition ratio. For a small data set, recognition ratios are higher than for the larger data set, even in the case of automated eye localisation. This results from higher variations of faces in the larger set and the higher number of outliers, which can be seen as a noise, and which affect the recognition.

4.4 Conclusion

In this chapter face recognition algorithms are presented and an analysis of how they can be used for unsupervised clustering is carried out. The subspace methods are best suited for unsupervised clustering as it is easy to extract recognition features from new images and new classes. Additionally such methods as PCA are characterised by good generalisation. Therefore, the basis vectors can be created using a training set containing none of the classes that can be found in the data

scenario	1	2	3	4	5	6	7	8	9	10	11	12
recognition [%]	56.4	54.5	60.0	43.6	41.8	40.0	30.8	33.3	37.2	21.3	23.4	24.5

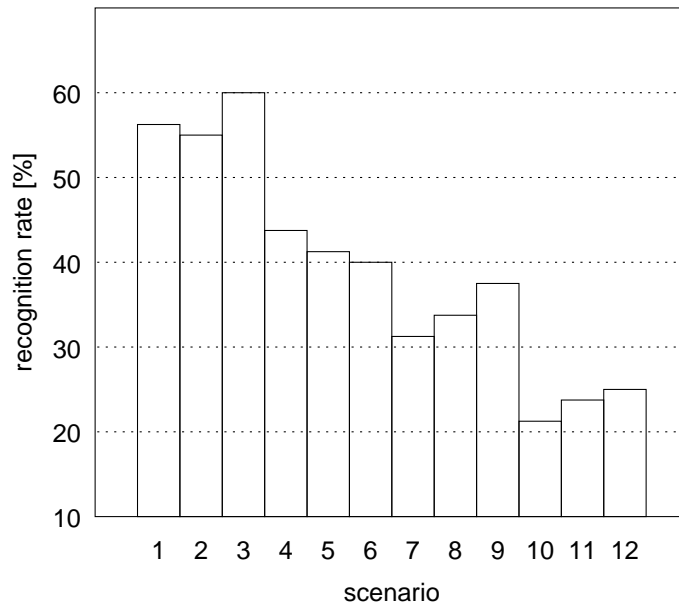


Figure 4.4: Recognition rates for different scenarios.

set used for recognition.

Histogram equalisation and feathering are proposed in this chapter for pre-processing of the normalised facial image before the MPEG-7 FR features are extracted. This pre-processing enhances the recognition effectiveness. The recognition ratios are much lower than ratios published in other works, but the experiments presented in this chapter were conducted using a database of photographs captured in an unconstrained environment. The recognition is less effective in such images than in images obtained in constrained conditions in a laboratory.

In the next chapter, the MPEG-7 FR features obtained with pre-processed normalised facial images are used for grouping photographs into clusters consisting of images of persons similar in appearance. Ideally, a cluster should contain photographs of only one individual. Therefore, effectiveness of grouping is measured by means of precision and recall of photographs of the same individual.

Chapter 5

Clustering

5.1 Introduction

In the previous chapter, features for face recognition are examined. This is the first step of a clustering process. The second step, a choice of similarity measure, is indirectly analysed in the same chapter, as similarity measure is closely related to features. In this chapter, grouping algorithms are analysed — grouping is the third step of unsupervised clustering. Some experiments with a modified k-means algorithm, a single-link algorithm and an algorithm based on the nearest neighbour classifier are described. The results of these experiments are also included and analysed. The evaluation of the algorithms presented in this chapter realises the fifth step of the clustering algorithm (see Section 1.2).

Unsupervised grouping is not a trivial task, as many parameters need to be found or estimated, based on limited knowledge of the data. It is important to distinguish between supervised clustering (supervised classification, discriminant analysis) and unsupervised clustering [22]. In unsupervised clustering, parameters of each cluster such as the shape and area have to be estimated, appropriate points classified to the clusters and most importantly the number of clusters needs to be estimated. The latter task is not trivial and usually is based on some kind of thresholding, or a pre-defined number of clusters is used. Supervised clustering usually estimates some parameters of a given number of known classes based on labelled training data and then classifies unknown data points to those classes.

Noise, in terms of outliers, is a difficult problem for clustering. Outliers change the parameters of the clusters. This can result in erroneous classifications. Therefore, there is a need for some mechanisms for outlier detection and removal (or any other means of dealing with them, e.g. placing outliers in a separate cluster dedicated to outliers).

The well known clustering algorithms like k-means [22], the Expectation-Maximisation algorithm [86] or N-cuts [87] require a pre-defined number of clusters (classes). The cumulative agglomeration (CA) algorithm [88] and its generalisations [89, 90] deal

with the problem of the number of clusters by discarding all clusters with cardinalities lower than the predefined number. This is effectively a threshold applied to the minimal number of points within each cluster.

Some known clustering algorithms were modified by the author in order to estimate the number of clusters represented by the analysed data set. These modifications are presented in this chapter. The modifications based on outliers detection are applied to the k-means and single-link algorithms. Those algorithms are employed by the author for unsupervised grouping of photographs containing people who are similar in appearance. The author also uses the algorithm based on the nearest neighbour classifier. This algorithm is very convenient, because it does not require any thresholding for estimating the number of clusters.

5.2 Known techniques

Clustering algorithms can be divided into the following classes [22]:

- hierarchical
 - single link
 - complete link
- partitional
 - square error
 - graph theoretic
 - mixture modelling
- fuzzy
- artificial intelligence
 - neural networks
 - evolutionary algorithms
 - simulated annealing

5.2.1 Hierarchical algorithms

Hierarchical algorithms are based on grouping the data set into hierarchical structure, usually a tree [22]. Each of the levels of the tree gives a partitioning of the data at a certain coarseness or level of detail (LOD). Closer to the root of the tree fewer clusters are observed and fewer details of the data are obtainable.

Hierarchical algorithms can be either agglomerative or divisive. However, the former one is more popular. In an agglomerative algorithm, each data point firstly

forms a separate cluster. Then, the ordered list of links between clusters is found. The order is usually based on a distance between clusters. Clusters linked with the minimal distance are merged and distances between clusters are updated. This procedure is repeated until there is only one cluster left. The required partition can be found by following the hierarchy.

The divisive algorithm splits clusters rather than merges them. The hierarchical divisive algorithm begins with a single cluster consisting of all data points and splits the clusters until the number of clusters is the same as number of data points, with each cluster consisting of single data point.

A difference between different agglomerative hierarchical algorithms is in the distance measure and the method of updating cluster parameters. Single-link algorithms [22] use the distance between the two points from different clusters that are closest to each other i.e. it is the minimum of all the distances between points of the two clusters. The complete-link algorithm [22] uses the maximum distance between points of two clusters as the distance measure between the two clusters. As a result complete-link algorithms produce more compact clusters, however, they are not able to model some complex structures of clusters that can be modelled by single-link algorithms. On the other hand, there is a danger of the chaining effect in the single-link algorithm [22].

Recent hierarchical clustering methods include Recursive Shortest Spanning Tree (RSST) and Binary Partition Tree (BPT) [53]. The BPT is a convenient representation of the data, most efficiently obtained by storing each level of RSST as a binary tree. The RSST is a recursive and iterative version of the Shortest Spanning Tree algorithm [57].

5.2.2 Partitional algorithms

Partitional algorithms aim at producing a single, optimal partition rather than a hierarchy of partitions. They usually require some initial partition (even a random one) to begin with and a stopping criterion for finishing the clustering process.

Square error

The family of k-means algorithms is best known for partitional clustering. The k-means and similar algorithms are based on minimisation of the squared error, where the *error* is a distance between points in a cluster and its centroid. The algorithm starts with an initial partition and reassigns data points to clusters of most similar centroids. Then, the centroids are recalculated and data points are again reassigned. This process is ceased when a stopping criterion is met. The process can be stopped, for example, when none of the data points changes its membership or a very small number of data points do so. A detailed analysis

of the variation of the k-means algorithm for unsupervised clustering introduced here is presented in Section 5.3.1.

The shape of the clusters depends on the distance measure. Typically, the Euclidean distance is used. This results in the creation of hyperspherical clusters. Therefore, the best results are obtained for data containing hyperspherical classes. If hyperellipsoidal clusters are expected, the Mahalanobis¹ distance should work best. The Mahalanobis distance also gives good results in situations where features are correlated [22].

Among the variations of k-means algorithms described in literature are the Generalised Lloyd Algorithm (GLA) [91] and the ISODATA algorithm [92]. The GLA aims at the choice of initial partition by starting with a single cluster containing all data points, and iteratively splitting clusters until a convergence is met. At each iteration clusters are formed using the k-means algorithm. The ISODATA algorithm allows both splitting and merging clusters based on the variance of the data within each cluster [22]. The number of clusters is estimated by discarding clusters with cardinalities below a pre-defined threshold.

Graph theoretic clustering

In the clustering algorithms based on graph theory, data is viewed as points on a connected graph. An example of the graph theory based algorithm is the minimal spanning tree (MST). It is a divisive algorithm based on splitting clusters at the links of the highest distance (cost) [93, 94].

The family of hierarchical clustering algorithms is directly related to graph-based algorithms as the hierarchy tree is a form of a graph. The Recursive Shortest Spanning Tree (RSST) algorithm is an example of a hierarchical agglomerative algorithm [57]. It is successfully used for image segmentation (see Section 3.2.2), in which case clusters consist of pixels belonging to the same region. The RSST algorithm starts with each data point in a separate cluster and recursively merges the neighbouring clusters, i.e. the ones with the lowest merging cost (Section 3.2.2).

Another popular algorithm based on graph theory is the minimum cut (M-cut) presented in [95]. This divisive algorithm partitions the data set in order to minimise the maximum cut between segments.

If one considers a graph V consisting of two disjoint subgraphs A and B , $A \cup B = V$, $A \cap B = \emptyset$, with data points $u \in A$ and $v \in B$, the cut is defined as:

$$cut(A, B) = \sum_{\forall u \in A, \forall v \in B} w(u, v), \quad (5.1)$$

where $w(u, v)$ denotes the cost of the edge (link) between points u and v .

¹Mahalanobis distance is a distance between a point described by the vector $\mathbf{x} = \{x_1, \dots, x_N\}^T$ and a cluster C described by its centre of gravity $\mathbf{m} = \{m_1, \dots, m_N\}^T$ and the covariance matrix Θ (of size $N \times N$). The distance is expressed as $d_{Mahalanobis}(\mathbf{x}, C) = \sqrt{(\mathbf{x} - \mathbf{m})^T \Theta^{-1} (\mathbf{x} - \mathbf{m})}$.

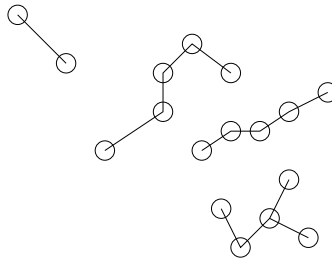


Figure 5.1: Sample clusters created with the nearest neighbour technique.

The iterative algorithm for finding partitions by using the minimum cut criterion starts with a single cluster — a graph spanning all data points. Then, the data is recursively bisected at the point of the minimum cut. This process stops when a predefined number of partitions is reached [95].

Mixture modelling

The mixture modelling algorithms are based on the observation that clusters can be described by a probabilistic distribution. The Expectation-Maximisation (EM) algorithm [86, 96] is the most common and the most efficient among the mixture modelling algorithms. The EM algorithm estimates parameters of distributions based on maximising expectation of the likelihood function in regard to parameters of distributions. More details on the EM algorithm are available in Section 3.2.4. The multivariate multimodal Gaussian distribution is the most common representation of clusters.

Nearest neighbour

Nearest neighbour clustering is a convenient and simple algorithm based on the proximity of points in the data set. Clusters are created by linking a given point with the point that is closest in terms of a given distance measure. This technique is very simple, yet very convenient as it does not require the number of clusters to be given in advance. Additionally the shape of clusters is not determined by any model of clusters, thus this technique can produce the clusters of any shape. The distance measure can be thresholded — if the distance to the nearest neighbour is higher than a threshold the current point is not linked and is regarded as an outlier.

There is a variant of the nearest neighbour technique used in supervised clustering — the k -nearest neighbours (k -NN). This algorithm is similar to the voting method in information fusion (see Chapter 6). Here, k points from the labelled data, that are nearest to the data point being classified, “vote”. The label that receives the highest number of votes is assigned to the analysed data point [97].

5.2.3 Fuzzy algorithms

Most of the algorithms presented above provide *hard* partitions, i.e. they classify a given data point to one and only one cluster. *Soft* or *fuzzy* clustering allows a degree of membership to each cluster to be assigned to every data point. For example, if there are 3 clusters, the membership of a data point can be described by three values: 0.6,0.1,0.3. Numbers in this example mean that the given point belongs to the first cluster with a degree of membership 0.6, to the second cluster with the membership degree 0.1 and to the third cluster with the degree 0.3. The *hard* partition can be obtained from a *soft* one by assigning a data point to the cluster with the highest value of the membership degree [98], although probabilistic assignment based on the degree is also possible.

The best known soft clustering algorithm is the fuzzy c-mean (FCM) algorithm [99]. This algorithm is based on minimisation of the objective function, which incorporates the membership function:

$$J_{FCM} = \sum_{n=1}^N \sum_{c=1}^C u_{nc} d^2(\mathbf{x}_n, \mathbf{x}_c) \quad (5.2)$$

where N is the number of data points, C denotes the number of clusters, u_{nc} is the value of the membership function of the n th point belonging to the c th cluster and $d^2(\mathbf{x}_n, \mathbf{x}_c)$ denotes the distance between the features of the n th data point \mathbf{x}_n and a representation of the c th cluster with its centre $\mathbf{x}_c = \sum_{n=1}^N u_{nc} \mathbf{x}_n$. This method requires an explicitly specified number of clusters C .

The EM algorithm described in Section 3.2.4 also provides fuzzy partitioning and can be viewed as a combination of fuzzy clustering with mixture modelling techniques. The posterior probability in the EM model is the membership function, which is a basis for soft clustering. However, instead of minimising the objective function, the estimation of the likelihood function is maximised.

5.2.4 Artificial intelligence

Neural networks

Artificial neural networks (ANNs) are inspired by the structure of the human brain and nervous system. General information on neural networks, how they are constructed and how they work can be found in [100]. They were and still are extensively used for classification, because they usually provide good generalisation.

For unsupervised clustering, the ANN was used in [101] as the basis of learning vector quantisation (LVQ) and a self organising map (SOM). The competitive, winner-take-all ANNs can be also used for clustering [22].

The architectures of ANNs for clustering are simple. They are usually single layer networks, therefore, they do not require any sophisticated learning algo-

rithms such as the back propagation algorithm. Some relationships can be found between ANNs for clustering and classical algorithms, e.g. the LVQ algorithm can be related to the k-means algorithm [102]. Outputs of an ANN can be regarded as probabilities, thus exhibiting similarities to mixture modelling methods.

Self organising maps became very popular and are used successfully in speech recognition. They produce a two-dimensional map, which is a visually convenient representation of a multidimensional data set. However, the results heavily depend on the initialisation of weights. An essential feature of a self organising map is that it preserves neighbourhood relationships between points.

All ANN algorithms require a predefined number of clusters, as the number of clusters is limited by the number of outputs of the net.

Evolutionary algorithms

Evolutionary algorithms (EAs) are based on the concepts of natural evolution. They seek for an optimal solution in a parallel way, modelling the solutions as chromosomes and applying genetic operators to modify them. The most known algorithms of this type are genetic algorithms (GA). The idea of a GA is presented in Section 2.2.6.

Evolutionary algorithms are used for clustering as solutions to the problem of minimising of the square error. What distinguishes GAs from other clustering algorithms is that GAs search for the global solutions, unlike other algorithms that utilise local information. Using such operations as *crossover* or *mutation* genetic algorithms are capable of finding new solutions that are very different from the existing ones.

For GA clustering, the representation of a data set proposed in [104] is a K -nary string of length N , where K denotes the number of clusters and N is the number of data points. However, this representation is not free of problems. For K clusters there are $K!$ chromosomes representing each partition of the data, which increases the search space by $K!$. Additionally there is a risk associated with the crossover operator, whereby using this operator can result in creating inferior clusters.

Various modifications have been proposed to solve these problems [22]. It is proposed in [105] to add a separator symbol to denote the border between partitions. In that way the clustering becomes a permutation problem, as partitioning is performed by moving the separator symbol.

Another modification, proposed in [106], models the clusters as closed graphs. The crossover operation is carried out in such manner that offsprings inherit edges (links) from their parents. However, this approach is computationally very expensive (of order $O(K^6 + N)$, where K denotes the number of clusters and N is the number of points). Therefore, it is not suitable for large data sets.

5.2.5 Modern approaches

Competitive agglomeration

The competitive agglomeration algorithm is a modern approach to fuzzy agglomerative clustering. It is based on minimising the objective function that in general can be formulated as [89]:

$$J = J_1 + \alpha J_2, \quad (5.3)$$

where J_1 is similar to the objective function in the fuzzy c -mean algorithm (Equation 5.2) and is responsible for the shapes and sizes of clusters. It forces clusters to be compact [88] as it reaches its minimum when each point is in a separate cluster. The second term J_2 is intended to reduce the number of clusters. In the original algorithm [88] J_2 is proposed in a form:

$$J_2 = - \sum_{c=1}^C \left[\sum_{n=1}^N u_{cn} \right]^2 \quad (5.4)$$

which reaches its minimum when all points are in a single cluster. The weight α provides the balance between the two components of objective function. By choosing carefully the value of α , the right combination of the number of clusters with their compactness can be obtained.

The membership function is constrained by the condition:

$$\sum_{c=1}^C u_{nc} = 1, \text{ for } n \in \{1, \dots, N\}. \quad (5.5)$$

An essential advantage of this algorithm is its ability to estimate the number of clusters required for accurate representation of the data. This is achieved by the careful choice of α values and by removing the clusters with cardinalities below a predefined threshold [89]. It is suggested in [88] to start with each data point in a separate cluster and to use a large value of α to allow rapid reduction in the number of clusters at the beginning of the clustering process. Then, the value of α should be decreased gradually to force clusters to compete for data points.

Some modifications to this algorithm were proposed in [89] and [90]. In the former article it is proposed to use the entropy function as the J_2 . In the later paper [90], an algorithm based on competitive agglomeration, called adaptive robust competition (ARC), is introduced. The ARC algorithm takes into account outliers that occur in data.

Normalised cuts (N-cuts)

Normalised cuts is a modern divisive (top-down) algorithm based on graph theory. It extends the idea of minimum cut [95] for partitioning a graph [87]. The

minimum cut depends on the number of links and the size of each subgraph and favours the creation of small subgraphs. In contrast, the normalised cuts criterion is independent of the size of subgraphs, avoiding the bias towards small clusters [87].

If one considers a graph V consisting of two disjoint subgraphs A and B , $A \cup B = V$, $A \cap B = \emptyset$, with data points $u \in A$ and $v \in B$, the normalised cut is defined as [87]:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \quad (5.6)$$

where $cut(A, B)$ is defined in Equation 5.1 and

$$assoc(A, V) = \sum_{\forall u \in A, \forall t \in V} w(u, t), \quad (5.7)$$

where $w(u, t)$ denotes the cost of the edge (link) between points u of the subgraph A , and points t of the full graph V . The major disadvantage of the N-cuts algorithm is the lack of ability to estimate the number of clusters.

A solution to finding optimal N-cuts for image segmentation by converting the problem to a generalised eigenvalue problem is proposed in [87]. It is also proposed to solve this problem in two ways: recursive two-way cut and simultaneous k -way cut with multiple eigenvectors. The former solution bipartitions the data set recursively, so that $Ncut$ is minimised. The decision to divide the current partition depends on the stability of the solution and the value of $Ncut$ (which should be below a predefined value). The latter method deals with the problems of oscillatory eigenvectors, which is not fully resolved in the former solution. It also is computationally more effective, because k partitions are created at each iteration [87].

5.2.6 Conflicting information

In recent years clustering has been approached using Belief Theory (BeT). BeT is described in detail in Section 6.2.2. The clustering methods based on BeT utilise the ability of the BeT to find out conflicting information.

In [97, 107] the k -NN (k -nearest neighbours) technique is adapted for utilising BeT. It is assumed that N patterns in a training set $\{x_1, \dots, x_N\}$ are to be classified into C classes $\Omega = \{\omega_1, \dots, \omega_C\}$. The classes form the set Ω of hypotheses and the basic belief assignment (BBA) function $m(\omega_c|x_n)$ expresses the amount of belief that evidence x_n belongs to the class ω_c . The value of BBA is derived from a distance measured between the given data point x_n and the c th cluster:

$$m(\omega_c|x_n) = \alpha\phi(d_{cn}) \quad (5.8)$$

$$m(\Omega|x_n) = 1 - m(\omega_c|x_n) \quad (5.9)$$

$$m(A|x_n) = 0, \quad \forall A \in 2^\Omega - \{\Omega, \{\omega_c\}\} \quad (5.10)$$

where $m(\omega_c|x_n)$ is the weight of belief that the n^{th} data point belongs to class ω_c given its distance d_{cn} to this cluster, and α denotes the value of discernment, indicating the reliability of given similarity measure. The function $\phi(d)$ maps the distance measure into the range of $(0, 1)$ providing $\phi(0) = 1$ and $\lim_{d \rightarrow \infty} \phi(d) = 0$. A function commonly used for Gaussian clusters is [97]:

$$\phi(d) = \exp(-\gamma d^2) \quad (5.11)$$

5.3 Clustering of similar faces

Three methods, from the ones described above, were chosen by the author and modified in order to deal with outliers and to estimate the number of clusters. The first approach chosen was k-means. The k-means algorithm was chosen because of its popularity and proven efficiency despite its simplicity. The modifications dealing with the initial number of clusters were inspired by the generalised Lloyd algorithm (GLA), a variant of k-means algorithm used in colour quantisation (see Section 5.2.2). Details on the modifications and the method and experiments carried out with the k-means algorithm are presented in Section 5.3.1.

The second algorithm chosen for further investigation and experimentation is a variant of a single-link algorithm. This method was chosen in order to test the behaviour and efficiency of an agglomerative clustering algorithm. Additionally this method exploits similarities between data points, not between a data point and the model of a cluster, and this seems to be more appropriate approach for clustering applied to face recognition.

The last method chosen is the nearest neighbour-based algorithm (see Section 5.2.2). This approach was chosen because the nearest neighbour classifier is most commonly used in face recognition. This method estimates the number of clusters and is flexible enough to create clusters of various, sometimes complicates shapes. Therefore, it should work well with face recognition data.

5.3.1 Approach I: Outlier based divisive algorithm

The algorithm

The algorithm provided in this section estimates the number of required clusters using outlier information. The outliers are detected by simple thresholding, i.e. points with the distances to all clusters above a given threshold are treated as outliers. The outliers are used for creating new clusters, as they are seen as points lying too far from any of existing clusters. The algorithm is realised in an iterative way.

Algorithm modified k-means:

1. Create one cluster containing all data points
 2. Compute parameters of this cluster
 3. find outliers by thresholding, create new cluster consisting of outliers
 4. stabilise new clusters
 - (a) calculate clusters parameters
 - (b) reassign data points to clusters
 - (c) if the number of clusters which changed membership is below a threshold stop; otherwise go to step 4a
 5. if there is at least one empty cluster stop; otherwise go to step 3
-

Iteration

The algorithm starts with one cluster that contains all data points. Then, its parameters are calculated and distances of all points from the centre of the cluster are computed. These distances are used for detecting outliers by thresholding.

The outliers are used for the creation of a new cluster. Now any of the clustering algorithms such as k-means or EM can be used for performing clustering on these two clusters. Once the convergence of these algorithms is reached (clusters are stabilised) outliers are detected. A new cluster is created and its parameters are computed as all current outliers become members of this new cluster. Then, another set of iterations of a classic clustering algorithm such as k-means is run on the increased number of clusters.

Stopping criterion

The number of clusters can be increased to a certain level, above which empty clusters would appear. When an empty cluster appears it means that no more clusters can be created, as points are attracted to the existing clusters, and detected outliers do not form any separate cluster. Therefore, the iterations are stopped when an empty cluster is detected.

Distance measure

The distance measure between points and cluster centres is the L_p norm. For $p = 2$ it ensures the Euclidean measure in the subspace. However, it does not take into account the spread of points within a cluster. Therefore, a given threshold results

in all clusters having a similar spread of points. However, this is not necessarily true, as clusters may contain points of wider spread than other clusters. In order to take into account the spread of points within clusters, the distance measure is normalised by the variance of clusters.

The L_2 norm ensures circular distribution of points within a cluster. In the case of the MPEG-7 face recognition descriptor, a weighted L_1 norm is used for finding distances between points (see Appendix B).

Outliers detection

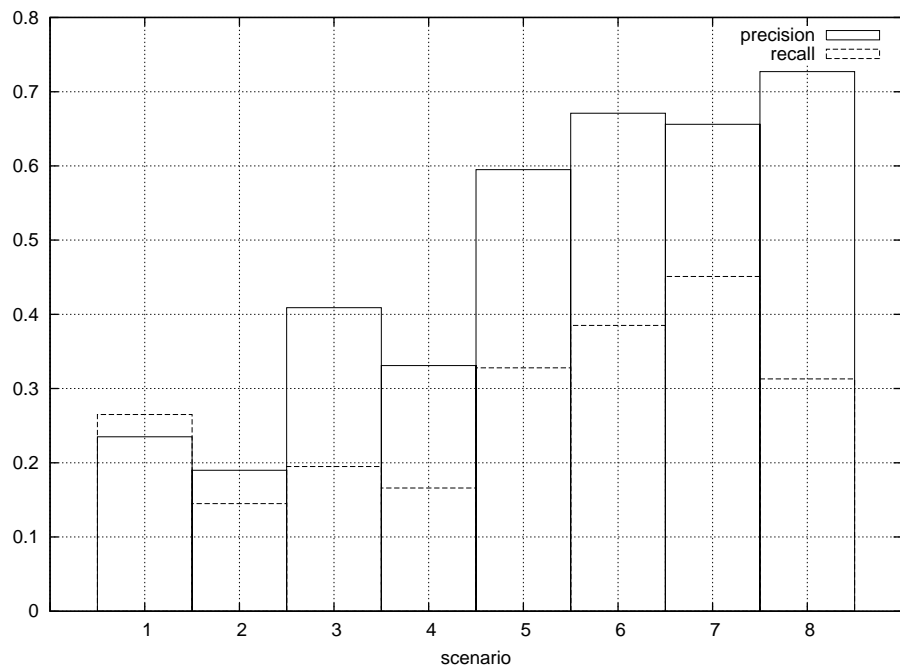
The outliers are detected by simple thresholding. It is assumed that the points lying far from the centres of all clusters (“far” in this case is defined as a value above a threshold) do not belong to any of the clusters and thus they are outliers. The accurate threshold is found by carrying out the clustering for several different values of the threshold and choosing the one that gives the highest number of clusters.

If the threshold is too low, then all points are viewed as outliers and assigned to one cluster — the one containing outliers. If the threshold is too high, then only the initial cluster is created and no outliers are detected. For a value of threshold between these extremes, the number of clusters is higher than one. If the number of clusters for a given threshold is higher than for any other threshold, then this threshold is well balanced between too strict and too weak outlier detection. At this value of threshold, results should be close to the optimal ones.

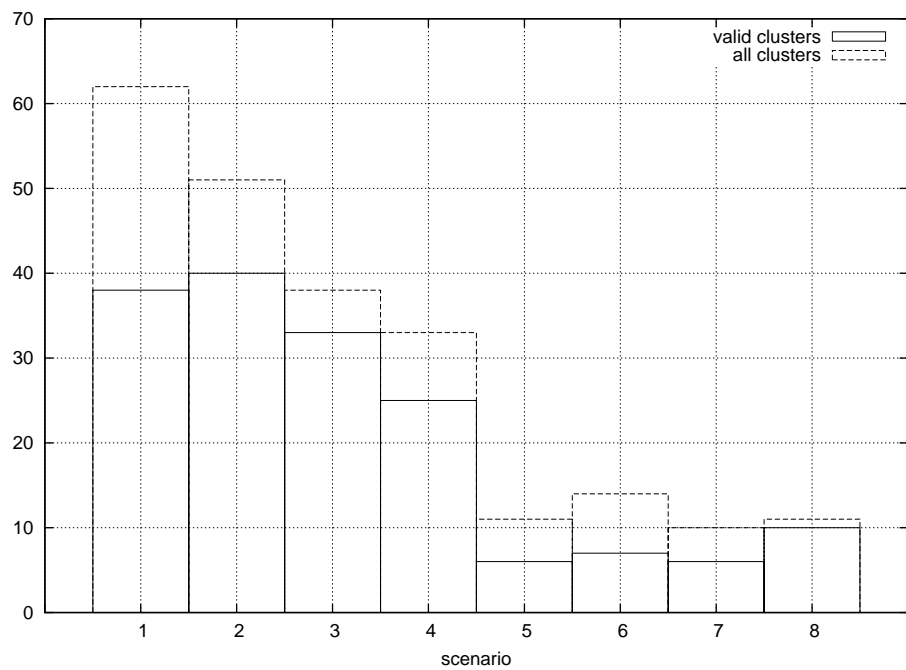
Evaluation

The evaluation of the technique was carried out by the author using the data set presented in Appendix C.2. The MPEG-7 FR features were used for similarity matching. In order to evaluate how outliers affect the clustering process, the experiments were carried out in both situations: using a data set with outliers (with “Unknown” identity) and with outliers removed (with every “Unknown” person removed from data set). Also, the influence of automated eye localisation (presented in Section 3.4) on the accuracy of the clustering process was analysed by running experiments with features extracted from facial images created using both manual and automated eye localisation.

The experiments were carried out in eight scenarios. These scenarios are presented in Table 5.1. For each scenario, several clustering experiments were carried out using thresholds from specified ranges. Figure 5.2 presents results obtained for each scenario with the best thresholds. These thresholds were chosen using the rule by which the best threshold is the one that gives the highest number of created clusters. Detailed results for different ranges of thresholds are presented in Appendix D.1.1. Figures D.1, D.2, D.3 and D.4 present results of clustering using



(a)



(b)

Figure 5.2: Summary of results obtained with modified k-means algorithm for scenarios presented in Table 5.1.

Table 5.1: Scenarios of experiments with clustering of similar faces.

scenario	data set	outliers	eye locations
1	large	yes	manual
2	large	yes	auto
3	large	no	manual
4	large	no	auto
5	small	yes	manual
6	small	yes	auto
7	small	no	manual
8	small	no	auto

MPEG-7 FR features extracted with manually located eyes. Figures D.5, D.6, D.7 and D.8 present detailed results of clustering in similar scenarios, but the eyes were located in an automated way described in Section 3.4. The meaning of values presented, such as precision, recall and the number of valid clusters is described in Appendix C.

In scenario 1, 38 valid clusters were produced at the threshold 420. This gives precision 0.235 and recall 0.265. The number of valid clusters is lower than the desired number, indicating that not all identities were captured in the clustering process. There is a high peak in both precision and recall at threshold 670 (see Figure D.1). However, in this case only two clusters are regarded as valid, thus this situation must be discarded.

In the case of the smaller set, the choice of optimal threshold is not easy as the highest number of clusters appears for a number of threshold values. Overall, for the smaller set, the precision is higher than for the larger set. This observation is exploited in Chapter 6 for enhancing clustering results.

In scenarios 3 and 4, the data sets without outliers are used. Still, the clustering process produces small number of non-valid clusters i.e. clusters consisting of one element. The appearance of faces in those clusters must be so distinct that they are seen as outliers even when there are no outliers.

When the automatically obtained eye locations are used for feature extraction, the range of usable values of threshold is narrowed (see Appendix C). In the second scenario, 51 clusters are produced, forty of which are valid. This gives average precision 0.190 and recall 0.145. For a small data set (scenario 6), just one threshold produces the highest number of clusters. Thus, there is no ambiguity in the choice of the threshold. For the threshold 290 there are 14 clusters. Seven of them are valid giving precision 0.671 and recall 0.385.

Similar experiments were conducted with a different distance function, which takes into account the spread of points within each cluster. There are two distance measures considered:

- the multivariate Gaussian distribution, equivalent to Mahalanobis distance

- one-dimensional normalised distance based on the dimension of the distance measure used

$$d_m(x, x_c) = \frac{d^2(x, x_c)}{\sigma_c^2} \quad (5.12)$$

where $d(x_i, x_c)$ is a distance (weighted Manhattan distance in the case of MPEG-7 FR) between the i th point x_i and the centre x_c of c th cluster, $\sigma_c^2 = 1/N_c \sum_{i=1}^{N_c} d(x_i, x_c)^2$ is an average spread of data points x_i , $i = 1, \dots, N_c$ (N_c denotes number of points in c th cluster) in the c th cluster measured in an appropriate distance.

The latter of the proposed distance measures does not describe shapes of clusters (which are described by the original distance $d(x_i, x_c)$ used) but rather the spread of points within the cluster. For example, if $d(x_i, x_c)$ is the Euclidean distance, the clusters have a hyperspherical shape, but might have different diameters depending on the spread of points within clusters.

The results of experiments carried out with the modified k-means algorithm and the normalised distance are presented in Figure 5.3. It can be clearly seen that this algorithm works well with a small data set but fails on a larger data set. Additionally the number of valid clusters is very low. Detailed examination of the results of clustering on the large data set (see Appendix D.1.2) allows one to conclude that the higher values of threshold are required, since most of the data points are classified as outliers. This suggests that the value of threshold depends on the size of a data set. Therefore, it should be adjusted for each data set separately.

5.3.2 Approach II: Single-link

The algorithm

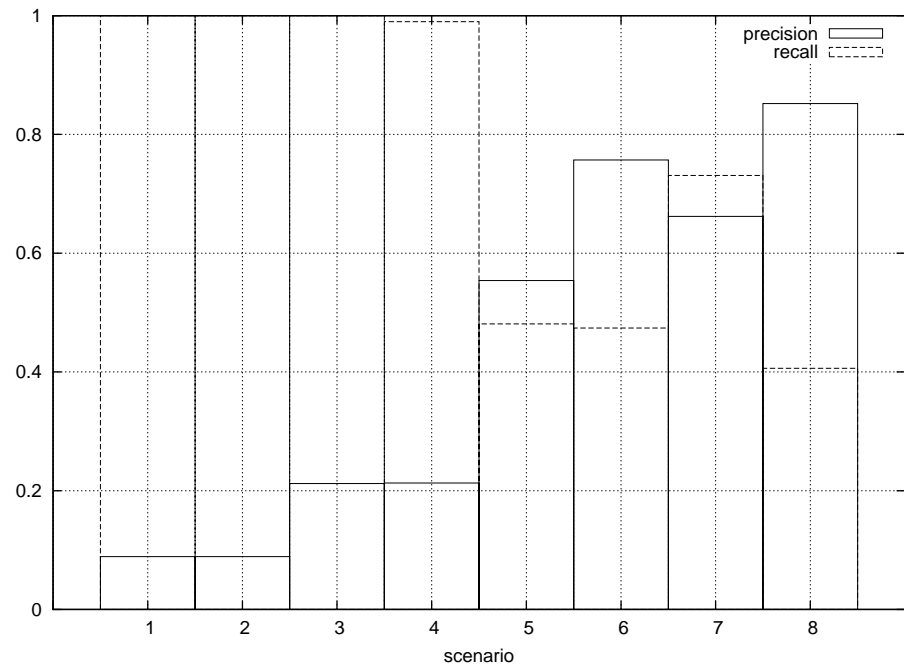
The single-link method proposed in this section is an agglomerative version of the outlier detection-based algorithm. The neighbourhood of each data point is searched within the distance defined by a threshold. Any point which lies within such defined neighbourhood is merged with the current point.

This algorithm is very quick as the process is not iterative. It uses distances between points stored in a distance matrix, which requires pre-computing of all $n(n-1)$ distances.

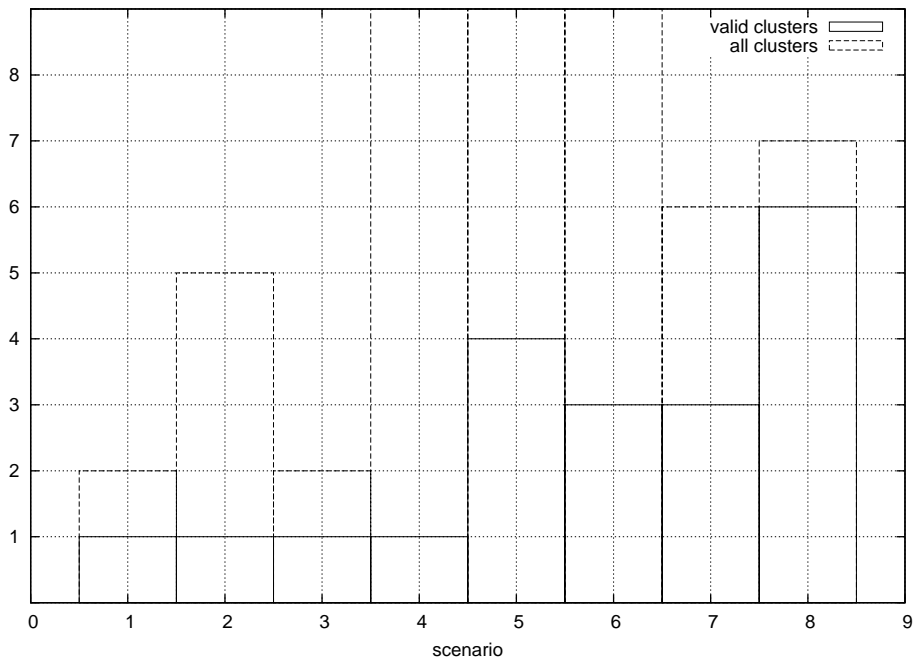
The algorithm inherits the drawback of outliers detection algorithm that the performance depends on a good choice of the threshold.

Distance measure

The distance measure used in this algorithm is a value of distances between each data point, not between a data point and a cluster centre as it is in the k-means



(a)



(b)

Figure 5.3: Results of clustering in scenarios outlined in Table 5.1 obtained with a use of Mahalanobis-like distance measure.

Algorithm single-link:

1. calculate all $n(n-1)$ distances between points
 2. go through every data point, for each:
 - (a) find points within a given distance
 - (b) merge these points to the cluster of current point
 3. if all points visited stop; otherwise go to step 2
-

algorithm presented in the previous section. This distance measure in the case of MPEG-7 face recognition features is the weighted L_1 norm. For n points, $n(n-1)$ distances need to be calculated. These are pre-computed before grouping is started and are stored in a matrix. This reduces the computational time as distances are computed only once.

Outliers detection

Again outliers are detected by simple thresholding. However, the values of thresholds are from ranges different than previously, as thresholding is applied to distances between points, not between points and clusters. In contrast to the previous method, where outliers are put into a dedicated cluster, in this method each outlier is placed into a separate cluster. Therefore, the number of all clusters can be very high.

It is more difficult to find an accurate threshold in this case. If the value of threshold is too large, the algorithm produces just one cluster containing all data points. However, if the threshold is too small, all points are detected as outliers and remain as separate clusters. Therefore, the method of finding the optimal threshold must be modified, as the maximal number of clusters is obtained when each data point is in a separate cluster, i.e. when the value of threshold is extremely small. A simple solution is to modify counting of clusters in such way that all clusters containing just one data point are omitted. Then, for small numbers of the threshold, the number of clusters with more than one member is small, increasing with increased value of threshold and then decreasing when the threshold becomes too large.

Evaluation

The experiments were carried out by the author using the testing collection described in Appendix C.2, with measures as in Appendix C.3. Figures D.17 – D.24 present graphs showing precision and recall for single-link clustering at different

threshold levels in scenarios outlined in Table 5.1. The summary of results is presented in Figure 5.4. The summary contains analysis of results obtained for the values of thresholds that give the highest number of clusters with cardinalities higher than one. This is the rule for choosing an optimal threshold in this method.

In scenario 1, clustering using the threshold 220 gives 40 valid clusters with precision 0.731 and recall 0.165. This looks very promising. However, it does not take into account the structure of created clusters. This clustering method produces one large cluster and many small clusters. The small clusters have large precision, but they contain just a fraction of all images of a given identity. The large cluster contains a high number of photographs e.g. for a threshold of 220, this large cluster contains 84 faces, which is almost 7.5% of all faces in the collection. The structure of clusters in other scenarios is very similar to clusters created in scenario 1, i.e. there is one large cluster with several small ones.

5.3.3 Approach III: nearest neighbour

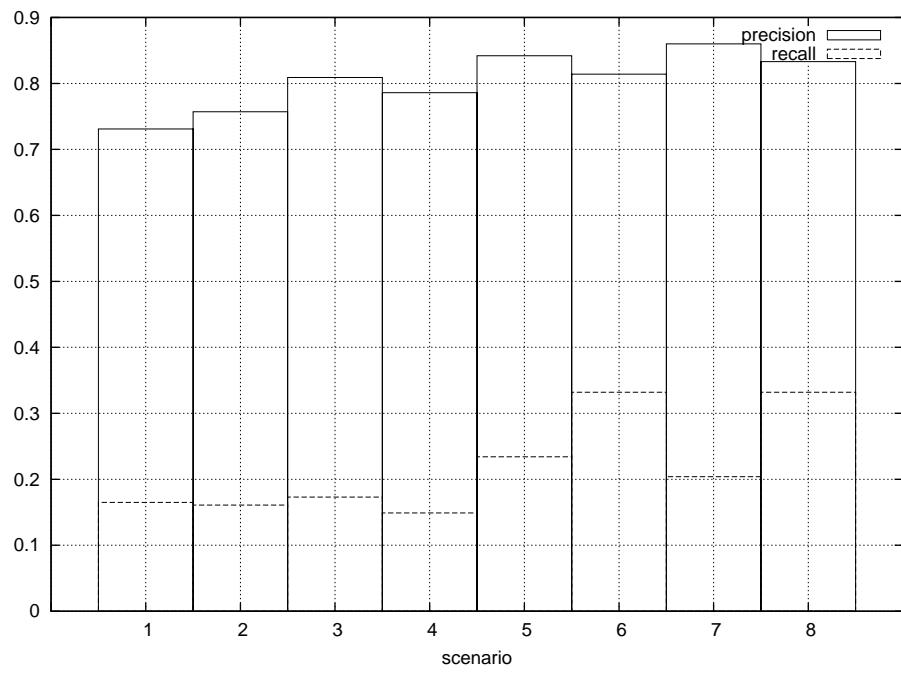
The algorithm

The algorithm based on the nearest neighbour classifier is an agglomerative method similar to the single link one. Unlike the single-link algorithm, for every data point just one other point is considered — the one that is closest in terms of a distance measure. This nearest point is linked to the current point. Linked points form a cluster that might consist of all points in a space. However, most often it results in a few large clusters surrounded by small clusters consisting of few points. Choosing only one data point to be linked can be seen as a thresholding. Here, thresholding is applied to the number of points in the neighbourhood in contrast to the approaches presented previously. This method also emerges from typical face recognition algorithms, where unknown faces are labelled with the label of the most similar known face, rather than the mean face of a given class (identity).

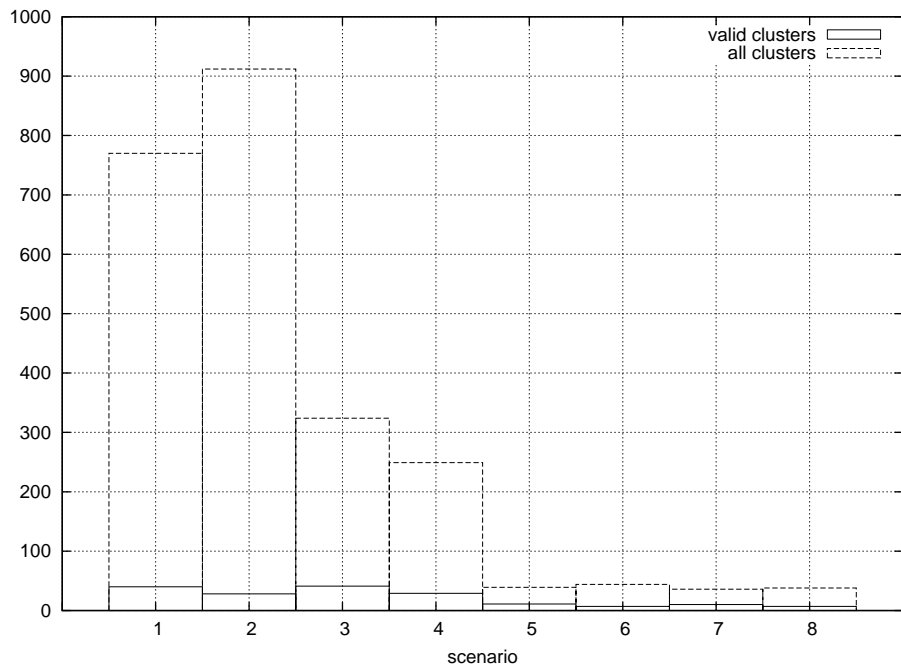
The advantage of this approach is that there is no need to determine the number of classes before running the clustering process. Points within a cluster might form several different and sometimes complex shapes, e.g. paths, stars, rings (see Figure 5.1).

Distance Measure

The distance measure is applied (similarly to the single-link algorithm) to the distance between data points. As the nearest points are merged into a cluster (these points might be already members of separate clusters), this can be viewed as a lowest distance between the edges of clusters. The distances between points can be pre-computed. Therefore, this computation is carried out only once and it speeds up the execution time of the algorithm.



(a)



(b)

Figure 5.4: Summary of results obtained with modified single-link algorithm for scenarios presented in Table 5.1.

Algorithm nearest neighbour based:

1. calculate all $n(n-1)$ distances between points
 2. go through every data point, for each:
 - (a) find the closest point
 - (b) merge cluster of the closest point with the cluster of the current point
 3. if all points visited stop; otherwise go to step 2
-

Outliers

The outlier detection can be realised again by thresholding. A data point that lies at a distance higher than a given threshold (given by means of the value of the distance) to any nearest point might be considered as an outlier. However, points that are very distant from other points might form very small clusters containing only a pair of data points. Therefore, even if the threshold is not set for outliers detection, outliers may be separated in small, two-points clusters.

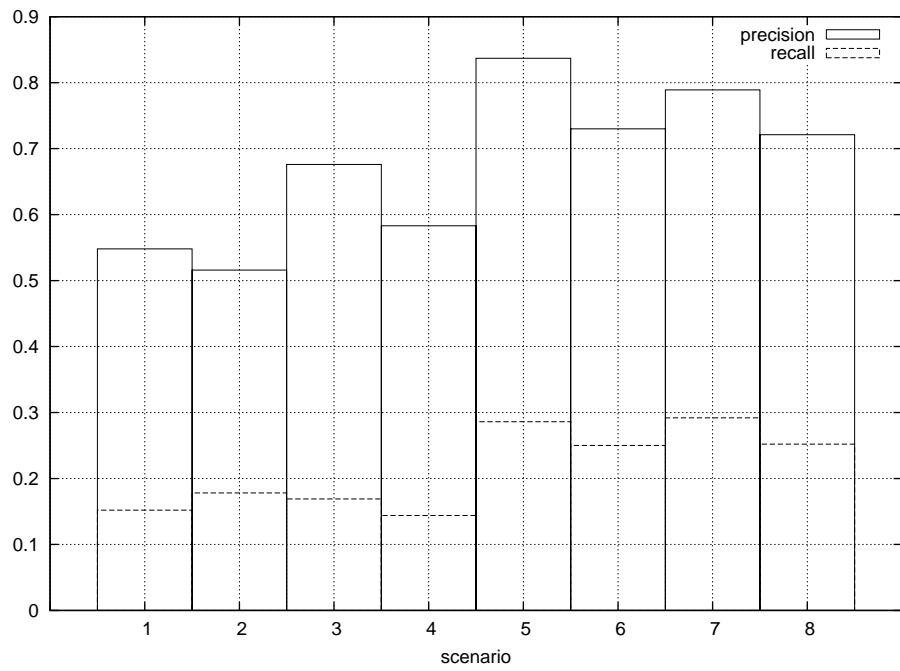
Evaluation

The algorithm was evaluated by the author using the test collection described in Appendix C.2, with measures as in Appendix C.3. Similarly to evaluating the k-means algorithm, the experiments were run using the data set containing outliers and the second set with outliers removed. The results are presented in Table 5.2 (there is no threshold to be set, therefore a single value is obtained) and the graph in Figure 5.5.

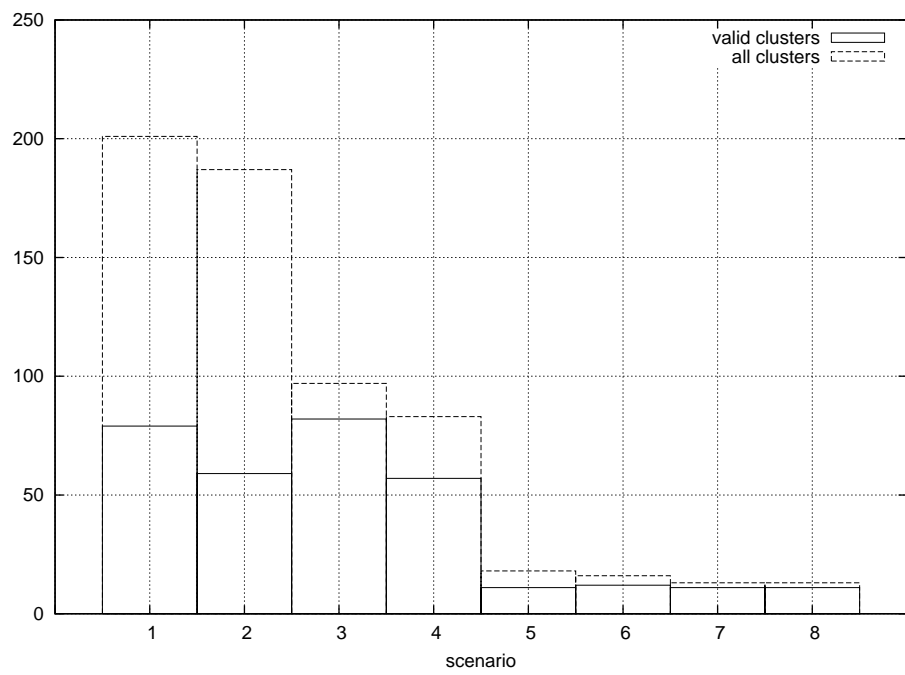
One can observe that the nearest neighbour-based algorithm produces more valid clusters than the previously investigated techniques. This number is larger than the number of identities in collections. This behaviour is more suitable for organising a photograph collection as several clusters can be presented to a user, thus most of the photographs of a given identity can be shown.

5.3.4 Conclusion

In assessing the effectiveness of clustering of human faces, the following measures must be analysed: precision, recall and the number of created clusters. The ideal number of clusters is 54 for the large data set and 7 for the small data set, as there are that many known identities in each data set. The nearest neighbour based technique gives the best results in estimating the number of clusters of the larger database. When outliers are included, the number of valid clusters is larger than the number of identities in the data set. This, however, can be seen as an



(a)



(b)

Figure 5.5: Results of clustering with nearest neighbour technique.

Table 5.2: Precision and recall of clustering using nearest neighbour based algorithm.

scenario	no of all clusters	no of valid clusters	precision	recall
1	201	79	0.548	0.152
2	187	59	0.516	0.178
3	97	82	0.676	0.169
4	83	57	0.583	0.144
5	18	11	0.837	0.286
6	16	12	0.730	0.250
7	13	11	0.789	0.292
8	13	11	0.721	0.252

advantage. In the situation when recall values are not high, it gives a greater possibility that the higher number of known faces is included in the valid clusters. It is important to note that the number of valid clusters does not depend much on the existence of outliers, it rather depends on the method of finding locations of eyes (see Figure 4.5(b)). For the smaller data set, the number of valid clusters is independent of scenario and again produces more valid clusters than the number of identities.

In the case of the k-means and simple-link algorithms, the estimated number of clusters is lower than the number of identities when dealing with the large data set. This is a serious drawback of these methods as it means that some identities are rejected as outliers. This happens even when the outliers are excluded from the data set. Similarly to the nearest neighbour-based method, the number of valid clusters does not depend on the existence of outliers in the data set, it depends more on the accuracy of eye locations. The k-means algorithm does not behave well in the scenario with the automated locations of eyes and removed outliers (scenario 4), as it produces very few valid clusters. When dealing with the smaller data set, the estimates of the number of clusters given by both algorithms are similar to the number of identities in the data set. However, considering a rather low recall in these cases, it would be preferable to have a larger number of valid clusters, which is provided with the nearest neighbour-based algorithm.

The presented clustering methods can be compared in regard to precision and recall using Figure 5.6. Points placed more to the right and higher indicate better performance. It can be clearly seen that the simple-link algorithm gives best results for all scenarios. The nearest neighbour-based technique gives good results in the scenarios with the larger data set. For the smaller data set the precision values are high, however, at the expense of low recall. As it is mentioned above this behaviour is compensated with the higher number of valid clusters. The method based on the k-means algorithm gives worst results among all three tested

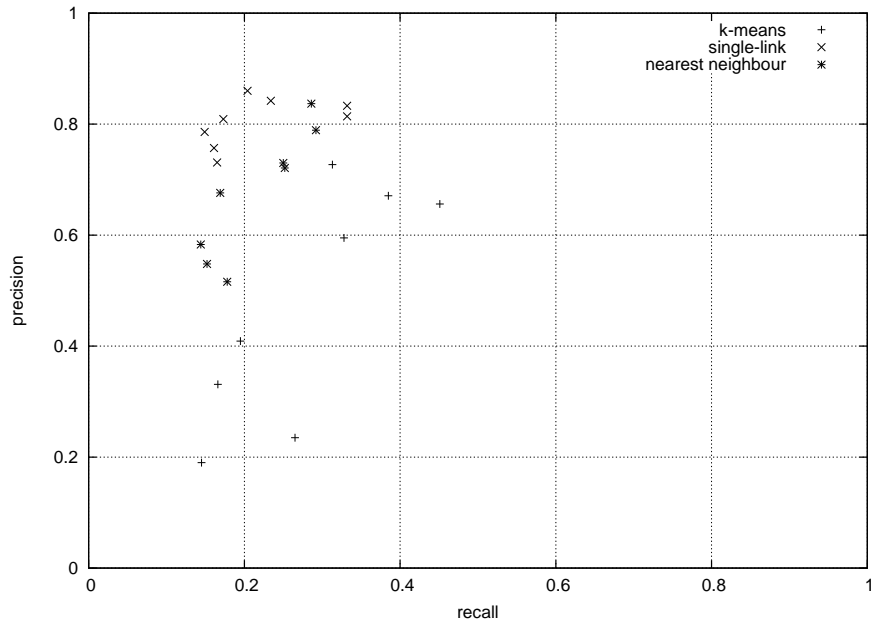


Figure 5.6: Precision as a function of recall for three clustering methods: k-means, simple-link and nearest neighbour based.

methods in the case of large data set. As it matches other methods in the recall values, precision values are very low. A different situation is when the smaller data set is considered. Then, the recall values given by the k-means algorithm are best among tested methods, which in a combination with relatively high precision values suggests that good quality clusters are produced by this algorithm.

The most convenient of the three presented techniques is the nearest neighbour-based one. This method does not require any threshold. A threshold can be set for detecting outliers. However, it is not compulsory as outliers are isolated by the algorithm itself. The two other techniques require a threshold to be set. The value of the threshold can be obtained by experiment. The clustering process needs to be carried out for several different values of the threshold, then the "best" threshold is chosen. However, this is less convenient and computationally more expensive.

In summary the nearest neighbour based method works best in the terms of estimating the number of clusters. It gives the average results in most scenarios if recall and precision are considered. However, they are acceptable, as they are consistent. The low values of recall can be explained by the higher number of valid clusters than the number of identities in the data set. The simple-link algorithm gives the best results regarding values of precision, without any loss in recall in the comparison to other techniques. The number of valid clusters produced with this technique is lower than the number of identities. This can be seen as a serious drawback. The k-means approach produces clusters characterised with the best recall. However, this is achieved at the cost of low precision and the small number

of valid clusters. This can be clearly seen in the case of the larger data set. For the smaller data set, k-means provides the best balance between precision and recall.

In the case when the number of clusters and precision are priorities, the nearest neighbour technique is the method of choice among the three presented techniques.

5.4 Duplicates detection

It might happen that due to similar environmental conditions or any other similarity, two faces appearing in the same image are classified to the same cluster. This should not happen as it is very unlikely that the same person appears twice in the same photograph, unless a reflection of the face was captured in such a way that both the face and its reflection are visible. Therefore, if two faces from the same photograph are classified to the same cluster, it indicates that at least one of them is wrongly classified and should be removed from this cluster.

Although detection of duplicates is trivial, the correction of such situation is not. In order to correct such wrong classification some decisions must be made. One of the most important is which one of the instances of persons should be removed. If the parameters of a cluster are available, as it is in the case of k-means clustering, the solution might be to remove the person (face) that is further away from the centre of the cluster, since it is less similar to the average content of that cluster.

In the case of nearest-neighbour clustering, choosing the data point for removing and the process of removal is not that simple. All points in a cluster are connected to their neighbours with links. Therefore, it is hard to find the parameters for a cluster, as it resembles a connected graph, and as a result of this there is no basis for making the decision on which point should be removed. Furthermore, removing a point would very often result in broken links in the cluster graph, splitting this cluster into two subclusters. It is then difficult to find the optimal cut point.

One could try to artificially increase the distance in a feature space between two faces appearing in the same image to a very large value. The higher distance should ensure that they will not be connected. However, they still can end up in the same cluster as they might get connected through other data points.

Another solution could be calculating the centre of the cluster and analysing distances of points inside this cluster from its centre. This would not describe the structure of the cluster properly. However, it would be much easier to choose which of the data points should be removed. As a side effect this would upset the parameters of the other cluster.

5.5 Conclusion

In this chapter three approaches to grouping of points similar in terms of face appearance are described. This is the third step of a clustering process as outlined in Section 1.2. The only features used for this grouping are features based on the appearance of the human face. These features are not sufficient for finding similar people in a large photograph collection as the efficiency of clustering decreases with the increase of the size of the collection.

In the next chapter the use of context and content information additional to facial recognition for enhancing the clustering process is analysed. It is proposed to exploit event information and user information. Also, additional information about the appearance of the person is used: the colour and texture of clothes the person is wearing.

Chapter 6

Information fusion

6.1 Introduction

The features for clustering, extracted from objects, do not necessarily need to be of the same kind, or come from the same source of information. Sources other than face recognition, can be used for enhancing the clustering presented in Chapter 5. In the case of a photo collection, in addition to face analysis, information can be extracted from body patch analysis, event context (a higher level information based on location and time) and the person who (or camera which) captured the photograph. This chapter presents major existing techniques for information fusion and also describes two new techniques: one based on Belief Theory (BeT) and another one, a novel approach proposed by the author, based on event information used for restricting the search space and then using user information for merging clusters.

The fusion can be done in two ways: the fusion of features (early fusion) and the fusion of decisions (late fusion). Most techniques perform the fusion at the feature level, combining distances between points based on several measures or sources and then making a decision based on the combined measure. The others, which do the fusion at decision level, firstly quantise each of the sources (make decisions) and then analyse decisions made on each source. Schemes of both methods are presented in Figure 6.1.

6.2 Existing techniques

6.2.1 Simple probabilistic techniques

The simple probabilistic methods are based on a probability model of classification. There are several methods that can be described:

- average sum

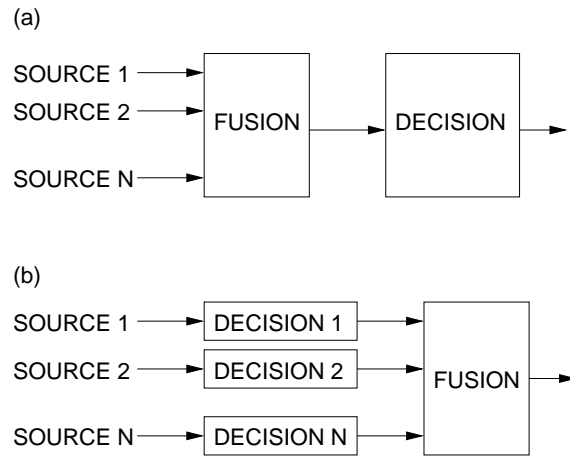


Figure 6.1: Two fusion models: (a) fusion of features, (b) fusion of decisions.

- generalised product
- maximum (equivalent to minimum inverted)
- voting

Average sum

The average sum is a method of feature fusion. The weighted average of probabilities given by different sources is obtained as:

$$P(H) = \sum_{i=1}^I w_i P_i(H) \quad (6.1)$$

where $P_i(H)$ is the probability given by the i th source supporting hypothesis H and w_i denotes weights assigned to each source, which represent the reliability ascribed to the particular source. This can be generalised in a form similar to the L_p norm:

$$P_r(H) = \sum_{i=1}^I w_i P_i^r(H) \quad (6.2)$$

where r is a power factor. For $r = 2$, one would obtain the fusion in terms similar to Euclidean distance. For $r = \infty$, this rule becomes the maximum rule.

The average sum works well in most scenarios. However, in some situations this rule might average and compromise results, especially when one of sources is very reliable and others are not. Then, using just the most reliable source gives better results than combining all sources by the average sum [108].

Generalised product

The product rule is very similar to the average sum rule, simply with a product used instead of the sum:

$$P(H) = \prod_{i=1}^I P_i^{w_i}(H). \quad (6.3)$$

The product rule can be easily converted to a sum using the logarithm function:

$$P(H) \propto \sum_{i=1}^I w_i \log(P_i(H)). \quad (6.4)$$

The product rule shares the same disadvantage as the average sum method, i.e. it might compromise results. It produces different results to the average sum, because the logarithm (Equation 6.4) reduces the influence of infrequent but very high peaks and puts more weight on the lower, more frequent values of $P_i(H)$.

In the case of just two hypotheses, true H_t and false H_f , one can compute a ratio

$$\frac{P(H_0|I)}{P(H_a|I)} = \frac{\prod p_i}{\prod (1-p_i)} \propto \sum (\log p_i - \log(1-p_i)). \quad (6.5)$$

This can be further simplified, and normalised by the number of sources:

$$\sum (\log p_i - \log(1-p_i)) \propto \lim_{r \rightarrow \infty} \sum (p_i^{1/r} - (1-p_i)^{1/r}) \quad (6.6)$$

$$\frac{P(H_0|I)}{P(H_a|I)} \propto \frac{\sum (p_i^{1/r} - (1-p_i)^{1/r})}{n} \quad (6.7)$$

Normalisation by the number of sources is useful for comparing systems with different number of sources. For example, in image segmentation each pixel can be viewed as a realisation of a probabilistic process. When the probabilities of regions consisting of variable number of pixels are analysed it is essential to normalise these probabilities in a way that they do not depend on the number of pixels in the region [109].

When the sources of information are independent, the product rule with the w_i 's, $i = 1, \dots, I$ can be seen as the multivariate probability distribution. For example, the multivariate Gaussian distribution presented in Equation 3.13 in Section 3.2.4 can be seen as an example of the use of the product rule for combining different sources of information. Because it is assumed that luminance, chrominances and coordinates are mutually independent, their joint probability $P(x_1, \dots, x_k)$ is a product of probabilities of each source $P(x_i)$, $i = 1, \dots, k$:

$$P(x_1, \dots, x_k) = \prod_{i=1}^k P(x_i) \quad (6.8)$$

The probabilities are modelled with a Gaussian distribution, therefore, the prob-

ability becomes:

$$P(x_1, \dots, x_k) = \prod_{i=1}^k \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2} \frac{(x_i - m_i)^2}{\sigma_i^2}\right) \quad (6.9)$$

$$= \frac{1}{(2\pi)^{k/2} \prod_{i=1}^k \sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^k \frac{(x_i - m_i)^2}{\sigma_i^2}\right). \quad (6.10)$$

If the x_i are independent we can write:

$$\theta_1 = \begin{bmatrix} m_1 \\ \vdots \\ m_k \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix}, \quad \theta_2 = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k^2 \end{bmatrix}$$

and use them in Equation 6.10 to obtain:

$$P(x_1, \dots, x_k) = \frac{1}{(2\pi)^{k/2} |\theta_2|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \theta_1)^T \theta_2^{-1} (\mathbf{x} - \theta_1)\right), \quad (6.11)$$

which is the same as Equation 3.13.

Maximum

The maximum rule takes into account the maximum value of probabilities given among the I sources:

$$P(H) = \max(w_i P_i(H)), \text{ for } i = 1, \dots, I. \quad (6.12)$$

This method favours sources giving the highest values of probabilities. Its main drawback appears in situations when one of the sources is very reliable and gives values of 0 or 1. Then the outcome of the fusion depends solely on this source, which results in discarding all other information and in poor performance. This rule is also a special case of the generalised average sum given by Equation 6.2 for $r = \infty$.

The minimum rule can be also used. It takes into account the minimum value presented by sources. This rule works similarly to the maximum rule. However, it makes weighting non-intuitive, favouring sources with lower weight applied. It can be easily changed into the maximum rule by transforming values by an inverting function such as $p'_i = (1 - p_i) / \sum_k (1 - p_k)$.

Voting

Voting is a method of decision fusion. First the decision of the supported hypothesis is made by each of sources, then the hypothesis which is supported by most of the sources is chosen. Decisions are usually based on thresholding, with arbitrary

thresholds defined for each source independently.

$$p(H) \propto \frac{|P_i(H) > t_i|}{n} \quad (6.13)$$

where n is the number of sources, $|\cdot|$ denotes the cardinality of the set, and t_i is an arbitrarily chosen threshold on the i th source.

In the case of (hard) clustering, a point is assigned to the cluster that has received the largest number of votes among all sources. To avoid ambiguities it is important that the number of sources is sufficient, e.g. using just two sources might produce a lot of ambiguities if a large number of clusters is considered and sources point to different clusters instead of a strong support for one cluster.

6.2.2 Transferable Belief Model

Simple probabilistic methods presented in the previous section might be not sufficient for combining different sources of information and, as is mentioned above, they might average the results. Therefore, in this section the Transferable Belief Model (TBM) is presented and its implementation for organising a photograph collection carried out by the author is described in Section 6.4.

The Transferable Belief Model is based on Belief Theory (BeT), also known as Dempster-Shafer theory. This system is based on a system of beliefs, which is build upon a belief function and a basic belief assignment (BBA). It is defined on the power set 2^Ω (or the set of all subsets of the set Ω of events).

Belief function

The belief function is a mathematical representation of uncertainty, ignorance and any other form of partial or total knowledge. According to the Dempster-Shafer theory [110] the belief function bel is a function from some power set onto the $[0,1]$ interval.

Let Ω be a finite set and 2^Ω be its power set. Then $bel : 2^\Omega \rightarrow [0,1]$ defines the belief function with the following properties [110]:

$$\begin{aligned} bel(\emptyset) &= 0 \\ \forall n \geq 1, \forall A_1, A_2, \dots, A_n \subseteq \Omega \\ bel(A_1 \cup A_2 \dots A_n) &\geq \sum_i bel(A_i) \dots - \sum_{i < j} bel(A_i \cap A_j) \dots - (-1)^n bel(A_1 \cap A_2 \dots A_n) \end{aligned} \quad (6.14)$$

Basic belief assignment

The basic belief assignment (BBA) is a function based on the belief function representing a *mass* (or *weight*) of the given belief that a certain hypothesis H is true

[110]. The BBA assigns a value between zero and one to each subset of the power set 2^Ω . The sum of all BBAs on the given power set must equal 1, i.e.:

$$\sum_{A \subseteq \Omega} m(A) = 1 \quad (6.15)$$

where $m(A)$ is the BBA supporting a hypothesis A . The relation between the belief function and the BBA is analogous to the relation between the cumulative distribution function and the probability mass function (the probability density function in the case of continuous probabilities) in probability theory. The belief function can be expressed with the BBA as a sum of all BBAs of these subsets that support the hypothesis A :

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B), \quad \forall A \subseteq \Omega. \quad (6.16)$$

The BBA provides a one-to-one correspondence with the belief function bel , therefore, it maintains all the information represented by the belief function without any loss of information [110].

The distinction between probabilities and beliefs is in the interpretation of the value zero. The probability $P(H) = 0$ given to the hypothesis H means that this hypothesis is false. But $bel(H) = 0$ expresses total ignorance about the hypothesis H , in other words we do not know whether this hypothesis is true or false. However, both $P(H) = 1$ and $bel(H) = 1$ denote total certainty that the hypothesis H is true ($P(\bar{H}) = 1$ and $bel(\bar{H}) = 1$ express total certainty that H is false).

Other functions based on BeT

Three other functions are defined in the belief framework which combine the belief function and the BBA: plausibility, commonality and implicability [110]. However, those functions are not employed in the clustering model presented in Section 6.4, thus there is no need to discuss them further here.

Dempster's rule of combination

Two belief functions can be combined using the Dempster rule of combination [110, 111], which lays the base for combining two and more sources of information:

$$m_1 \cap_2(A) = \sum_{B \cap C = A} m_1(B) m_2(C) \quad \forall A \subseteq \Omega \quad (6.17)$$

where m_1 and m_2 are two BBAs on Ω . The operator of Dempster's rule of combination is very often denoted as \oplus and is symmetric, associative and commutative. As a result of its characteristics it is easy to extend the rule to more than two

sources, as

$$m_{1 \cap 2 \cap 3 \cap \dots \cap N} = (((m_1 \oplus m_2) \oplus m_3) \oplus \dots) \oplus m_N. \quad (6.18)$$

The Dempster rule of combination as defined in Equation 6.17 is called a conjunctive rule of combination and can be used when all sources of information are reliable. In the case when some sources are not reliable and one does not know which source is reliable, the disjunctive rule of combination defined in Equation 6.19 should be used [110]:

$$m_{1 \cup 2}(A) = \sum_{B \cup C = A} m_1(B)m_2(C) \quad \forall A \subseteq \Omega \quad (6.19)$$

Normalisation

One of the basic assumptions of the Dempster-Shafer theory, outlined by Shafer [111] is the assumption that $m(\emptyset) = 0$ and $bel(\Omega) = 1$. For these conditions to be met one needs to normalise the result of Dempster's rule of combination. The normalisation can be obtained by rescaling the resulting BBA:

$$m'_{12}(A) = \frac{1}{1 - m_{12}(\emptyset)} m_{12}(A) \quad (6.20)$$

Normalisation is intended to conform to the assumption of zero value of the BBA of the empty set. Therefore, it is compulsory only when the $m(\emptyset) = 0$ is required, i.e. in a closed world¹. Normalisation may be questioned and in some cases might give wrong results or a wrong conclusion can be obtained as argued in [111]. However, the weight $m(\emptyset)$ might have some meaning, which can be exploited. Let us assume that the BBA $m(A)$ is the mass of belief that the current data point belongs to the set of clusters A . In such a case the BBA $m(\emptyset)$ suggests that the current data point does not belong to any of existing clusters, thus this point might be an outlier or belongs to a new cluster that needs to be created.

Decision making

The decision in the belief theory is made using the function defined in Equation 6.21 and called the *pignistic* function, which is used for transforming the belief function into a probabilistic function [110, 111]. The pignistic function is of the

¹The term "closed world" denotes the situation when the set Ω contains all possible elements and cannot be expanded. In an "open world" scenario, it is possible that some elements are not initially included in the set Ω and thus the set Ω can be expanded by adding new elements. In terms of human identities in a photograph collection, in the closed world scenario only the certain number of identities is available. If there is a new image showing a new person, this person can be assigned only the identity that exists in the set of identities. If this is an identity that is not in the set, this person can be labelled as "Unknown", if the "Unknown" class exists. In the open world scenario, this new identity can be added to the set Ω .

form:

$$BetP(A) = \sum_{X \subseteq \Omega} \frac{|A \cap X|}{|X|} \frac{m(X)}{1 - m(\emptyset)} \quad (6.21)$$

where $|A|$ is the cardinality of the set A and $m(X)/(1 - m(\emptyset))$ is a normalised BBA. This probability function can be used for calculating expectations and provides the base for decision making.

Transferable Belief Model

The Transferable Belief Model (TBM) is a model based on Belief Theory and provides a convenient way for applications of Belief Theory [110]. It is assumed in this model that the set Ω consists of a set of worlds. The belief function expresses the belief of a source (called an agent) that the actual world (described in some way by the agent) belongs to one of the subsets of worlds.

The TBM is not based on probabilities, it just quantifies the belief of an agent about the present world. Therefore, it is more general than probability based approaches [110]. In the TBM, beliefs of several agents are combined using Dempster's rule of combination.

For the clustering problem, the set of worlds is defined as a set of clusters, and for each data point the BBA expresses the amount of a classifier's support for the hypothesis that this data point belongs to the certain cluster. More details on TBM for clustering is presented in Section 6.4.

6.2.3 Bayesian inference

In previous section, Belief Theory (BeT) is presented. In this section, Bayesian inference is presented as an alternative to BeT in order to give the reader an indication of the range of possible methods. The Bayesian Inference is broadly used for solving common fusion problems. Its principles are different from those of BeT. However, in some situations (e.g.) both methods lead to the same results.

Bayesian inference methods are based on the Bayesian rule, which is used to obtain *a posterior* probability $P(H|E)$ that a hypothesis H is true given evidence E [112]

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}, \quad (6.22)$$

where $P(E|H)$ is a conditional probability that the evidence E occurs when the hypothesis H is true and $P(H)$ is *a prior* probability that the hypothesis H is true. Two pieces of evidence E_1 and E_2 received from two sources can be combined using Bayes rule in the form of equation:

$$P(H|E_1, E_2) = \frac{P(E_1|H, E_2)P(H|E_2)}{P(E_1|E_2)}. \quad (6.23)$$

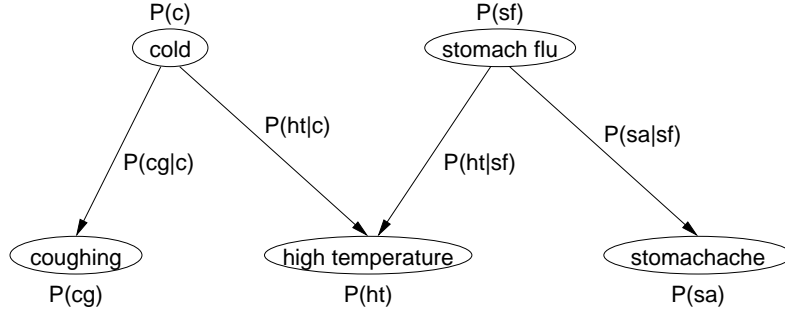


Figure 6.2: A sample Bayesian network for medical diagnosis.

giving the posterior probability of the hypothesis H : $P(H|E_1, E_2)$. If E_1 and E_2 are independent then

$$P(H|E_1, E_2) = \frac{P(E_1|H)P(E_2|H)P(H)}{P(E_1)P(E_2)}. \quad (6.24)$$

The combination of different sources using Bayesian inference can be obtained (if sources are independent) in an iterative way. This is done by substituting a prior probability in a next step with a posterior probability from the previous step:

$$P_{j+1}(H) = P_j(H|E)$$

Let us consider two pieces of evidence discussed above and calculate the posterior probability in two steps. In the first step let us consider evidence E_1 :

$$P_1(H|E_1) = \frac{P(E_1|H)P(H)}{P(E_1)}. \quad (6.25)$$

Bayesian inference is the basis of probabilistic networks, also called Bayesian networks. A Bayesian network is a graphical representation of probabilistic relationships between sources of information [113]. It is best suited for combining prior knowledge with data. However, it can also be created from the data itself. A Bayesian network takes a form of an acyclic directed graph, with the pieces of evidence (sources) in the graph's nodes and conditional probabilities represented by links (edges). Such a representation helps to reveal relationships between the various pieces of evidence.

A sample Bayesian network is presented in Figure 6.2. This is a probabilistic sample network for medical diagnosis of a cold or a stomach flu based on observations whether a patient has stomachache, high temperature and/or coughing. A probability that diagnosed patient has a cold $P(c|cg, ht)$ is:

$$P(c|cg, ht) = \frac{P(cg|c)P(ht|c)P(c)}{P(cg)P(ht)}, \quad (6.26)$$

where $P(c)$ denotes a prior probability that a patient has a cold (which might be

obtained through statistical analysis of the history of the patient's health or might depend on the season of the year, i.e. it might be higher in autumn and lower the during summer). $P(cg)$ and $P(ht)$ are probabilities that the patient shows symptoms such as coughing or high temperature respectively — these observations can be obtained by interviewing and examining a patient. The conditional probability $P(cg|c)$ is the probability that a patient has a symptom of coughing giving that she/he has a cold. Similarly $P(ht|c)$ is a probability that a patient having a cold suffers from high temperature. These two pieces of evidence might be found in a medical textbook for example.

Conclusion

In this section, some techniques for combining different sources of information are presented. These are simple probabilistic approaches such as average sum or average product, the more sophisticated approach based on Belief Theory and its alternative Bayesian approach. In the next sections, an implementation of the Transferable Belief Model proposed by the author to the problem of clustering human identities in a photograph collection is described, and the application of simple probabilistic methods to the combination of facial and body features is presented. Before that, however, available sources of information are described, since they are used by the author in implementations discussed in later sections.

6.3 Available sources

In Chapter 5, only the face recognition features are used to provide information on similarity between persons for clustering. The previous section describe techniques for combining more than one sources of information, without suggesting what the sources might be. In this section the sources of information about images and persons shown in those images are presented. These sources are used by the author for improvement of the organisation of a photograph collection . Sources of information are based on both the context and content of a photograph. The contextual sources include location, time, owner of a photograph, automatically detected events and face locations. Events are features of higher level than time and location, but based on these two. In the next sections, events, instead of time and location, are used for improving matching of similar faces. There are thus two sources of context information used by the author: events and ownership (collections of each user). A source of content information, apart from face recognition, is a person's body (torso) captured in photographs.

6.3.1 Events

Events are detected in this work using the technique described in [16]. It is based on differences in time between photographs, and the GPS coordinates of the place where photographs were captured. The GPS location is used for finding the name of the place and this name is used for event detection too.

Usually just a small subset of all identities in the collection is captured at a given event. Therefore, events can be used for enhancing the grouping of occurrences of the same persons. It is more likely that a given person appears in other photograph from the same event than in photographs captured at another event. This information is a useful source of prior probabilities.

There are possibly some identities that are more likely to appear in photographs regardless of event. It could be for example the partner of the user, or her/his child. It is expected, that using event information would increase precision. However, recall values can be lower due to some possible scatter of the images of the same person across different events. When clustering is restricted to using only images captured at the same event, the instances of the given identity appearing in other event are prevented from inclusion in the current clusters. This is what leads to the low recall.

6.3.2 User collection

The testing collection consists of photograph collections from several users. Each photograph in a user collection is annotated (automatically) with the user name. Therefore, the information about ownership is available, as it is assumed that the user is the owner of the photographs in her/his collection.

The identities of people captured in photographs of different users might overlap (users can be co-workers and take photographs at a work event) but the majority of identities would be unique for each user. Therefore, by restricting the collection to a user collection, one reduces the search space of identities, limiting the number of outliers and noise. At the same time, the loss in terms of finding all instances of a given identity is not high as only a few identities will be found in several user collections.

6.3.3 Body patch

The body patch technique for finding re-occurrences of identities in a photograph collection was proposed in [21]. It is based on the MPEG-7 Structure Colour descriptor (SC). In [21], the MPEG-7 SC is extracted from the region below the bounding box of a located face. It is based on the assumption that people in the photograph are captured in an up-right position. Thus the torso should be placed

below the face. The MPEG-7 SC extracts both colour and texture of the region, in this case these are the colour and texture of the person's clothes [21].

The use of the body patch technique by the author is based on the observation that people usually do not change their clothes during a particular event. Therefore, this technique can be successfully used for searching for re-occurrences of identities within events [21]. Some errors can be expected, as it can easily happen that clothes are either occluded by a table, bar or other obstacles, or are not visible due to dim lighting. Confusion can also easily occur if people wear similar clothes, for instance, in group photographs of co-workers wearing uniforms. Thus it is expected that this feature should give better results when combined with face recognition features, than when used on its own.

6.4 Transferable Belief Model I

6.4.1 TBM for clustering

Let us consider a set Ω of N_c clusters, each cluster representing an identity of each person captured in the collection of photographs. This gives 2^{N_c} subsets A of Ω , $A \subseteq \Omega$.

Let us denote by $m_j(C_i)$ a belief of the j th source (with m_1 being the facial analysis, m_2 assigned to events and m_3 denoting belief received from ownership information) expressed on membership of a given data point to the i th cluster. Clusters in this case are hypotheses, and sources express their belief on each hypothesis given at the current data point, i.e. they express their belief on the i th hypothesis, that the given data point belongs to i th cluster.

The combined belief from all sources is obtained using Dempster's rule of combination

$$m(C_i) = m_1(C_i) \oplus m_2(C_i) \oplus m_3(C_i). \quad (6.27)$$

Once the initial partition is available (e.g. as a result of facial clustering), the number of occurrences of elements of each cluster in any event can be calculated. This number is a base for calculating the BBA for each event and cluster. Let us assume that there are n_{ei} faces in the i th cluster in the current event, and photographs from this event contain in total n_{etot} faces. Then the BBA that the face from this event belongs to the i th cluster is:

$$m_2(C_i) = \frac{n_{ei}}{n_{etot}}. \quad (6.28)$$

Similarly one can obtain the BBA of the belief that the faces in the cluster C_i appear in a given user's collection. Denoting by n_{ci} the number of occurrences of faces from the i th cluster in a user collection containing n_{ctot} faces, the value of

Table 6.1: Sample table of masses associated with evidences and clusters; assuming three clusters in a set.

	D_1	D_2	D_3
	eye analysis	event	owner
\emptyset	$m_1(\emptyset)$	$m_2(\emptyset)$	$m_3(\emptyset)$
Cluster 1	$m_1(C_1)$	$m_2(C_2)$	$m_3(C_3)$
Cluster 2	$m_1(C_2)$	$m_2(C_2)$	$m_3(C_3)$
Cluster 3	$m_1(C_3)$	$m_2(C_3)$	$m_3(C_3)$
Clusters 1 or 2	$m_1(C_1, C_2)$	$m_2(C_1, C_2)$	$m_3(C_1, C_2)$
Clusters 1 or 3	$m_1(C_1, C_3)$	$m_2(C_1, C_3)$	$m_3(C_1, C_3)$
Clusters 2 or 3	$m_1(C_2, C_3)$	$m_2(C_2, C_3)$	$m_3(C_2, C_3)$
Clusters 1 or 2 or 3	$m_1(C_1, C_2, C_3)$	$m_2(C_1, C_2, C_3)$	$m_3(C_1, C_2, C_3)$

the BBA $m_3(C_i)$ can be obtained as:

$$m_3(C_i) = \frac{n_{ci}}{n_{ctot}}. \quad (6.29)$$

The two equations 6.28 and 6.29 can be modified for the inclusion of more than one cluster. The beliefs $m(A) = m(C_1, C_2)$ express the belief based on the evidence that the given person belongs either to cluster C_1 or cluster C_2 . The BBA $m(C_1, C_2, C_3)$ expresses the belief that the given person belongs to any of the available clusters, which, in the scenario presented in Table 6.1, represents total ignorance of the source.

Table 6.1 presents a sample TBM consisting of 3 clusters $\Omega = \{C_1, C_2, C_3\}$. There are $2^3 = 8$ BBAs that can be constructed, which are shown in the table. Given a soft clustering of faces, one can obtain three BBAs for every face: $m_1(C_1)$, $m_1(C_2)$ and $m_1(C_3)$. These are just values of the membership function. Other values of m_1 are zero.

In the case of event BBAs and user BBAs, all values can be calculated. For a given subset A of clusters, the BBA can be obtained similarly to Equations 6.28 and 6.29:

$$m(A) = \frac{\sum_{C_i \in A} n_i}{n_{tot}}. \quad (6.30)$$

This must be normalised in order to meet the condition 6.15. It can be easily shown that the normalisation factor is $2^{N_c - 1}$.

6.4.2 Implementation

Implementation of the model presented in Section 6.4.1 is difficult and challenging as the number of clusters is very large, reaching hundreds or thousands. The number of subsets of the set of clusters is 2^{N_c} where N_c denotes the number of clusters. For example in case of just 10 clusters the number of all subsets is 1024, which would require 4 kB of computer memory for storage of single precision

floating point representation, but for 40 clusters it reaches already 2^{40} which would require 4TB of storage. The values of BBAs can be computed on the fly when they are needed. This reduces the memory cost, but it increases the computation time dramatically.

As outlined in section 6.4.1, belief functions based on facial analysis are non-zero for only these subsets of the power set that contain just one cluster. Just non-zero BBAs need to be stored. Therefore, the storage required for facial BBAs depends linearly on the number of clusters.

The event BBAs and ownership BBAs require more storage, since all subsets of the powerset can be calculated. For a large number of clusters, the storage requirements become very difficult to meet. Therefore, some modifications must be made in order to cope with such huge data.

Reduction of clusters

One possibility for complexity reduction is limiting the number of clusters that are taken into account. Large values of BBA can be usually assigned only to a few of the closest clusters. For other clusters, the values of BBA are close to zero. Therefore, just those clusters that are closest to the considered face are analysed. These are chosen by restricting the number of considered clusters to the K nearest clusters. The value of $K = 10$ has been arbitrarily chosen as a good trade-off between processing speed and the accuracy. Actually this could be reduced further, as usually two or three of the closest clusters obtain large enough values of BBA, that there exists a little possibility that the given person belongs to a cluster with very small BBA. The number of subsets that are analysed is reduced from 2^{N_c} to 2^K ($K \ll N_c$) for every person in the collection of photographs.

Addressing subsets

Another issue that needs to be solved for implementing the TBM presented in Section 6.4.1 is generating all possible subsets. In our implementation it is resolved by creating an array of integers, each bit of elements of this array representing a cluster. As the TBM is implemented in the Java programming language, an integer contains 32 bits. Therefore the size of the array is:

$$array_size = N_c/32 + 1, \tag{6.31}$$

where N_c is the number of clusters. Each bit in the array denotes the presence or the absence of the cluster with the same index in the subset. For example, if the third bit in the array is 1, then the cluster C_3 is present in the subset. All subsets of the powerset can be easily generated by regarding the array as a N_c -bit counter and increasing value of this counter by 1 for generating each subsequent subset.

Table 6.2: Example of generating subsets of a power set 2^3 .

bit array	subset
000	\emptyset
001	$\{C_1\}$
010	$\{C_2\}$
011	$\{C_2, C_1\}$
100	$\{C_3\}$
101	$\{C_3, C_1\}$
110	$\{C_3, C_2\}$
111	$\{C_3, C_2, C_1\}$

Table 6.2 presents an example of generating all subsets of a powerset for $N_c = 3$.

In the case of a restricted number of clusters, when $K \leq 32$ the array of integers can be simplified to just a single integer.

6.4.3 Experiments

Some initial experiments were carried out with the proposed TBM for clustering similar faces in a photograph collection. These experiments showed that the model gives little improvement over the clustering with facial features only. A reason for this might be a dependency between sources, as TBM is built using the results of an initial segmentation based on face recognition features. Also, these experiments were time consuming due to the complexity of the model. The reduction method proposed in Section 6.4.2 reduces processing time significantly — from two days for 40 clusters to several minutes for any number of clusters, when the reduction to the 10 highest BBAs for each instance of a person was applied (experiments were carried out on a PC with Pentium III 450MHz processor and 256MB RAM).

6.4.4 Conclusions on Transferable Belief Model

In this section, the implementation of the Transferable Belief Model to the problem of unsupervised clustering of people in photographs is proposed by the author. In this approach, the TBM is used for combining face recognition features with event and user information. This method, however, has several disadvantages. It is a very complex, computationally expensive model. Thus it requires a lot of computational power and long processing time. Unless TBM analysis is carried out as a background process, this model is not suitable for use by a home user.

The event BBAs and user BBAs are computed using the results of initial clustering based on facial similarities. Therefore, the BBAs are correlated with facial information. As they are not independent from facial BBAs, the use of Dempster's rule of combination can be questioned.

Since the TBM described in this section is very complex and requires a lot of computational power, a novel simpler and faster approach based on restricting the number of images analysed by the clustering algorithm is proposed by the author in the next section. Event and user information is used for creating subsets for clustering and subsequent merging of clusters.

6.5 Three level clustering

Looking at the results of clustering presented in Chapter 5, one can easily observe that accuracy decreases with the increase in the size of a data set. This might be a result of an increased level of noise in a larger data set caused by outliers or just by inaccurate feature extraction. Based on this observation (that the better clustering results are obtained for a lower number of faces) a novel approach for combining event, user and facial information is proposed by the author in this section. The proposed method consists of splitting the data set into smaller subsets, conducting the clustering within these subsets and then merging similar clusters between subsets.

6.5.1 Data set levels

As information about the owners of photographs and the events at which the images were captured is available, it is used for creating meaningful and useful subsets of the data set. The collection of photographs consists of collections of several users. Thus one of the levels could be the “user” level, at which photographs captured by the same user are grouped in the same subset.

Photographs can be further divided into events, which are defined and detected within each user collection. These events are detected using the technique proposed in [16] (see Section 6.3). Therefore, there are three levels at which a collection of photographs can be split: the most detailed level is the “event” level at which a small number of images is assigned to a given event; next is the “user” level gathering photographs from all events captured by a given user. This level is more coarse and provides fewer larger groups of photographs. The most coarse level is a “collection” level containing all photographs in the dataset. These levels are graphically presented in Figure 6.3.

6.5.2 Event level

In the approach proposed here by the author, the unsupervised clustering is carried out at the event level as many times as there are events. Each time only those photographs are used that were captured at the given event. The technique described in Section 5.3.3 is employed by the author for those clustering processes.

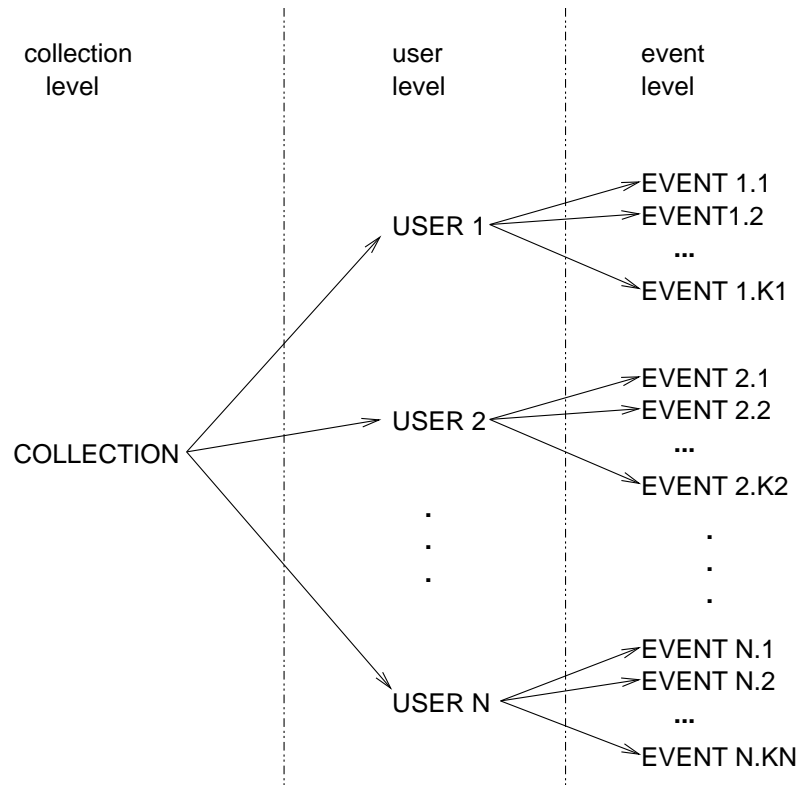


Figure 6.3: Three levels at which photograph dataset can be divided.

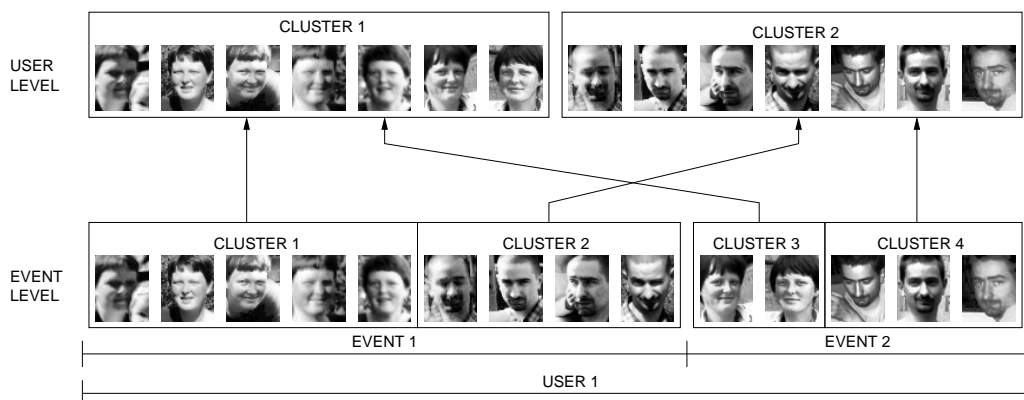


Figure 6.4: Merging at user level.

This produces a large number of small clusters, containing only images captured at the same event. Therefore, the resulting clusters consist mainly of appearances of the same identity. However, this identity may appear in many clusters. Thus finding all images containing a given person would require retrieving many of the clusters. This problem can be overcome by merging (e.g. at user level) similar clusters created at the event level.

The restriction for merging faces only from the same event can be seen as a special case of the fusion using the product rule. Let us consider that the nearest neighbour is defined as a point with the highest value of similarity measure. Let us define the facial similarity measure s_f as a function of a distance in face recognition features space $d(f_i, f_j)$ between two points f_i and f_j :

$$s_f(f_i, f_j) = \exp(-\lambda d(f_i, f_j)), \quad (6.32)$$

or, if $0 \leq d(f_i, f_j) \leq 1$

$$s_f(f_i, f_j) = 1 - d(f_i, f_j). \quad (6.33)$$

Also let us define a similarity s_e in regard to events:

$$s_e(f_i, f_j) = \begin{cases} 1 & \text{if } f_i \text{ and } f_j \text{ were captured at the same event} \\ 0 & \text{if } f_i \text{ and } f_j \text{ were captured at different events} \end{cases} \quad (6.34)$$

Then the similarity between two points is expressed as a product of similarity in the face recognition feature space and similarity in regard to events:

$$s(f_i, f_j) = s_f(f_i, f_j)s_e(f_i, f_j). \quad (6.35)$$

Then the nearest neighbour clustering can be based on this similarity measure defining the nearest point as a point of highest value of the similarity $s(f_i, f_j)$.

6.5.3 Merging clusters

In the technique proposed by the author, the clusters at the user and collection levels are merged only if the similarity, measured in terms of their location in the subspace and their structure, is above a given threshold. The author uses the normalised distance, the same as used for experiments in section 5.3.1 as a distance measure between two clusters. This measure takes into account the locations of cluster centres and the spread of points around the centre of each cluster. Any distance measure such as Euclidean or cosine distance can be used for calculating distances between points and centres of clusters. In our case, as the MPEG-7 FR descriptor is used, the distance proposed for this descriptor [24] is used (see Appendix B).

The two distances calculated between two clusters C_i and C_j are:

$$d(C_i, C_j) = \frac{d_p(m_i, m_j)}{\sigma_j}, \quad d(C_j, C_i) = \frac{d_p(m_j, m_i)}{\sigma_i}, \quad (6.36)$$

where d_p is a distance measure for a given feature space, m_i is a centre of gravity of the C_i cluster, σ_i is the standard deviation of the distances between points and the centre of the cluster. The clusters are merged if both distances are below a certain threshold, the same for each distance value $d(C_i, C_j)$ and $d(C_j, C_i)$. This ensures that both centres of merged clusters are located within the same multiple of the standard deviations from each other.

6.5.4 Evaluation

Single levels

Event level The experiments for testing the improvement given by restricting clustering only to events, were carried out first. Facial images were grouped firstly into groups containing images captured at the same event. Then the clustering process based on the nearest neighbour technique (see Section 5.3.3) was carried out on every (event) group. Figure 6.5 presents the flow chart of the system for clustering within events.

The results of clustering within events are presented in Table 6.3. The same table also shows, for comparison, the results of clustering across events and user collections. It can be easily seen that splitting the dataset into groups at each event and carrying out clustering within such groups increases both precision and recall. The number of created clusters is higher, due to the initial pre-grouping into events.

User level Clustering within events forces images of the same identity to be classified to different clusters in situations where a given identity was captured at more than one event. It is, therefore, interesting to see how restricting clustering to user collections improves (or does not improve) the clustering performance. The experiments were carried out by clustering facial images only within user collections. Figure 6.6 presents the diagram of the system used and Table 6.3 shows the results of the experiments.

As can be seen in the results table, clustering within user collections improves precision over clustering across the whole collection. However, the increase is not as large as in the case of the event level clustering. This is not surprising as the number of images in each group is not as strongly restricted as it is in clustering within events. However, the numbers of valid clusters are comparable in both cases. The number of all created clusters is much higher in the case of clustering

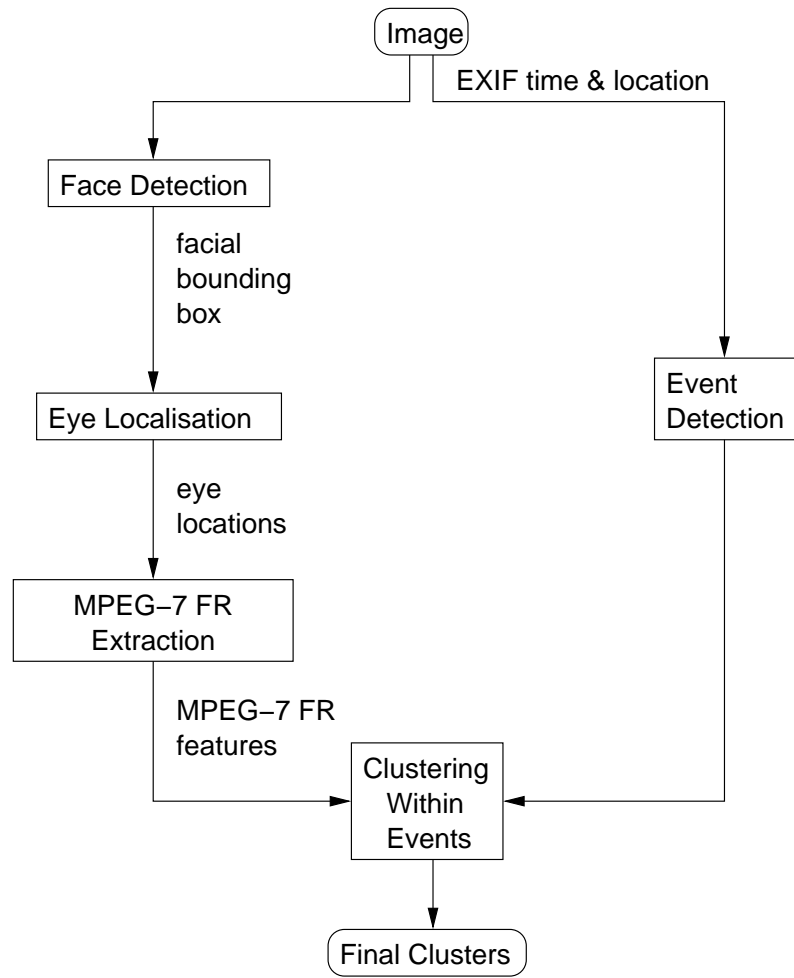


Figure 6.5: Diagram of the system for clustering similar faces within events.

within events. This might indicate that a strong restriction put on the number of images (so that only photographs from the same event are analysed together) might help in separating outliers.

Two levels - event and user

Experiments were also carried out on two-level clustering, consisting of event and user level. The nearest neighbour method was used for clustering at event level. At the user level similar clusters were merged as described in section 6.5.3. Several values of threshold were investigated. Results are presented in Figures 6.8 and 6.9 in two scenarios: with manually located eyes and with automatically located eyes.

As one could expect, values of mean precision decrease with the increase of the threshold, while values of recall increase in general. Also the number of created clusters is lower where the threshold is higher, which is not surprising. The increase in values of recall is not large in comparison to clustering within events. The use of a possibly inappropriate merging technique could explain this effect. However, it is also possible that events affect the appearance so much that

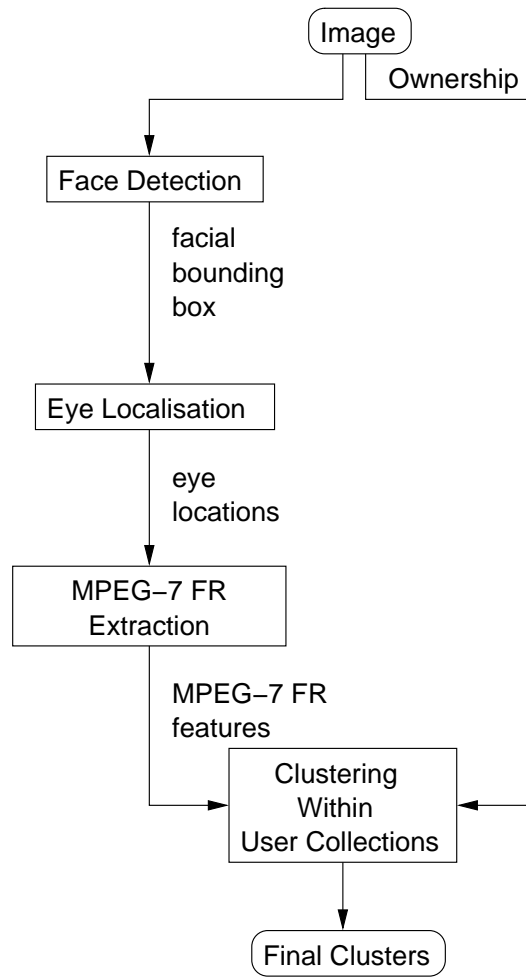


Figure 6.6: Diagram of the system for clustering similar faces within users collections.

the appearance of a human face is event specific, which makes merging between events difficult.

The value of threshold that gives a close to optimal number of clusters, is 0.08 for manual eye locations and 0.06 for automated eye locations. In the former case it gives 53 valid clusters with average precision 0.696 and average recall 0.224. When eye locations are extracted automatically this method gives, at the threshold 0.06, 51 valid clusters, with average precision 0.633 and recall 0.236.

Small values of the threshold indicate that cluster centres are placed very close to each other and clusters probably overlap (as the threshold is a fraction of the value of the spread).

If these results are compared to clustering within user collections (Table 6.3) one can easily see an improvement in both precision and recall. However, the improvement in recall values over experiments of clustering within events is not large.

Table 6.3: Results of clustering at single levels.

level	features	no of all clusters	no of valid clusters	precision	recall
event	eyes manual	402	89	0.791	0.207
event	eyes auto	386	74	0.750	0.211
user	eyes manual	229	87	0.640	0.206
user	eyes auto	230	78	0.581	0.195
collection	eyes manual	201	79	0.548	0.152
collection	eyes auto	187	59	0.516	0.178

Two levels - event and whole collection

Experiments were also conducted in a scenario where clusters created within events were merged within the whole collection, across all user collections. Results are presented in Figures 6.11 and 6.12.

In this case there is a large increase in precision at threshold levels of 0.09 for scenarios with both manually and automatically located eyes. However, at these points the number of valid clusters is unacceptably low. The optimal number of valid clusters is obtained for values of threshold at 0.06 and 0.04 for features obtained with manually and automatically located eyes respectively. Precision values are 0.853 and 0.823, recall values are 0.203 and 0.189 respectively. Interestingly, the values of recall at the lower values of threshold decrease with the increase in threshold values, which is unexpected.

Three level - events, users, collection

The experiments on three level clustering were carried out at event, user and collection level. Results are presented in Figures 6.14 and 6.15. Figure 6.13 shows the diagram of the system used for experiments.

Firstly clustering based on a nearest neighbour classifier was carried out within events. Then merging of similar clusters was carried out within each user collection, which was followed by merging between user collections. Because clusters were merged in two stages, two values for threshold were needed.

Let us denote by t_e the threshold that is used for merging clusters across events within user collections. Let t_u denote the threshold used for merging clusters (obtained with merging within user collections) across user collections. Charts presented in Figures 6.14 and 6.15 show the results of clustering and merging with several values of thresholds. When the facial features were extracted using manually located eyes, the optimal or nearly optimal number of valid clusters was obtained for a few different thresholds. Best results were obtained for $t_e = 0.5$ and $t_u = 0.4$, when 54 valid clusters were obtained giving a precision of 0.904 and a recall of 0.210. This is great improvement in precision. There is no improvement

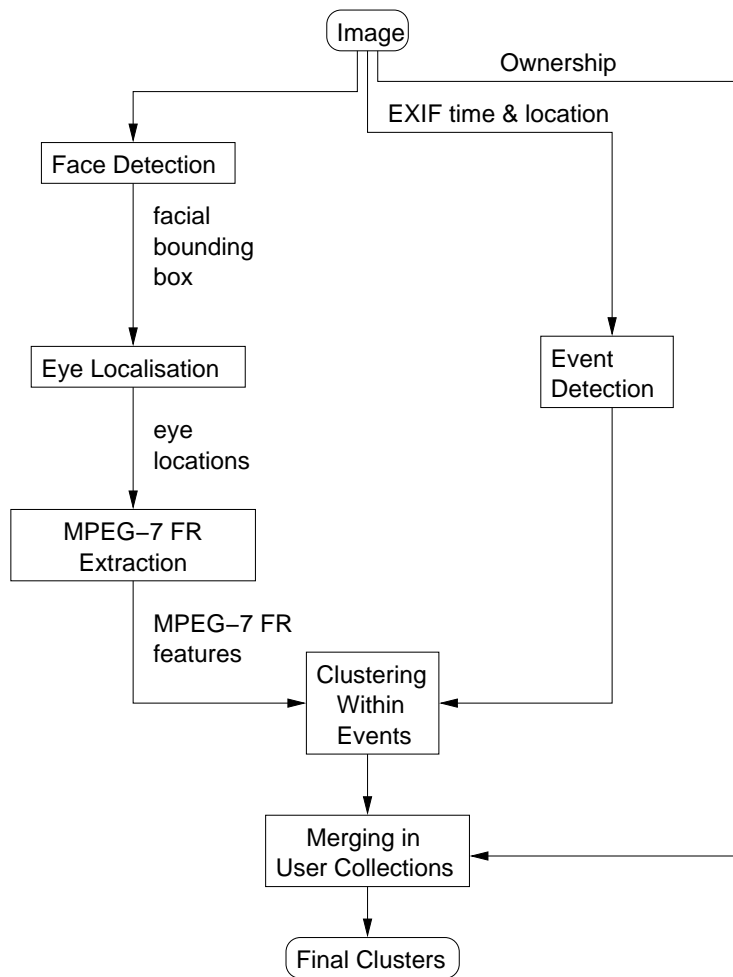


Figure 6.7: Diagram of the system for two-level clustering of similar faces within events and users collections.

in recall, however. This means that these valid clusters contain just a fraction of all images containing known identities and roughly just 1/5 of interesting photographs can be retrieved (with quite high average precision, though).

When the facial features obtained with automated eye locations were used, the values of threshold at which the optimal number of valid clusters was reached, were lower. For $t_e = 0.03$ and $t_u = 0.02$ the precision is 0.852 and recall 0.180. But if $t_e = 0.01$ and $t_u = 0.04$ were used, the number of valid clusters remained the same, giving, however, precision becomes 0.831 and recall becomes 0.195.

A general tendency can be observed that higher precision values are obtained for higher values of t_e and lower t_u , while higher recall values are achieved for lower values of t_e and higher t_u . This means that in the collection used for experiments, several users might had made photographs of the same events or at least the same group of people. Then some identities occur in several user collections and probably were captured at the same events. Therefore, their appearance is very similar in spite of being in another user's collection. In such situation more flexible merging (with higher threshold) across user collections captures more oc-

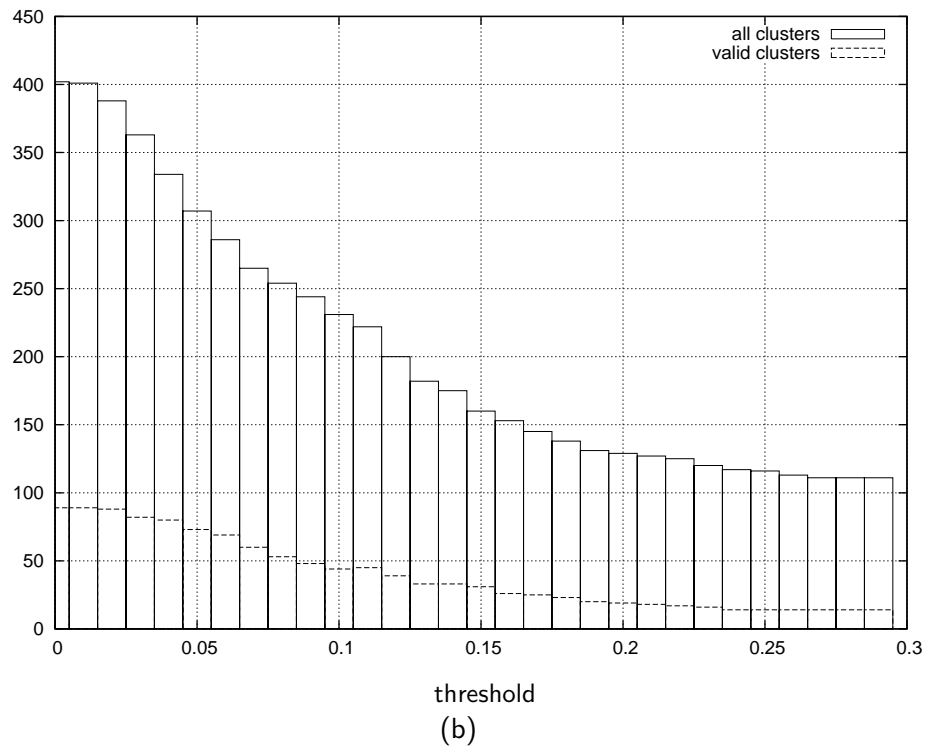
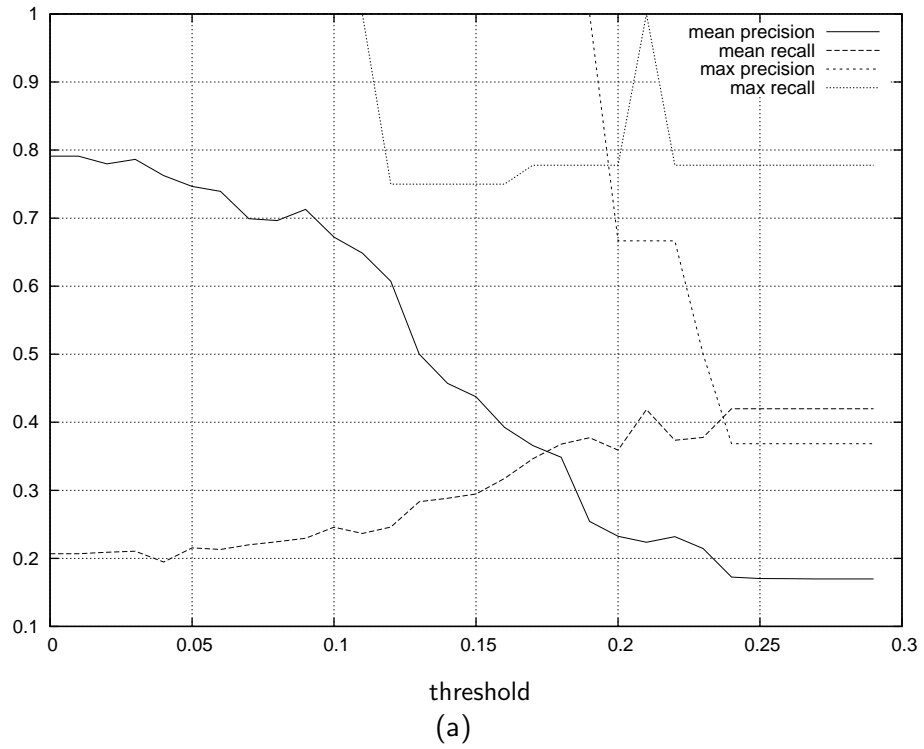
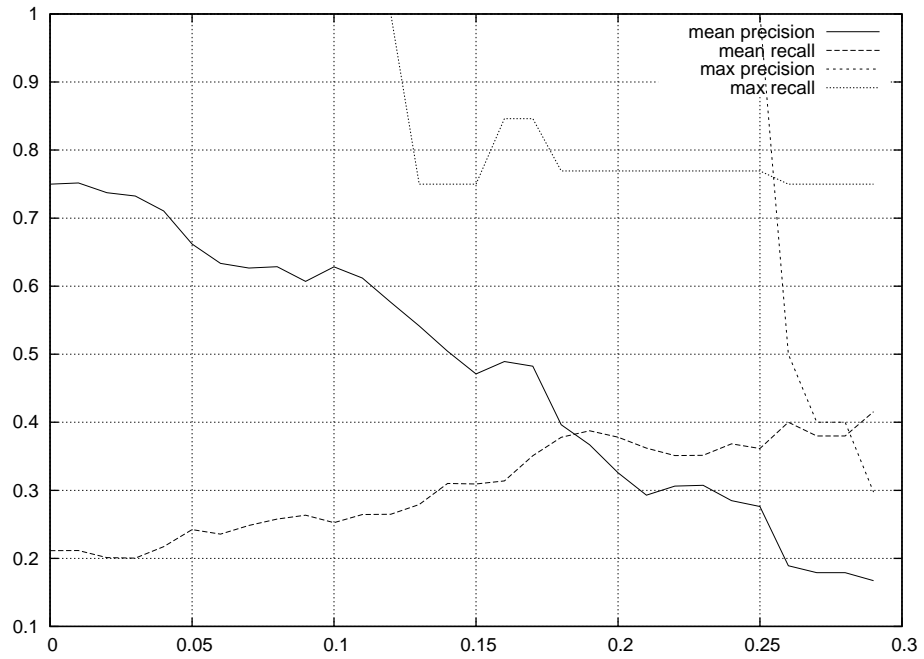
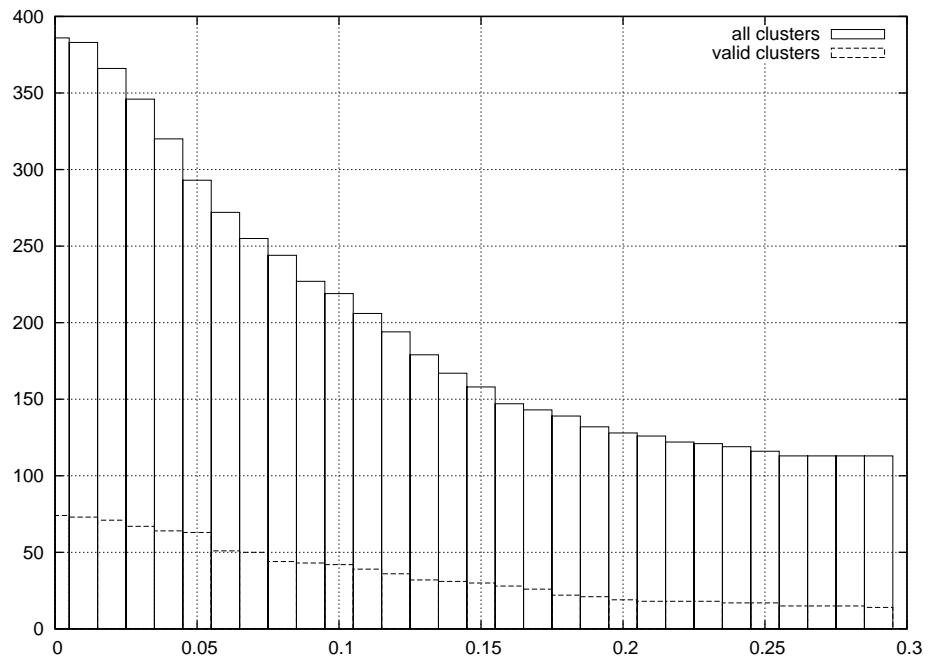


Figure 6.8: Precision and recall (a) and number of clusters (b) for different values of threshold for merging clusters and collection level; manual eye locations.



(a)



(b)

Figure 6.9: Precision and recall (a) and number of clusters (b) for different values of threshold for merging clusters and collection level; automated eye locations.

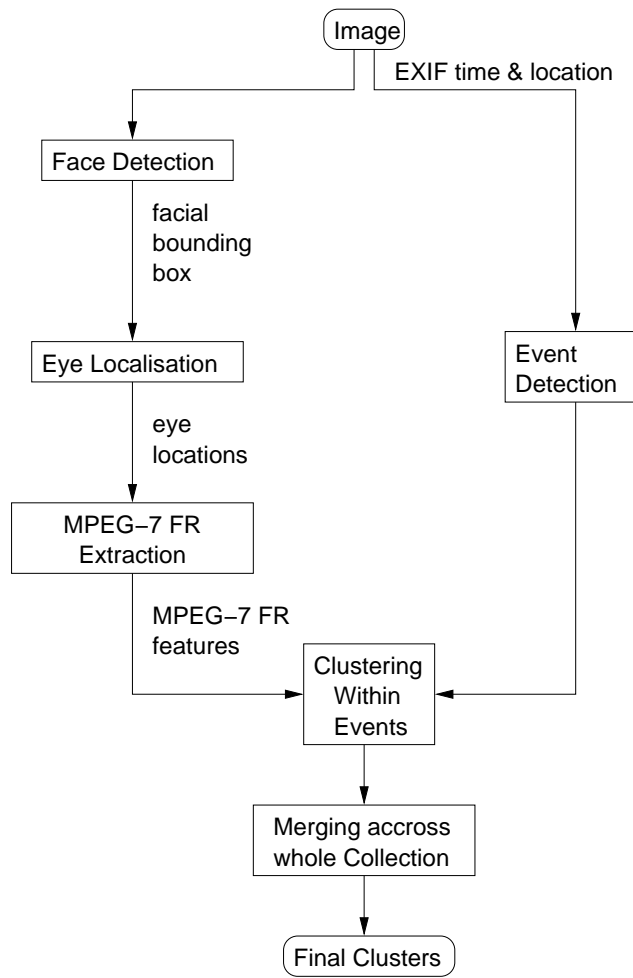


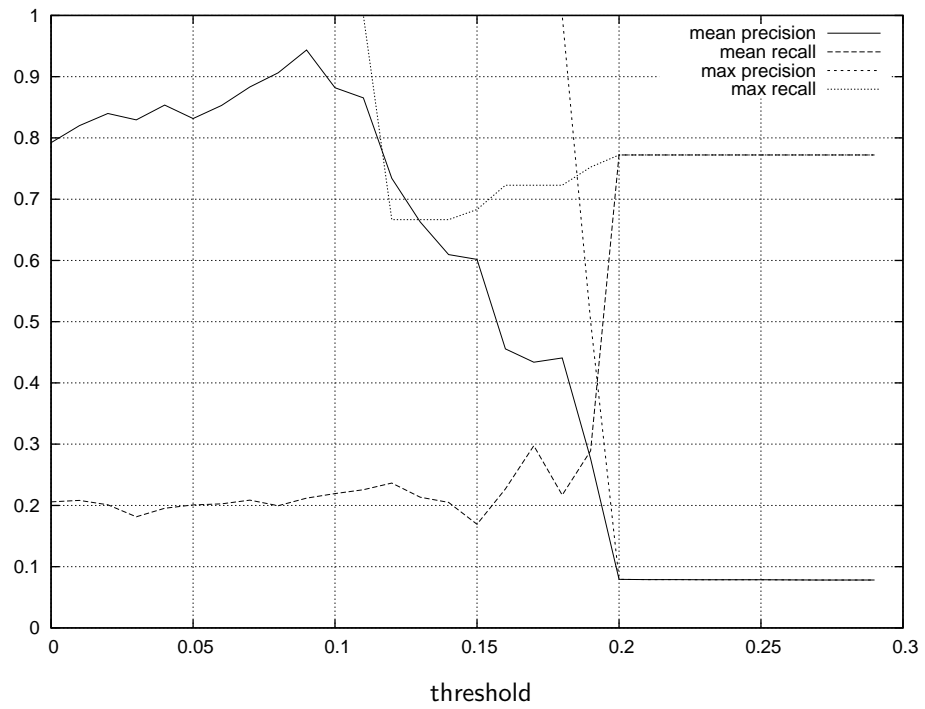
Figure 6.10: Diagram of the system for two-level clustering of similar faces within events and the whole collection.

currences of a given identity in the collection, resulting in higher recall. However, this decreases precision because more outliers can be captured.

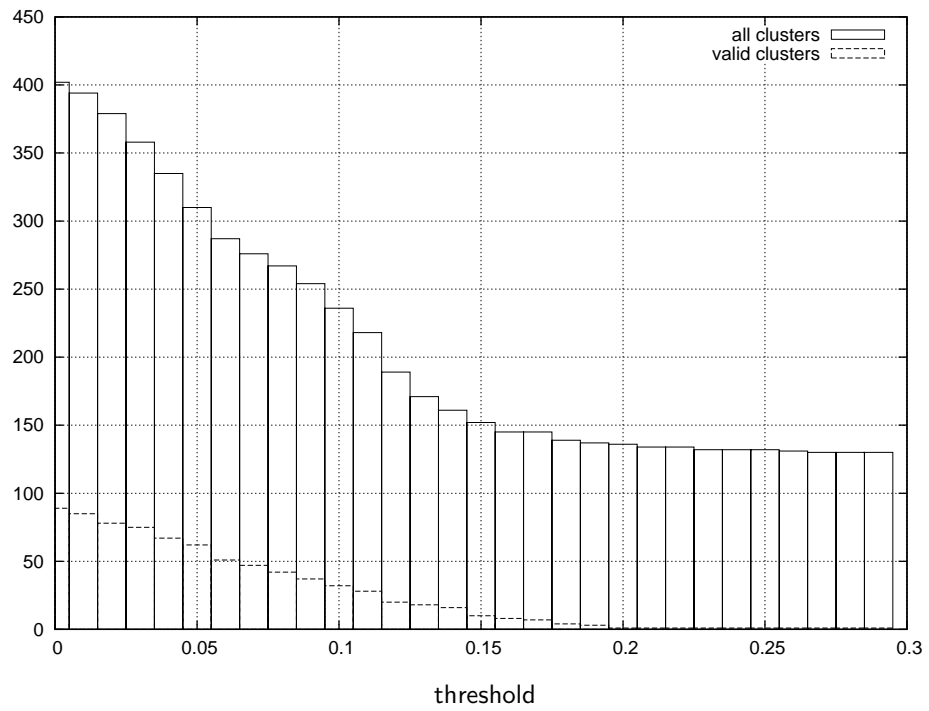
6.6 Combination of body patch with facial analysis

6.6.1 Body patch

The body area of the person in an image can be used for finding a re-occurrence of the person in the collection [21]. It is based on an observation that people rarely change their clothes during an event. Therefore, the analysis of the colour and pattern of a body region in an image can give a good hint to match the re-occurrences of the corresponding person in images. Colour and pattern are very simple, and robust features. Thus they are very efficient for matching a person across images. One must remember, however, that accurate results can be expected only among the images which were taken in situations when the person

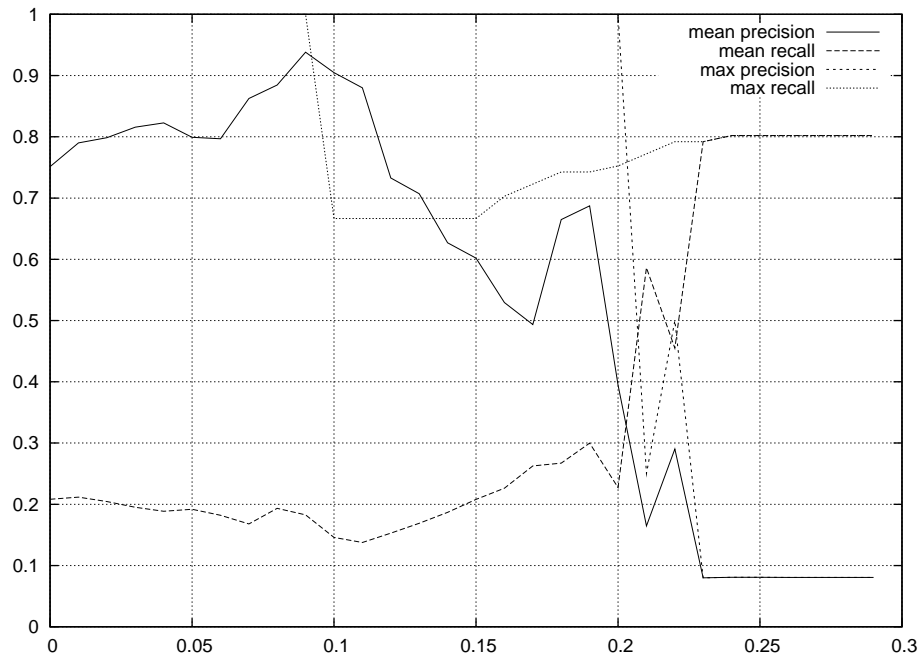


(a)

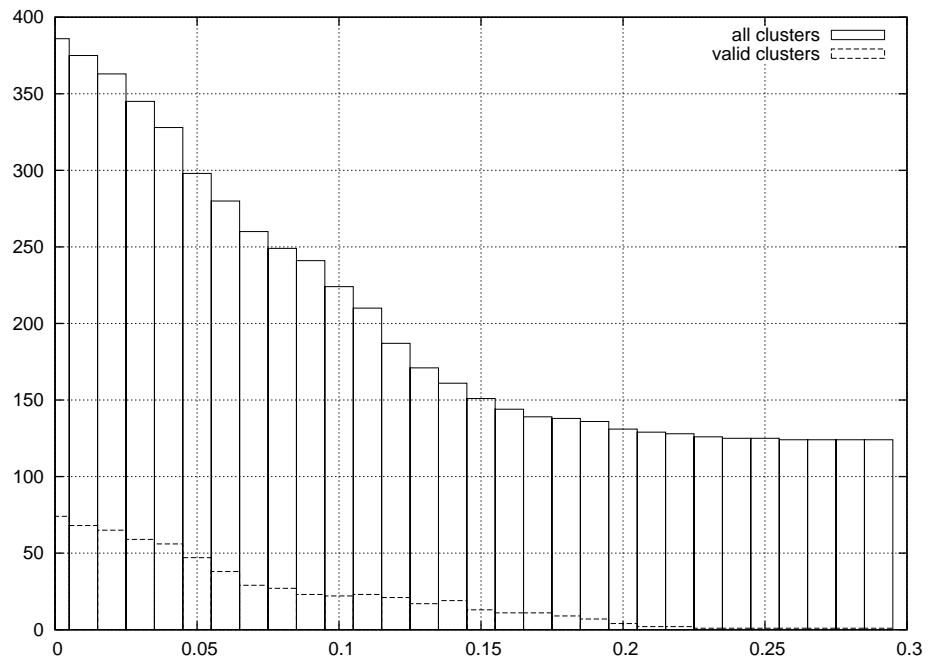


(b)

Figure 6.11: Precision and recall and numbers of clusters produced at two level clustering on event and collection level, as a function of the merging threshold (for manually located eyes).



threshold
(a)



threshold
(b)

Figure 6.12: Precision and recall and numbers of clusters produced at two level clustering on event and collection level, as a function of the merging threshold (for automatically located eyes).

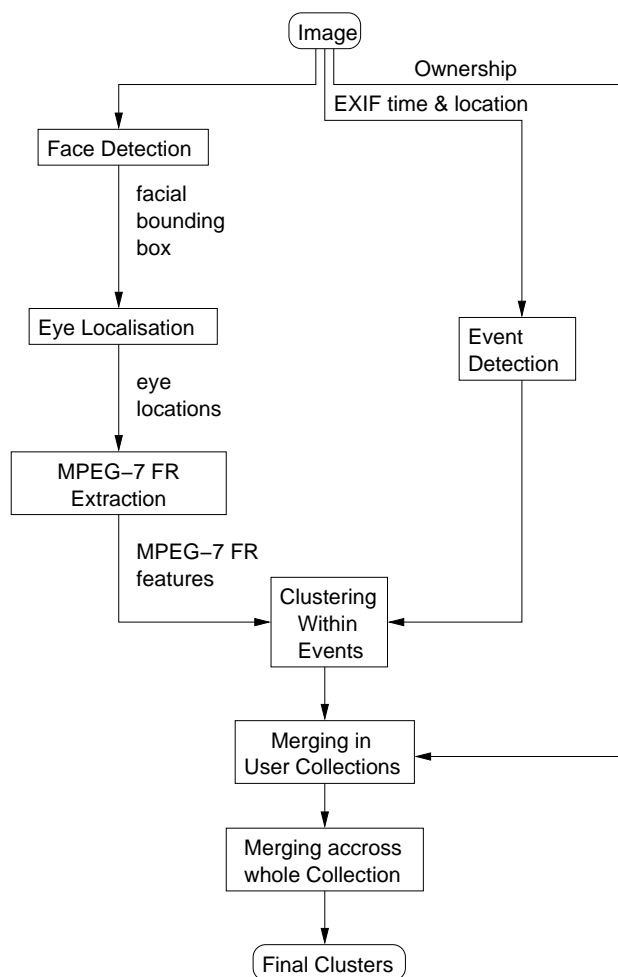


Figure 6.13: Diagram of the system for three-level clustering of similar faces: within events and at user and collection levels.

was wearing the same clothes e.g. taken at the same event.

Additional difficulties can be encountered when the body area is not available. This can happen due to occlusions, torso outside the frame of a photograph, or torso placed in an image region other than predicted. Therefore, using only body information seems to be not sufficient for indexing the whole collection.

6.6.2 Combination

As the body patch technique is efficient in clustering within events, it is reasonable to combine this feature with facial analysis features. This should increase accuracy of clustering within events, leaving indexing across events to facial analysis only.

Three techniques for combining body information with eye analysis were used:

- weighted sum

$$p = \alpha p_f + (1 - \alpha) p_b \quad (6.37)$$

- weighted product

$$p = p_f^\alpha p_b^{1/\alpha} \quad (6.38)$$

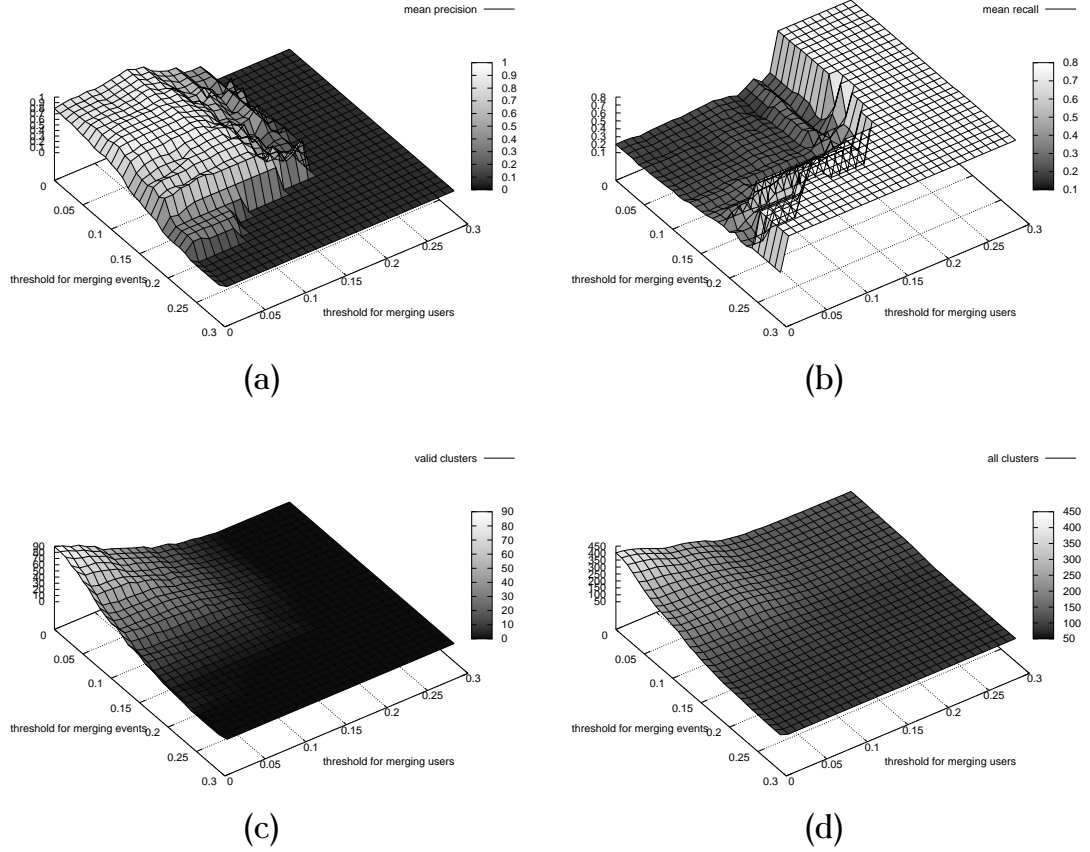


Figure 6.14: Results of three level clustering for different values of thresholds (manual eye locations): (a) mean precision, (b) mean recall, (c) number of created clusters, (d) number of valid clusters.

- weighted maximum

$$p = \max(\alpha p_f, (1 - \alpha) p_b) \quad (6.39)$$

where p_f is a similarity measure of eye features and p_b is a similarity measure for body features, and both these values can be seen as probabilities. A weight α indicates importance of each of the information sources. In the case of a weighted sum and weighted maximum, $\alpha = 0$ results in only the body analysis being used. For $\alpha = 1$ only facial analysis is employed. These are simple probabilistic approaches to information fusion. However, they are usually sufficient.

A more sophisticated approach would be a Transferable Belief Model. Let us assume that a belief that a person given in an image belongs to the cluster C_i (N clusters in total, $i = 1, 2, \dots, N$) given by eye analysis is $m_f(C_i)$, and that a similar belief given by body analysis is $m_b(C_i)$. Then using the normalised version of Dempster's rule of combination [110] we obtain:

$$m_{fb}(C_i) = m_f(C_i) \oplus m_b(C_i) \quad (6.40)$$

$$= \frac{\sum_{C_k \cap C_l = C_i} m_f(C_k) m_b(C_l)}{1 - \sum_{C_k \cap C_l = \emptyset} m_f(C_k) m_b(C_l)}, k = 1, \dots, N, l = 1, \dots, N \quad (6.41)$$

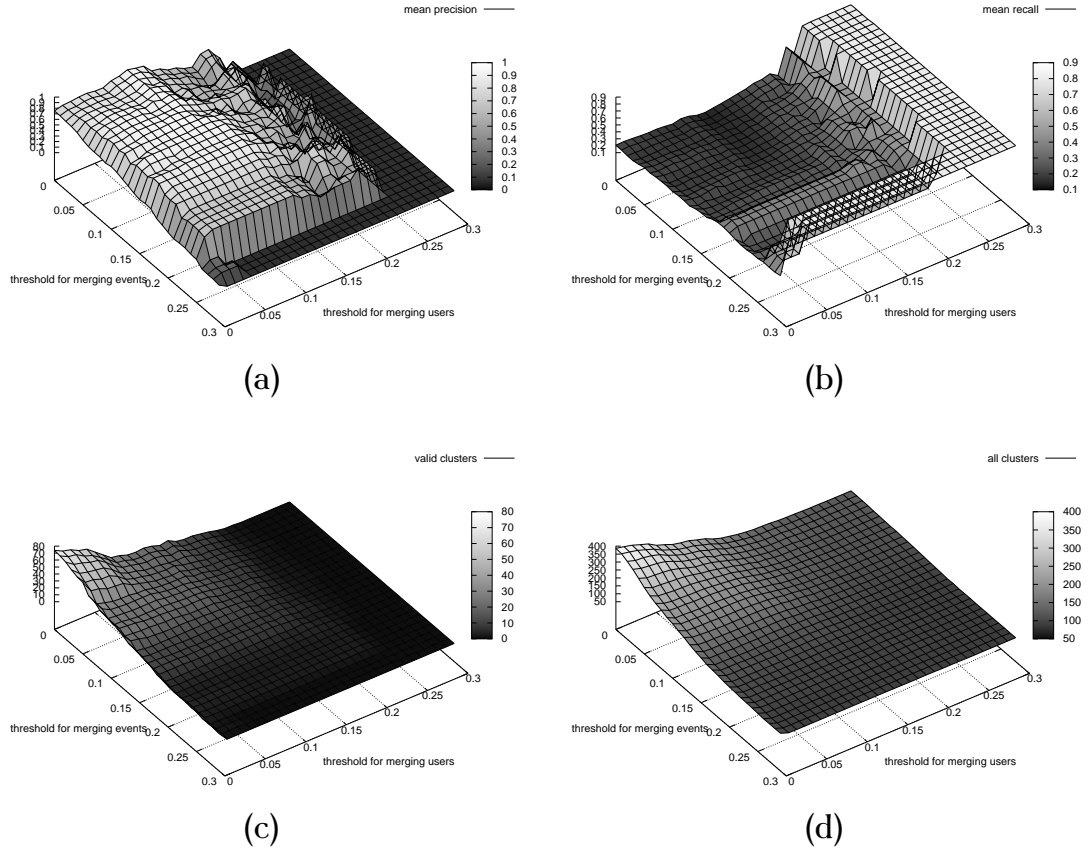


Figure 6.15: Results of three level clustering for different values of thresholds (automated eye locations): (a) mean precision, (b) mean recall, (c) number of created clusters, (d) number of valid clusters.

$$= \frac{m_f(C_i)m_b(C_i)}{\sum_j m_f(C_j)m_b(C_j)} \quad (6.42)$$

bearing in mind that if we obtain masses of belief m_f and m_b as functions of similarity measures, these masses are determined for one element subsets of a set of clusters (i.e. we can obtain $m_f(C_i)$ and $m_b(C_i)$, but not $m_f(C_i, C_j, \dots)$ or $m_b(C_i, C_j, \dots)$). In the case of clustering, where we compare beliefs, the value of the denominator in Equation 6.42 is the same for each cluster and can be ignored. Thus the resulting combination is the same as the weighted product with $\alpha = 1$, and no improvement over the weighted product method can be expected.

6.6.3 Experiments

Data set

A detailed description of the data set used for the experiments can be found in Appendix C.2. The set that was used in the experiments is the set containing 1127 instances of a human. The MPEG-7 FR features were extracted using manual locations of eyes. The body patch features (MPEG-7 SC) were obtained using the

technique described in [21]. The dataset is also pre-organised into events with the method presented in [16].

Combining within events

The experiments were carried out in two scenarios. In the first one, facial features are combined with body features within events. In the second scenario, features are combined within user collections. The first scenario follows the observation that people rarely change their clothes during an event. Figure 6.16 presents a flow chart of the system used for experiments in scenario 1.

Figures 6.18, 6.19, 6.20 present precision and recall values obtained for clustering with different combination methods for various values of α in the first scenario. Lower values of α indicate that more importance is put on body features, higher α makes the clustering depend more on facial information. In the cases of the average sum and maximum rule, extremes are reached for $\alpha = 0$ and $\alpha = 1$. In the former case clustering depends only on body features, in the later one just on facial features.

When the average sum is used, results of clustering using combined features are lower in both precision and recall than for body features alone (Figure 6.18). Additionally, the generated number of clusters increases, which is not good, as the number of clusters is already too high for single features. There is no gain in using this method of fusion.

The results of combining features using the product rule are shown in Figure 6.19. There is a small improvement in precision at $\alpha = 0.7$. However, at the same point recall decreases. Another point of improvement is for $\alpha = 3$ and in this point both precision and recall are increased. This is not a large increase, however.

Combination of facial and body features using the maximum rule gives the results presented in Figure 6.20. There is a peak in values of precision for $\alpha = 0.41$, but this do not correspond with a peak in recall values. Therefore, the peak in precision might have been obtained accidentally.

Combining within users' collections

The flow chart of the system used for experiments within users collections is shown in Figure 6.17. Figures 6.21, 6.22, 6.23 present precision and recall values obtained in these experiments. As expected, results are not as good as when clustering was carried out within events. However, characteristics are surprisingly similar to ones discussed in the previous section. The body patch features work much better across events within a user collection than face recognition features, in spite of an expected change in clothing. It looks like the variety of clothes is so high, that different people rarely wear clothes similar enough to significantly deteriorate the results of clustering based only on body patch analysis. This simple feature

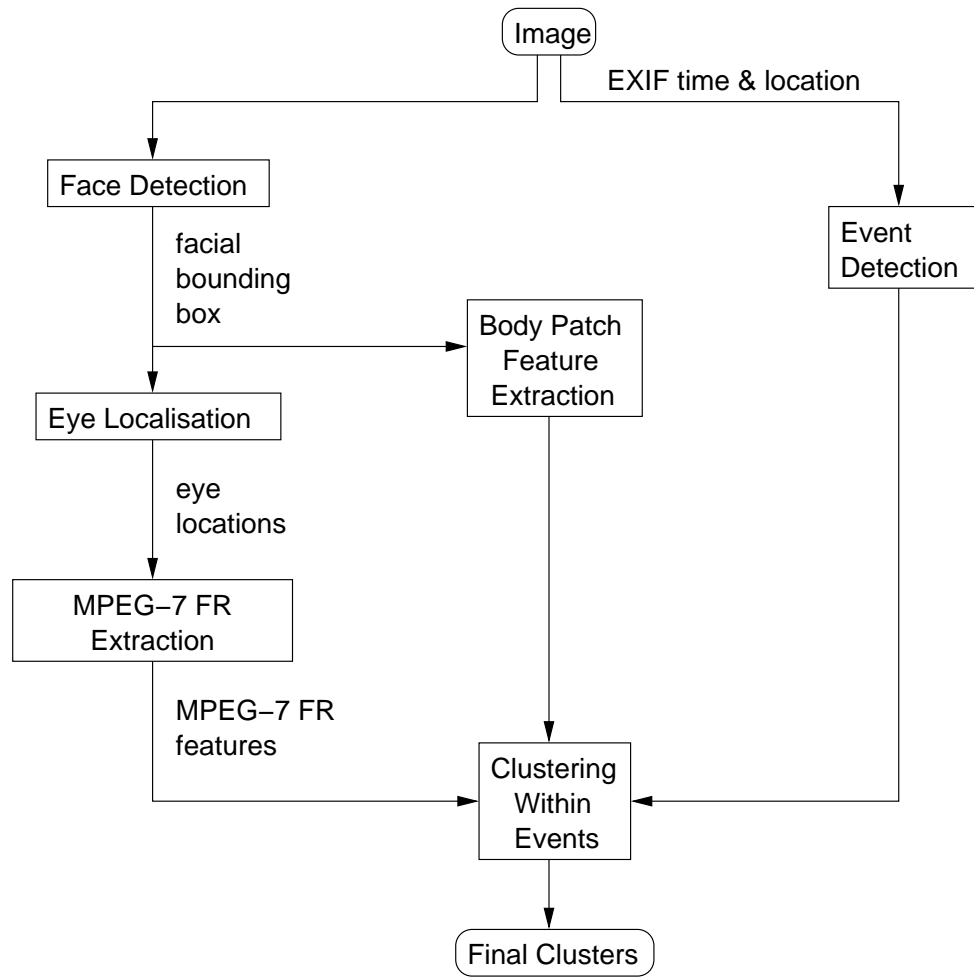


Figure 6.16: Diagram of the system for combining body patch with facial features and events.

captures the structure of events well even when information about the event is not available, i.e. the clusters created consist mostly of images from the same event.

Results obtained with the average sum rule are not different from the ones obtained in scenario one, just that the changes in values are larger. There is a minor difference — a peak in recall values which corresponds to a local peak in precision.

For the product rule, results are different from scenario 1. There is large decrease in precision value when both facial and body features are combined. There is, however, a high peak in recall at $\alpha = 1.5$. For this value of α , the facial features influence the clustering a little bit more than the body features. The peak in recall can be an effect of a high proximity between the body features of the same identity in events and the facial features outside events.

Very interesting results were obtained using the maximum rule. Both precision and recall have got peaks for $\alpha = 0.41$. This corresponds with a peak in precision for the same value of α as in scenario 1, and a local peak in recall for $\alpha = 0.40$ in scenario 1. Therefore, the maximum rule is most suitable (among these three

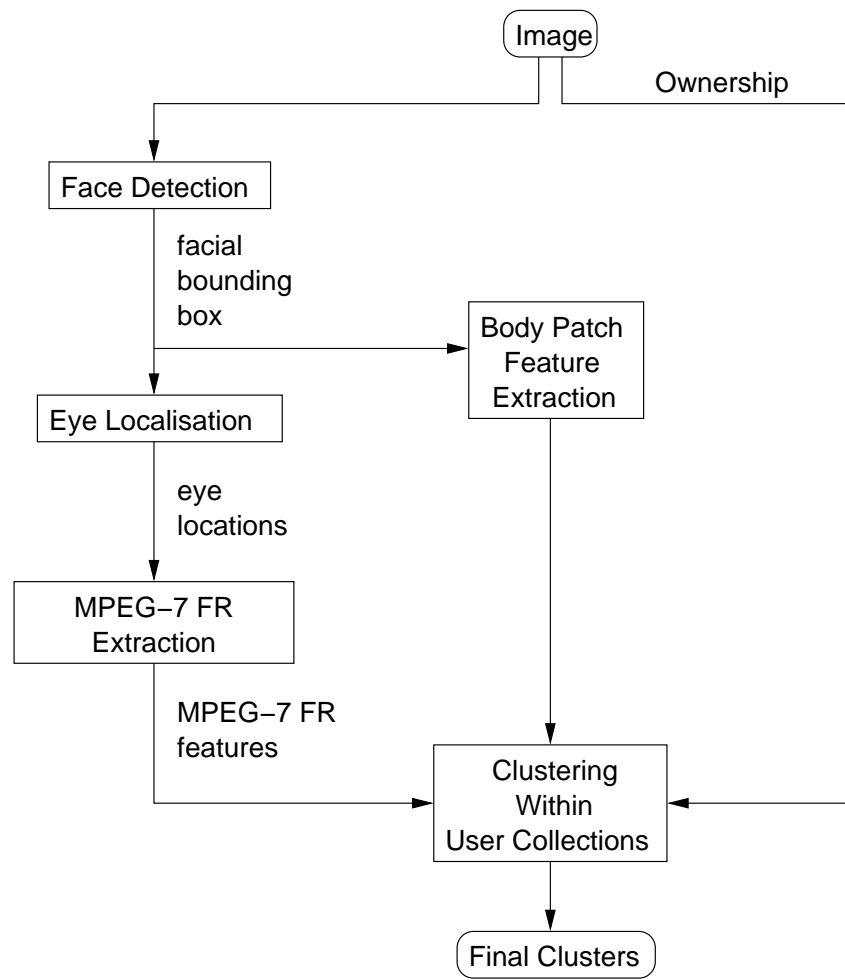
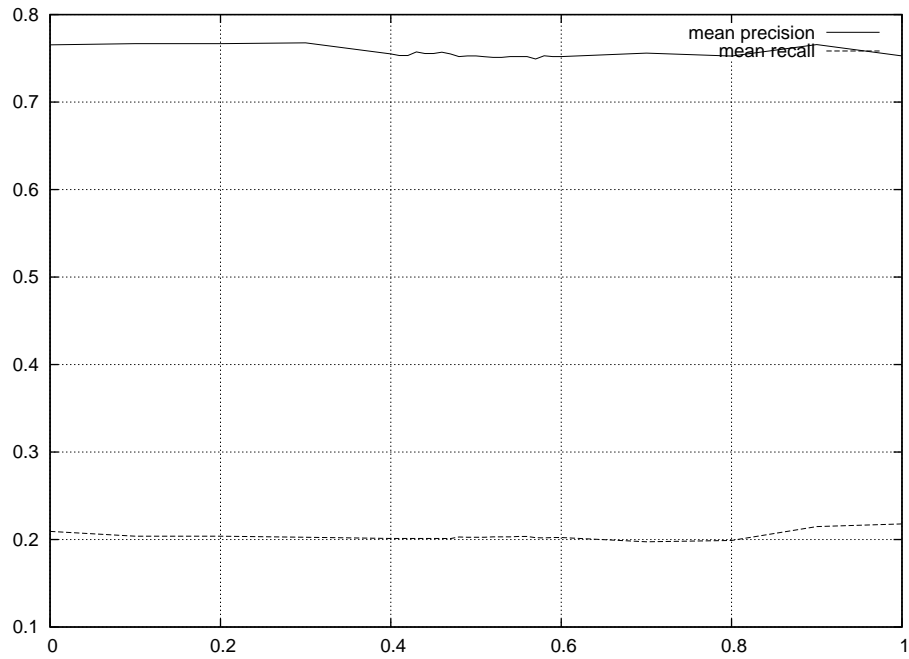
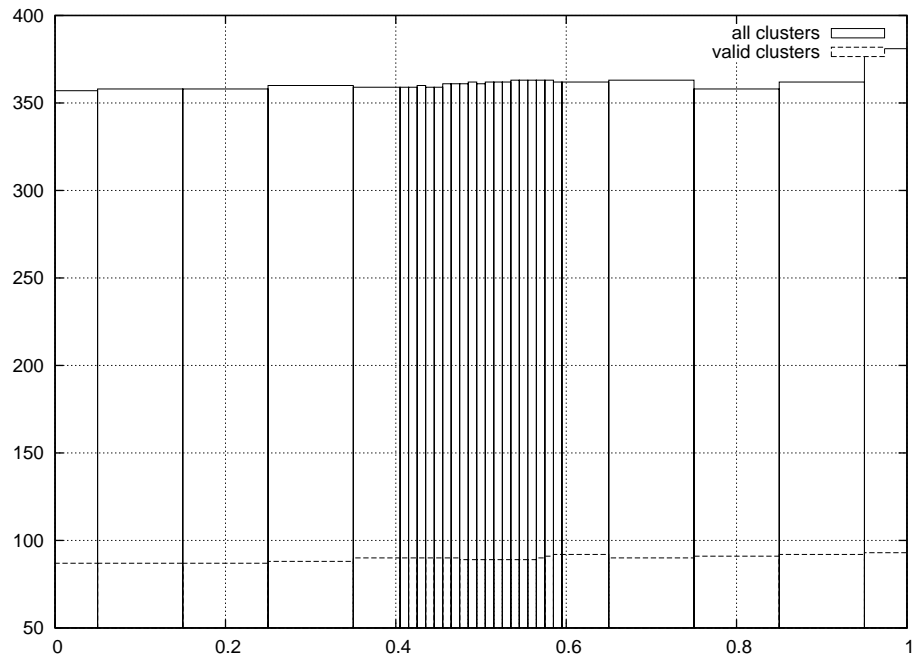


Figure 6.17: Diagram of the system for combining body patch with facial features and user information.

probabilistic rules investigated) for combining facial features with body features, as it gives the highest gain in the precision of clustering, and it corresponds with an increase of recall values.

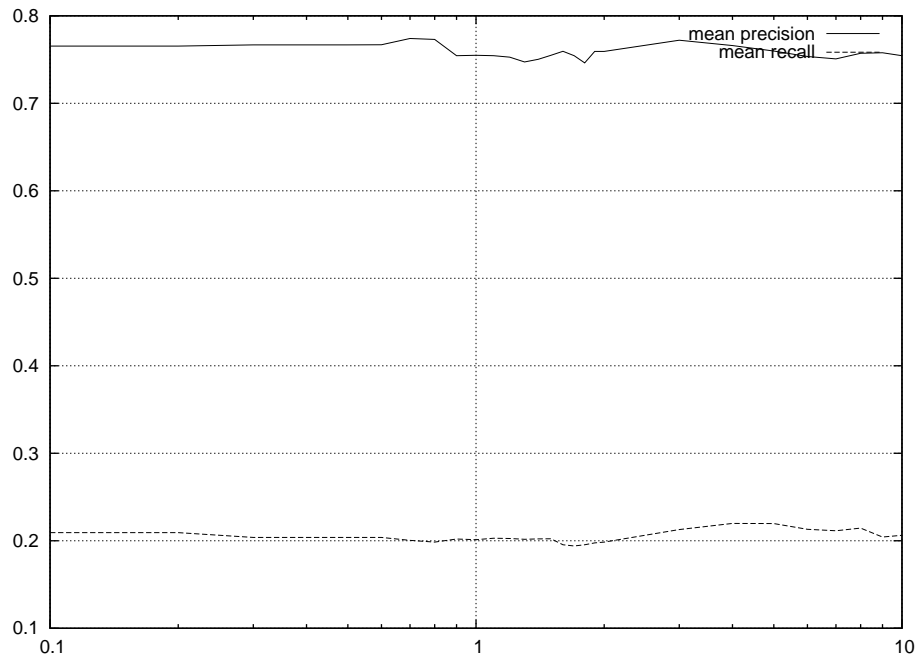


(a)

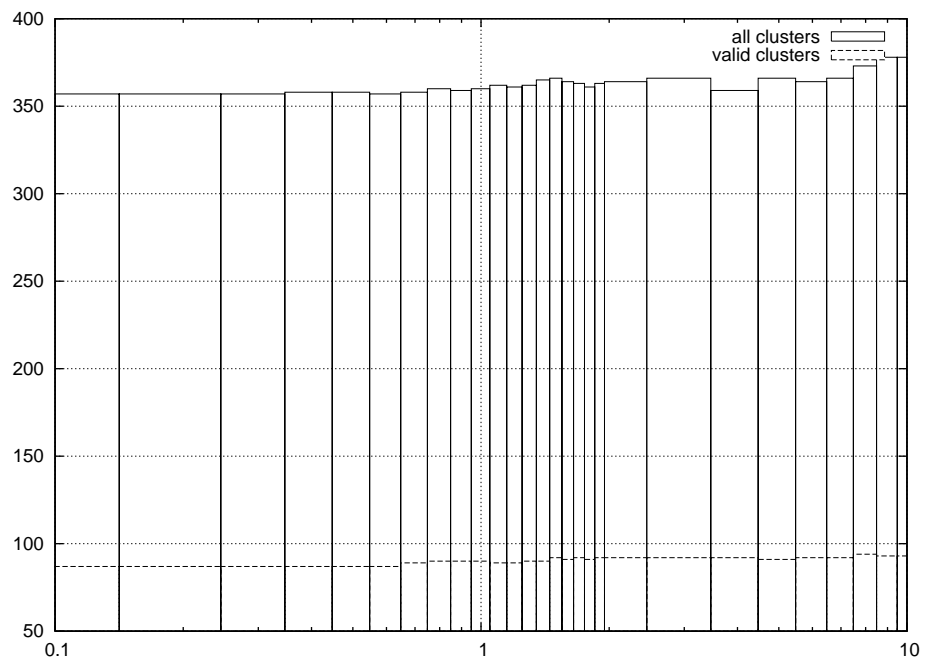


(b)

Figure 6.18: Results for combining facial and body features within events using average distance value $p = \alpha p_f + (1 - \alpha)p_b$.

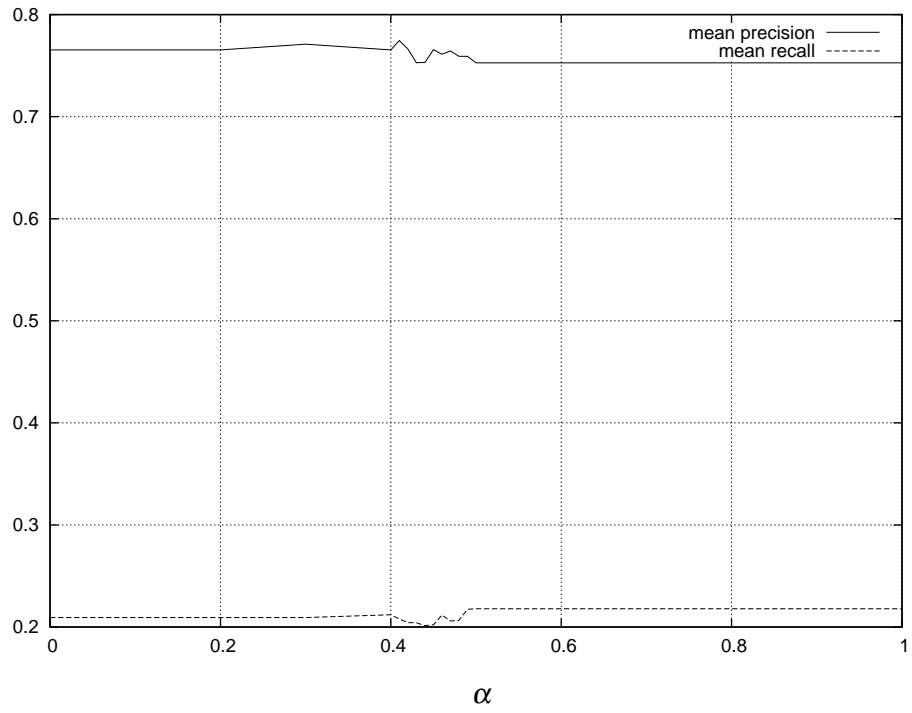


(a)

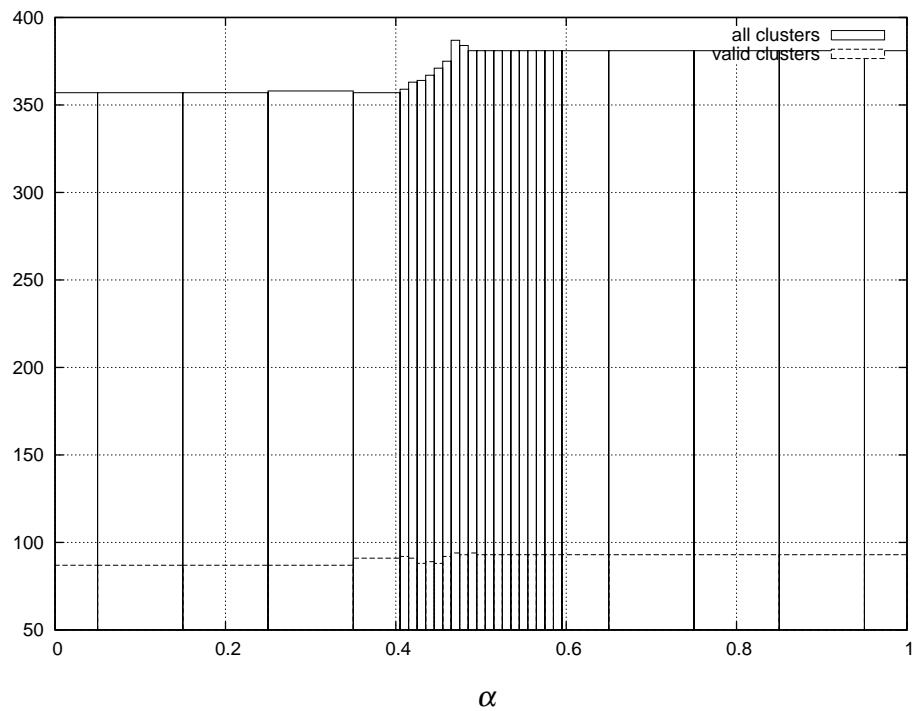


(b)

Figure 6.19: Results for combining facial and body features within events using product of probabilities $p = p_f^\alpha \cdot p_b^{1/\alpha}$.



(a)



(b)

Figure 6.20: Results for combining facial and body features within events using maximum $p = \max(\alpha p_f, (1 - \alpha) p_b)$.

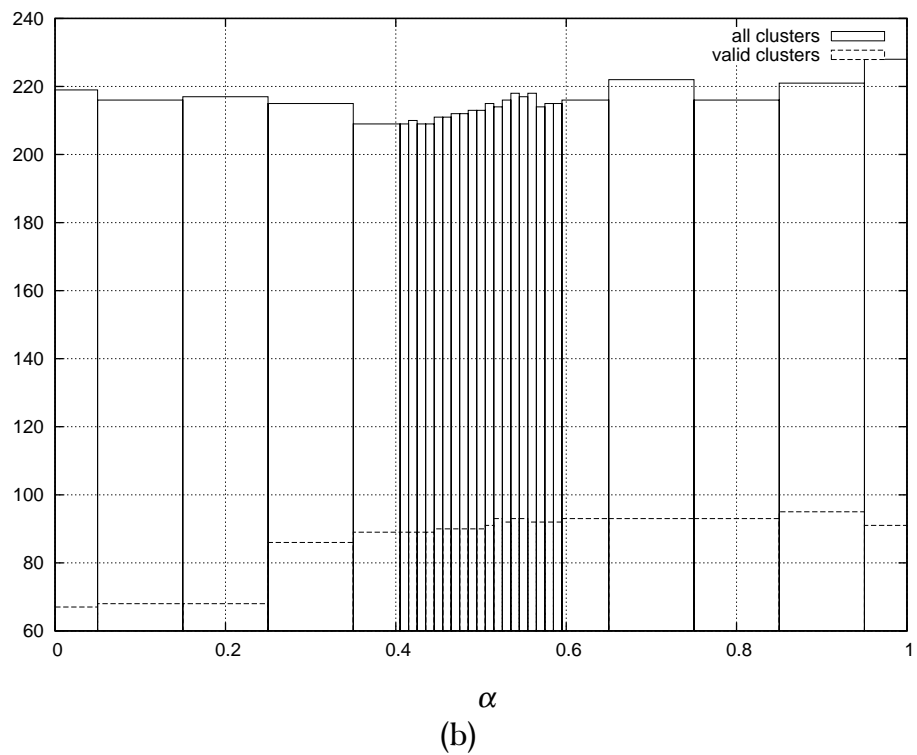
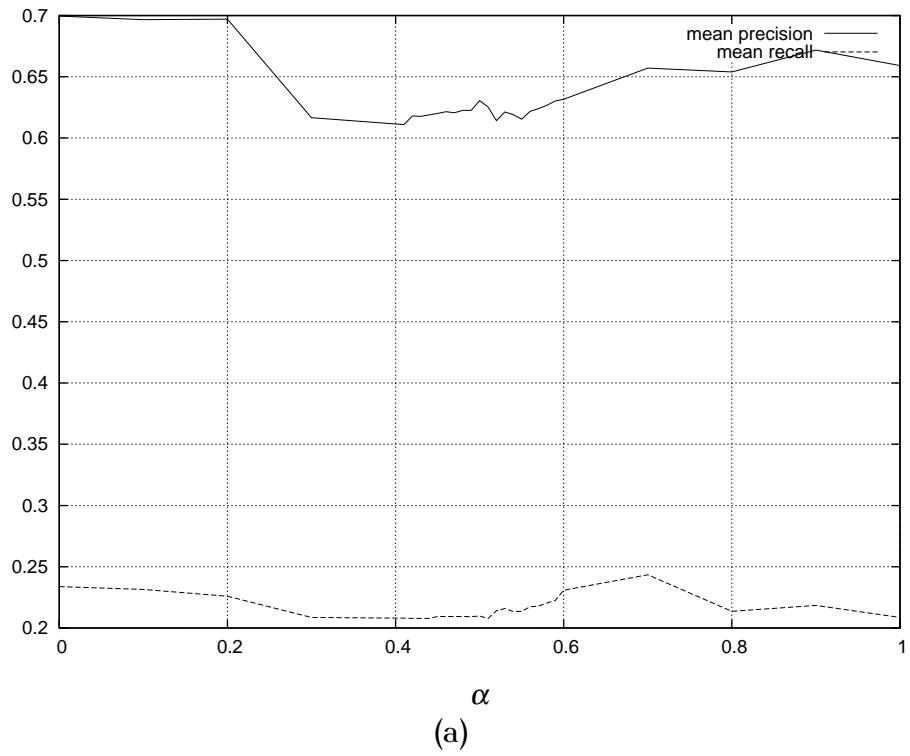
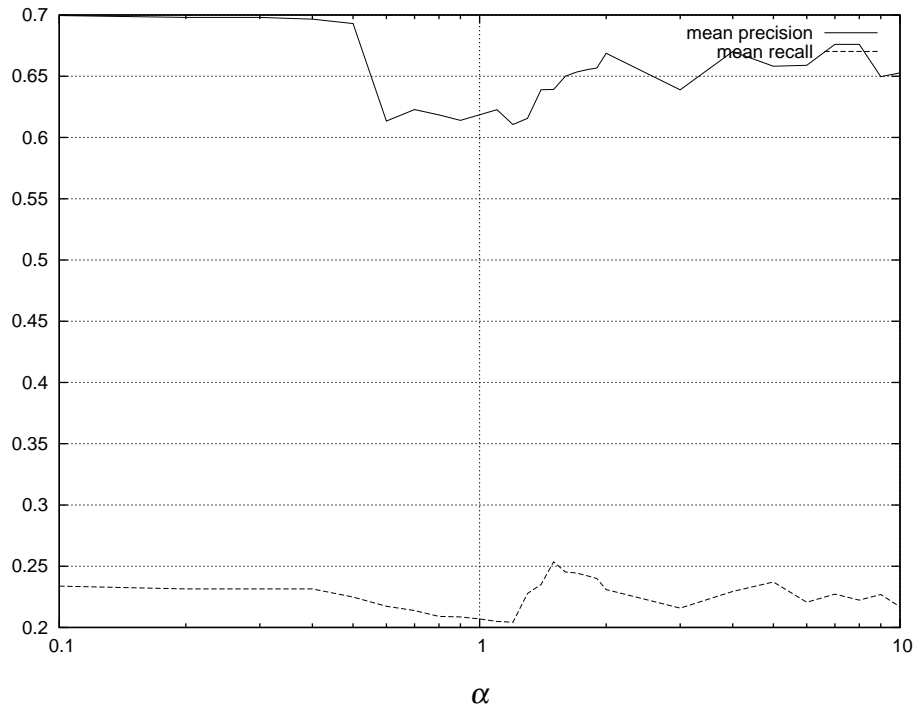
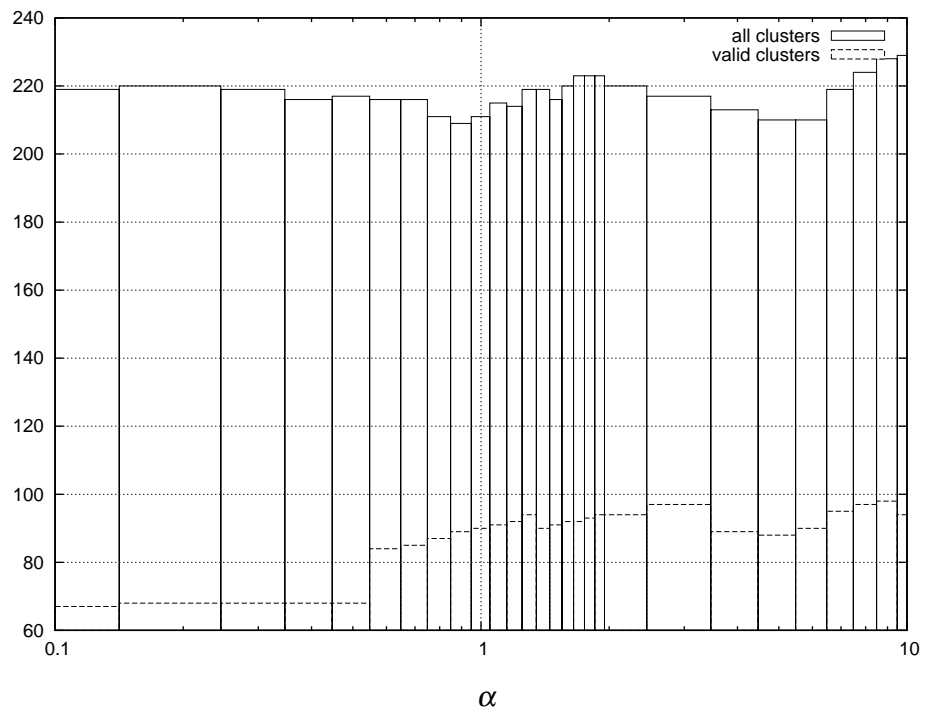


Figure 6.21: Results for combining using average distance value $p = \alpha p_f + (1 - \alpha)p_b$.



(a)



(b)

Figure 6.22: Results for combining using product of probabilities $p = p_f^\alpha \cdot p_b^{1/\alpha}$.

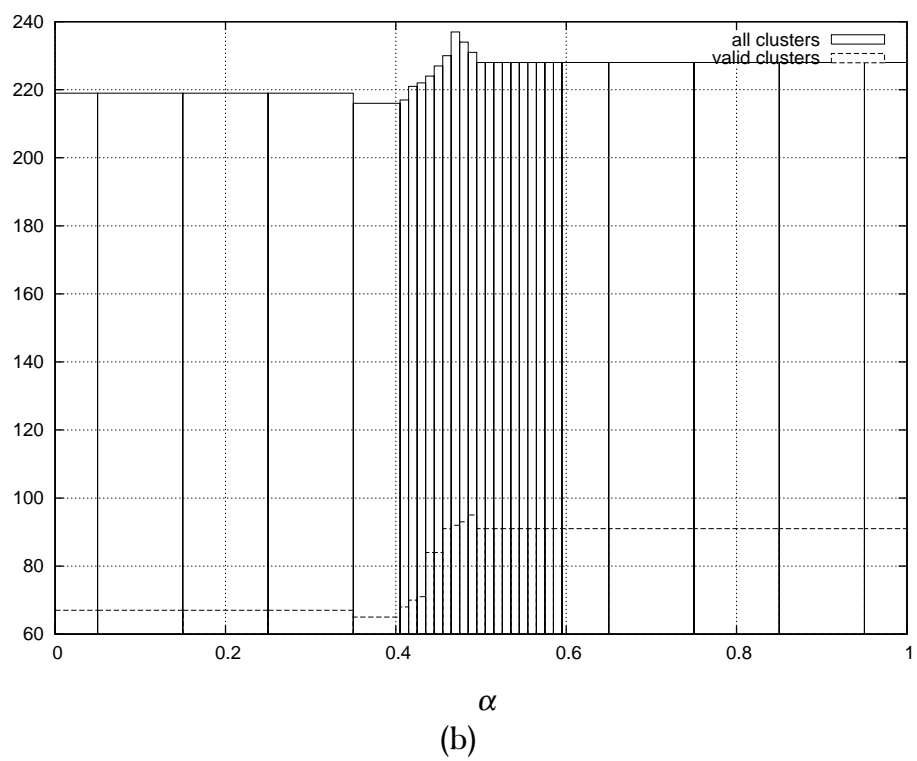
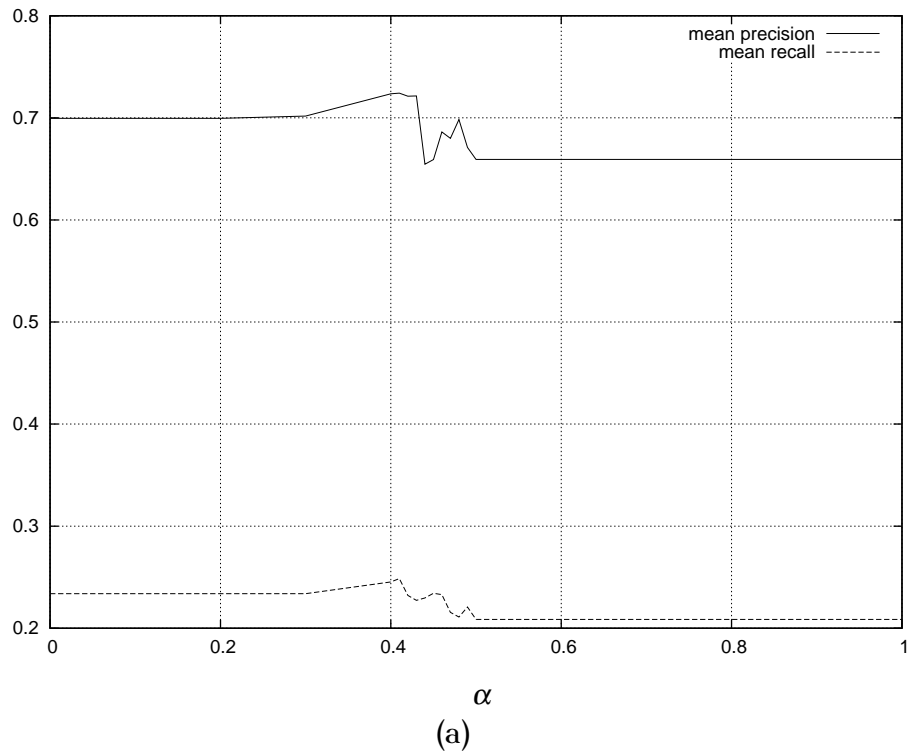


Figure 6.23: Results for combining using maximum $p = \max(\alpha p_f, (1 - \alpha)p_b)$.

6.7 Conclusion

In this chapter, methods for improving clustering of similar persons are presented and discussed. The improvement is achieved by using not only face recognition features but also other sources of information. The sources considered are context-based (events, owners of photographs) and content-based (body patch). A Transferable Belief Model is proposed for fusing face recognition with events and owner information. However, the implementation of such a model is very complex and execution is time consuming. A method for reducing the processing time by analysing a certain number of clusters with the highest values of belief is proposed. The full potential of Transferable Belief Model is not fully exploited, as beliefs associated with events and users are not independent from beliefs associated with face recognition. To use in full the potential of Transferable Belief Model the beliefs of each source should be independent.

Another method, which is proposed in this chapter, combines event and user information with face recognition by analysing a photograph collection at three levels: event level, user level and collection level. At the event level, the photograph collection is divided into subcollections containing only images captured at the same events. Then, nearest neighbour-based clustering is carried out in each subcollection. At the user level, similar clusters created in event subcollections are merged within the user collections. Eventually at the collection level all clusters are analysed and similar clusters merged. The three level clustering provides visible improvements to the clustering across the whole collection at once (see Chapter 5). This is obtained by restricting the initial clustering to images within events only. It reduces the influence of outliers on the clustering and can be seen as clustering in small collections which is more efficient as outlined in Section 6.5.

The observation that during an event, people are likely to wear the same clothes is exploited in Section 6.6, where the analysis of combining the person's body features with face recognition features is presented. Three probabilistic methods for fusion are analysed and experiments carried out, both within events and within user collections. Experiments show an improvement when using the maximum combination method, while the sum method compromises efficiency of both sources.

In the next chapter methods for choosing key images from each of created clusters are analysed. These key images are used for presenting results on a screen with limited space (so that more clusters can be presented).

Chapter 7

Presentation of clusters

7.1 Introduction

In the previous chapter, methods for combining face recognition features with context and content information are presented. This combination is used for enhancing the results of unsupervised clustering of people's identities in a photograph collection. Events, ownership and body patch are the sources of the additional information.

In the previous chapters, the techniques for analysing the appearance of a human face and the methods for grouping facial images into clusters containing images of the same person are described and analysed. This is done in order to aid the user to organise her/his photograph collection for browsing or searching through the organised collection is easier than browsing or searching the unorganised set of photographs. While browsing, the user might not be interested in looking at all the images given in a group. She/he might be more interested in seeing only the most representative photographs. For example, if the user wants to see images from holidays that she/he attended with her/his partner and children, she/he might prefer to see a photograph with the full family accompanied with the photographs of each member of the family separately rather than seeing a hundred photographs captured during the holidays.

In this chapter, methods for presenting the organised collection are discussed with the emphasis put on the choice of the most representative images (called key images). Different methods are described for browsing and different for searching.

7.2 Presentation of clusters

The presentation of created clusters to an end user is as crucial as clustering itself. Let us consider a user browsing her/his collection of photographs. The user would like to see who is present in her/his photographs. In such a situation it would be better to show only a single image of each person in the collection rather than

throwing on the screen all images with people. The vital part of the presentation of photographs is the choice of most representative images from each group of photographs. The most representative images are called here *key images*, *key photographs* or *key-frames*.

In the scenario presented above, the user browses photographs without any specific query to the system (or rather the query specifies only that images containing people are to be shown, which if the collection contains only images of people, is a very general query). In this case the response of the system is not query-based (non-query-based). When the user picks up one or more of the images and tries to find more photographs of the given person, the response of the system is based on the user's query (query-based). The query based presentation of photographs takes place usually when the user searches the collection for some particular images. The non-query-based presentation can be viewed as a special case of the query-based one with a very wide and unspecified query (as it is the case in the scenario presented above). However, those two approaches are presented separately in the following sections.

The goal of key image extraction is the effective and efficient representation and presentation of personal clusters. In the literature many articles can be found on extracting key-frames from video [114, 115, 116, 117, 118, 119, 120]. They usually concentrate on video shot cut detection to find out when scenes start and end. Then, some features describing the scene are extracted and the mean value of features is used for selecting the key-frame — one that is closest to the mean features values. The features consist of any type of information (usually colour) extracted from the mean frame calculated on all frames from the scene.

The still image/photograph management system PhotoTOC [10] selects key images on the basis of the nearest images in terms of colour histogram to the centre of the given group/cluster. The FotoFile system [7] presents key frames extracted from video sequences. However, for still images there is no key image selection. Similarly, other photo management systems like MiAlbum [9], PhotoFinder [11] and systems proposed by Cooper *et al* [14] and Lim *et al* [13] show all images resulting from the query in a thumbnail view rather than choosing key images or do not propose any method for presenting images at all. In this chapter, it is assumed that the photograph collection is already organised by means of events and persons (either unsupervised clustering of people as in Chapter 5, or the propagation of labels/names). In other words there are “event” clusters that group together images taken at the same event (an event is defined as a certain period of time in a certain location, e.g. a trip to the zoo, a party, etc., see Section 6.3), and there are personal (people, identity) clusters, that gather images showing the same person. The discussion here concentrate on the best selection of key images from each group, since the clustering algorithms are described in the previous chapters.

Several decisions need to be made for the successful presentation of personal clusters. It needs to be decided whether clusters are to be presented regardless of events or if it is better to show people in events. Then, it should be decided how many images are needed to be shown for the good representation of an event or a cluster.

The non-query-based presentation takes into account several aspects. One can expect that the chosen key-frames would show most salient images in an event or a cluster for the good representation of the variety of person's appearance in the cluster, or showing as many identities in the event as there occur. For example, having three identities in an event represented with five key-frames, one could show two images of the most populated clusters and a single image of the remaining cluster. Or one could show three separate clusters with the image that is lying closest to the centre of the cluster as the most representative image, accompanied with images lying most apart from the centre in order to show the variety of the faces of the person in the cluster. The statistics of features representing clusters, like the number of identities, the number of human faces, and the number of faces of each identity, etc. can be used for the better representation of personal clusters. As in the examples above, the statistics on the number of images in each cluster would be useful for choosing the distribution of images

The choice of key-frames from a cluster or an event can be viewed as the colour quantisation of a population of images in a cluster or an event. However, the representation would not consist of images obtained as the mean image in a bin, but the closest one to the centre of the bin.

7.3 Key-frames extraction

7.3.1 Non-query-based key-frames

Although they can be viewed as the special case of query based key-frames, the non-query-based key-frames are discussed first, as usually the presentation of clustering results to a user begins with showing the non-query-based key-frames. In this case it is essential to show as diverse examples of the given subject as possible. In the case of personal clusters, they can be grouped in events, showing the variety of clusters (e.g. at least one image from every cluster within an event) or, the clusters can be presented with key-frames chosen from as many events as possible.

Statistics gathered on the collection can be very useful in this case. The statistics might be:

- the number of images in each cluster ($N_c, c = 1, \dots, C$, C is the number of all clusters)

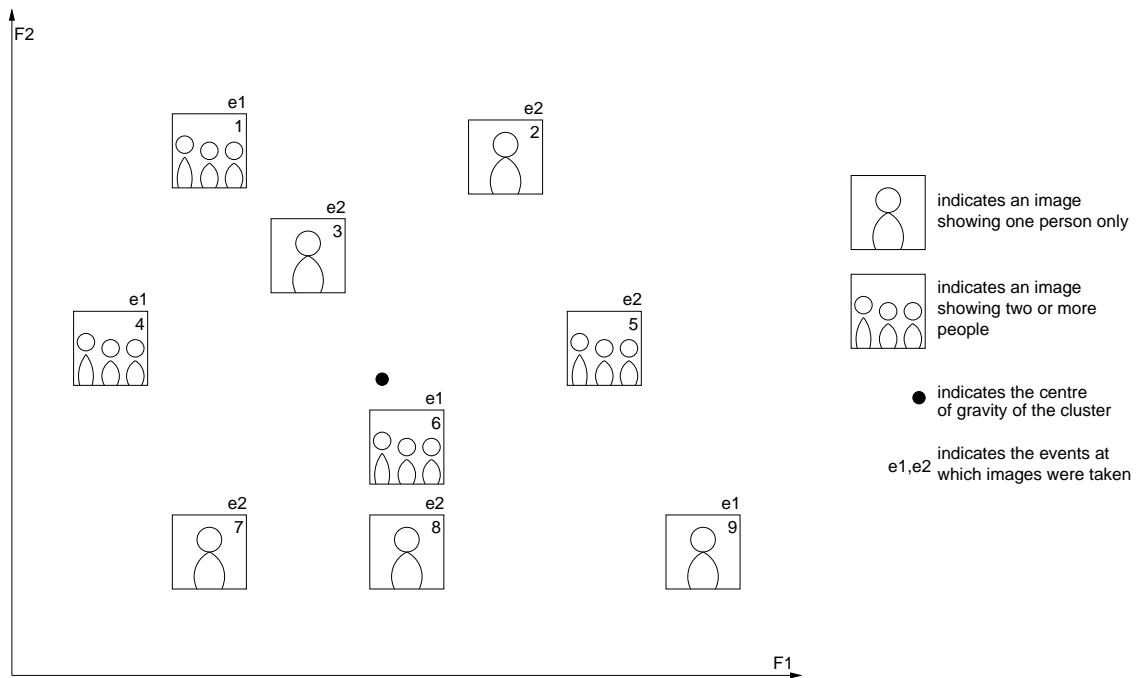


Figure 7.1: An example of a single personal cluster placed in the two-dimensional facial space. The images 1,4,6 and 9 were captured at event e_1 , and the images 2,3,5,7 and 8 were captured at event e_2 . The black dot denotes the centre of gravity of the cluster.

- the number of clusters in an event ($N_e, e = 1, \dots, E$, where E denotes the number of events)
- the number of events that the given cluster is spread across ($N_{ce}, ce = 1, \dots, C$)
- the number of all images containing people ($N_i, i = 1, \dots, I$, where I is the number of images with detected faces)
- the number of all people captured in the collection ($N_p, p = 1, \dots, P$, P is the number of all detected faces).

Additionally, in the case of showing personal clusters, the preference might be given to those photographs that contain only the person of interest. In this way one avoids arbitrariness that occurs when shown images contain several people. Also, the good quality images (i.e. those with faces in focus, with faces occupying large areas, etc.) might be preferred, as they show clearly the face and the user should not have any difficulties in recognising the person.

Proposed methods: Let us denote by K the number of key images to be selected out of each cluster.

- **Method 1:** Centres of clusters

For each cluster choose K images that are nearest to the centre of a cluster (see the flow chart in Figure 7.2(a)). For example, if one would like to show $K = 3$ key images from the cluster presented in Figure 7.1, then this method would produce images 3,6 and 8, because they are nearest the centre of the cluster (Figure 7.2(b)).

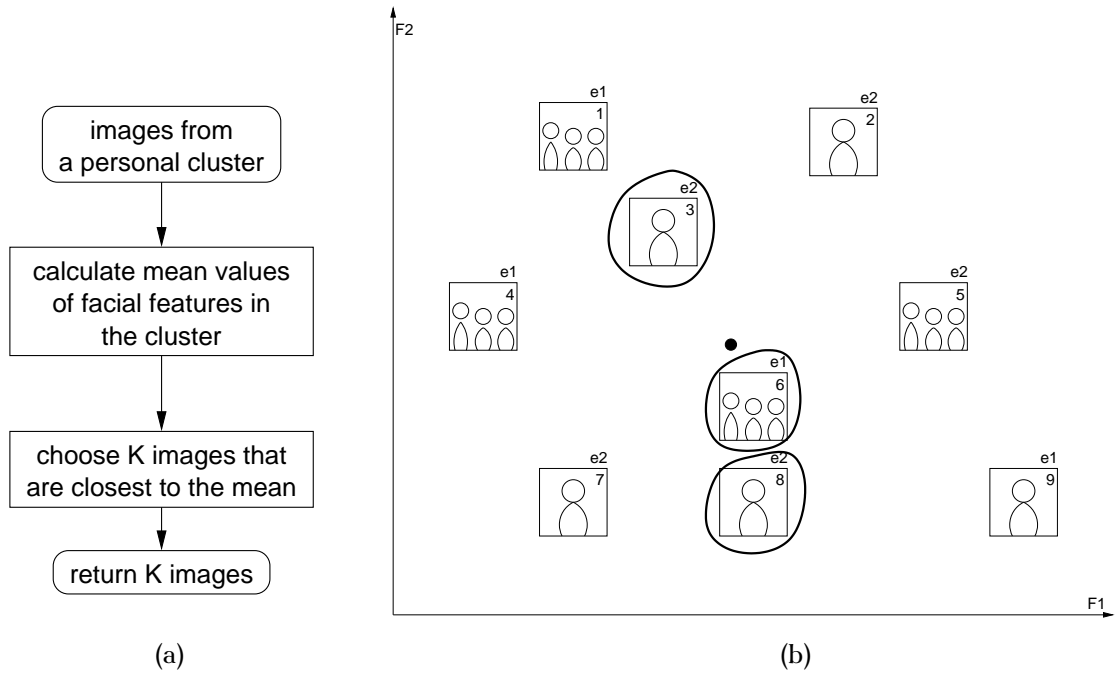


Figure 7.2: (a) the flow chart of the non-query-based Method 1; (b) sample choice of key images from the sample cluster using Method 1.

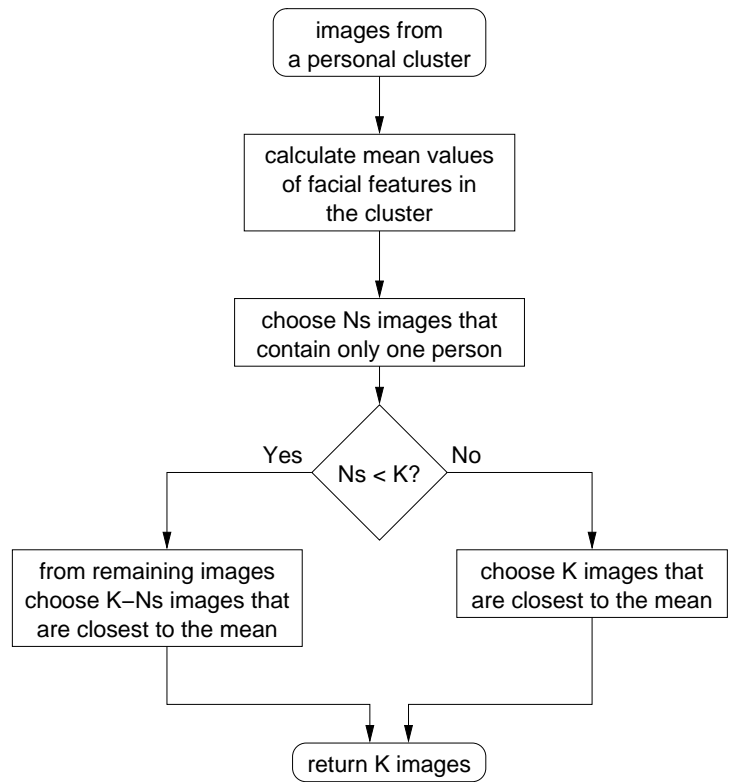
- **Method 2: Single centres**

For each cluster choose the images in which only one person is shown; among these select K that are closest to the centres of clusters. If the number of “single” images is lower than K , then choose the ones nearest to the centre. In the case of the cluster presented in Figure 7.1 the key images are 3,5, and 8 (Figure 7.3(b)). Figure 7.3(a) presents the flow chart of this method.

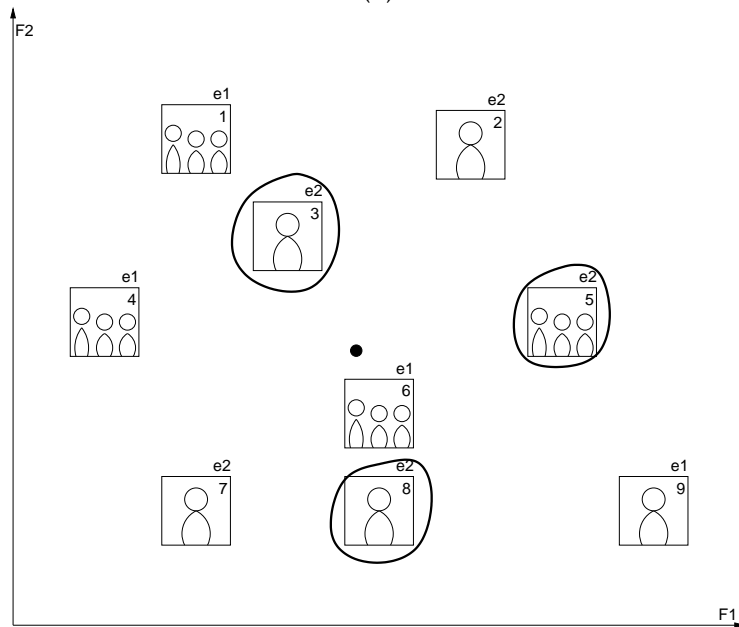
- **Method 3: Events**

If clusters are spread across events, the distribution of the photos in the cluster onto events can be taken into account when considering the selection of key images.

- If N_{ce} is available and $N_{ce} = K$ then one example (using the Method 2) from each event should be chosen.
- If $N_{ce} > K$ then choose a single image from each of K events with the highest N_e , the image as in Method 2,



(a)



(b)

Figure 7.3: (a) the flow chart of the non-query-based Method 2; (b) sample choice of key images from the sample cluster using Method 2.

- If $N_{ce} < K$ then select multiple images from the events of the highest number of images; the number of included images can reflect the relative numbers of images in events.

This method is presented in the form of a flow chart in Figure 7.4(a).

In the example from Figure 7.1, there are two events ($N_{ce} = 2$). This is less than the number of key images $K = 3$, therefore the condition $N_{ce} < K$ is valid. The event e2 contains more images, therefore, the method chooses two key images from event e2 and one from e1. The image 6 is chosen from e1 as the one nearest the centre of the cluster, and images 3 and 8 are chosen from the event e2 (Figure 7.4(b)).

- **Method 4 Quantisation**

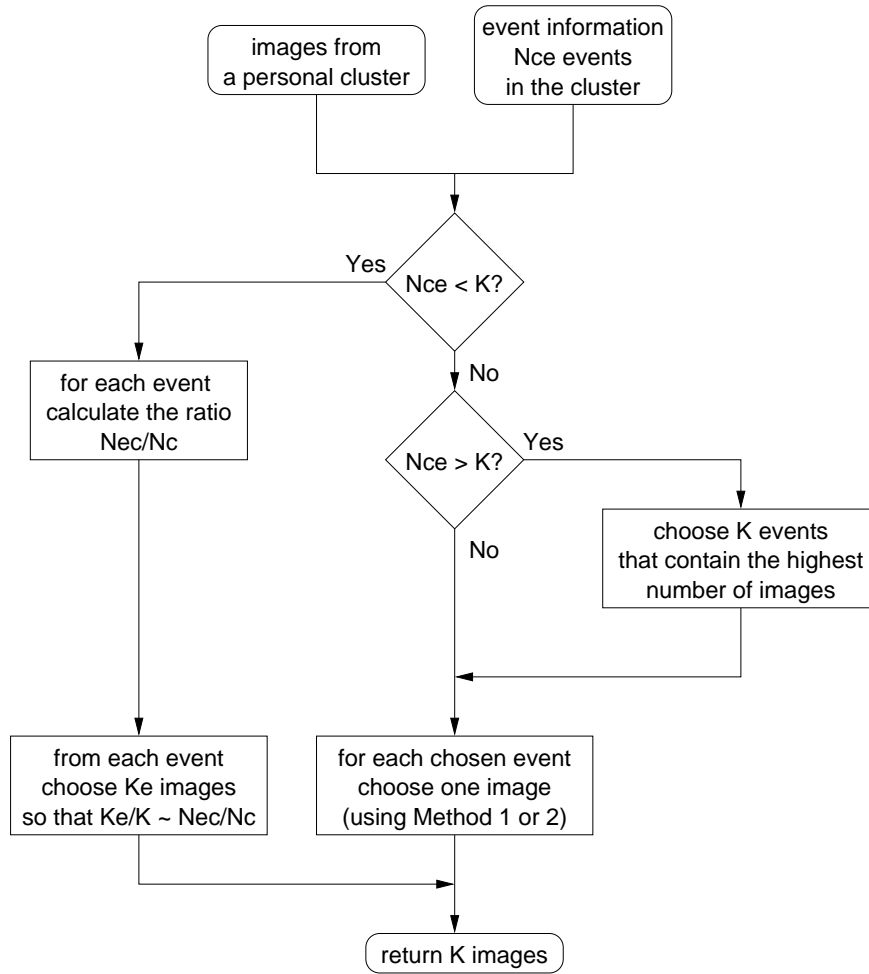
A method similar to the colour quantisation can be used for selecting the key-frames. This method can be seen as clustering within the cluster that results in K groups within the cluster. Then, the key images can be chosen from the images nearest the centres of these groups. Because K is known, the internal clustering is relatively easy, as k-means, n-cuts or the EM algorithm can be used (these algorithms are described in Chapter 5). The flow chart presenting this method is shown in Figure 7.5(a).

For example, in the case presented in Figure 7.5(b) the personal cluster is reorganised into subclusters outlined in Figure 7.5(b) with ellipses. From each of the three subclusters the image that is nearest the centre of the subcluster is chosen. In this example these are images 4,5 and 6.

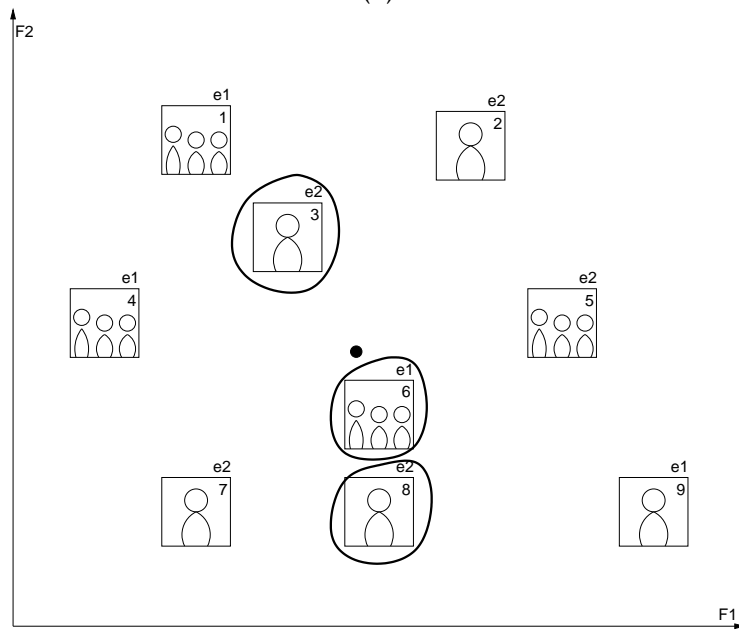
7.3.2 Query-based key-frames

For the query-based key-frame extraction, the circumstances are different, as one would like to show those images that best match the query, rather than the largest possible variety of examples. However, this depends on how specific the query is. If, for example, the user specifies the identity to view, the system still can choose the images from the given cluster that spread as widely as it is possible across various events, or different weather conditions etc. Similarly, if the user specifies the range of time that she/he would like to see photographs from, then the key-frames should contain the large variety of identities photographed in the specified time span.

Queries can become complicated. For example, let us consider a query-by-example, where the example is a photograph containing three people. The results of searching with such a query should ideally contain all three persons, and either show only images containing all three of them (AND operator) or at least one of



(a)



(b)

Figure 7.4: (a) the flow chart of the non-query-based Method 3; (b) sample choice of key images from the sample cluster using Method 3.

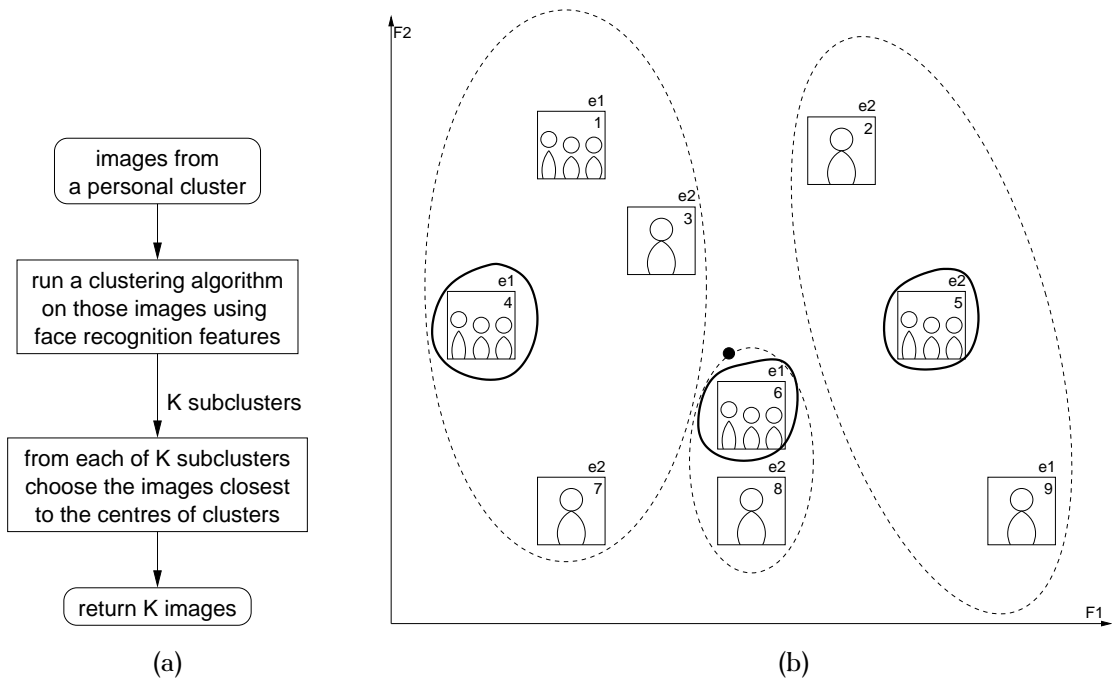


Figure 7.5: (a) the flow chart of the non-query-based Method 4; (b) sample choice of key images from the sample cluster, large ellipses enclose subclusters.

them (OR operator). The user could also draw a rectangular bounding box around the interesting face and use the content of this box for the query.

It should be taken into account that due to non-ideal clustering, several clusters that are more likely to contain the queried person need to be shown, because a single cluster might not contain all of the occurrences of the queried identity.

It is vital that the key-frames extracted using a query-based method reflect the query as well as it is possible. Four methods for the query-based extraction of key images are presented below:

- **Method 1:** When the query consists of a single identity or a cluster, the clusters with centres nearest (in a feature space) to the query image are chosen and the key images of these clusters are chosen as in the unsupervised approach. The flow chart of the method with consecutive steps is presented in Figure 7.6(a).

Let us consider the example presented in Figure 7.6(b). There is the query image positioned in the two dimensional facial space with three nearest personal clusters shown. The user would like to see one key image ($K = 1$) from three nearest clusters ($N = 3$). When Method 1 is applied in this scenario, the images 5 from cluster 2, 8 from cluster 3 and 1 from cluster 1 are chosen (and sorted in the order presented as cluster 2 is the nearest to the query and cluster 1 is the furthest). The obtained key images are outlined in Figure 7.6(b).

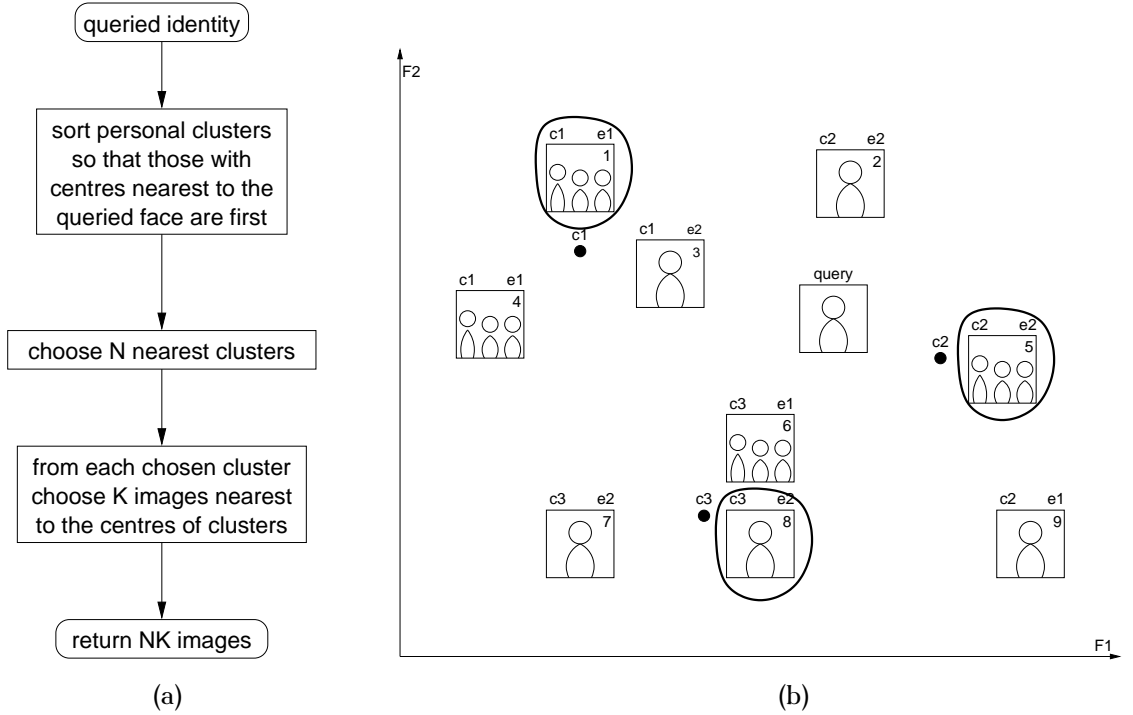


Figure 7.6: (a) the flow chart of the query-based Method 1; (b) sample choice of key images for the query using Method 1.

- Method 2:** Similarly to Method 1, the query consists of a single identity. The key images in the selected clusters are chosen according to their distance to the query image instead of the distance to the centre of the cluster. The flow chart of this method is presented in Figure 7.7(a).

Let us consider again the example query given previously (with the description of Method 1). When Method 2 is applied the images nearest the query image in the facial space are chosen. In this example those are images 2, 6 and 3 as presented in Figure 7.7(b).

- Method 3:** When the query is for retrieving two or more identities, the clusters closest to the query images or identities are retrieved, and based on the cardinality of each cluster N_c , the number of extracted key images is chosen. Let there be $N_{tot} = \sum_{i=1}^n N_{ci}$ images retrieved out of n clusters in response to the n images in the query. The number N_{ki} of key images chosen from the i th cluster needs to conform with the ratios $N_{ki}/K = N_{ci}/N_{tot}$. Figure 7.8(a) presents the flow chart of this method.

Let us consider a sample query image containing two people placed in order to obtain $K = 3$ key images. In the facial space the query image is placed in two locations, accordingly to each face in the image. This is shown in Figure 7.8(b). The clusters with the centres nearest the queried image are chosen, in this example those are the cluster 3 for the person with stripes

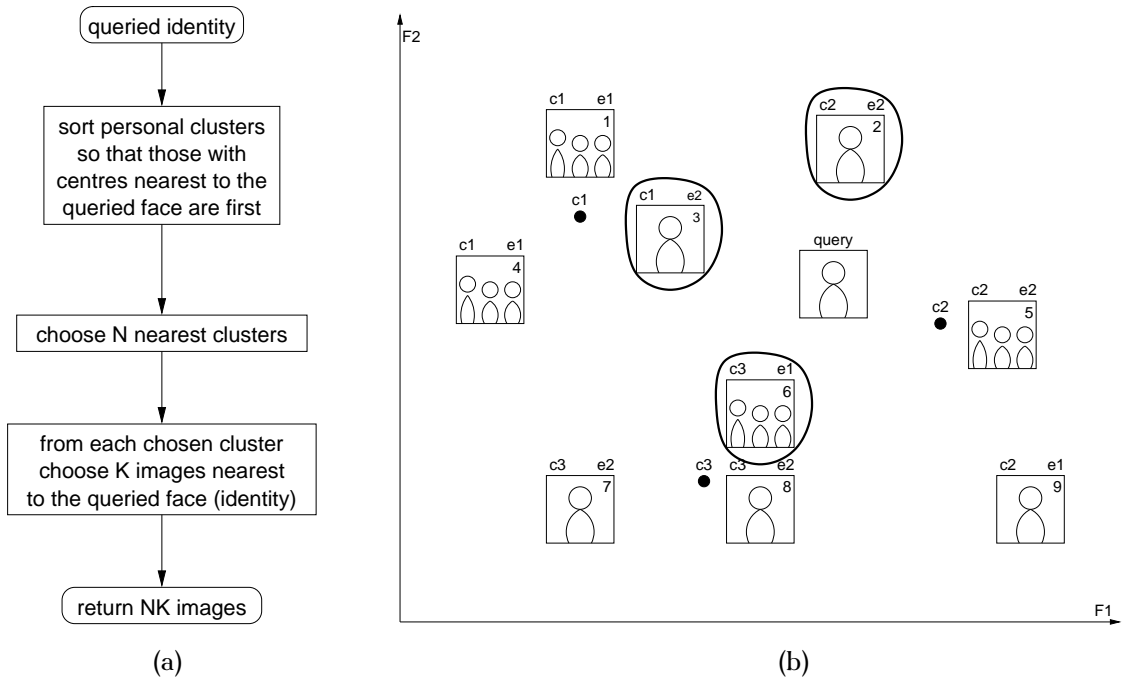


Figure 7.7: (a) the flow chart of the query-based Method 2; (b) sample choice of key images for the query using Method 2.

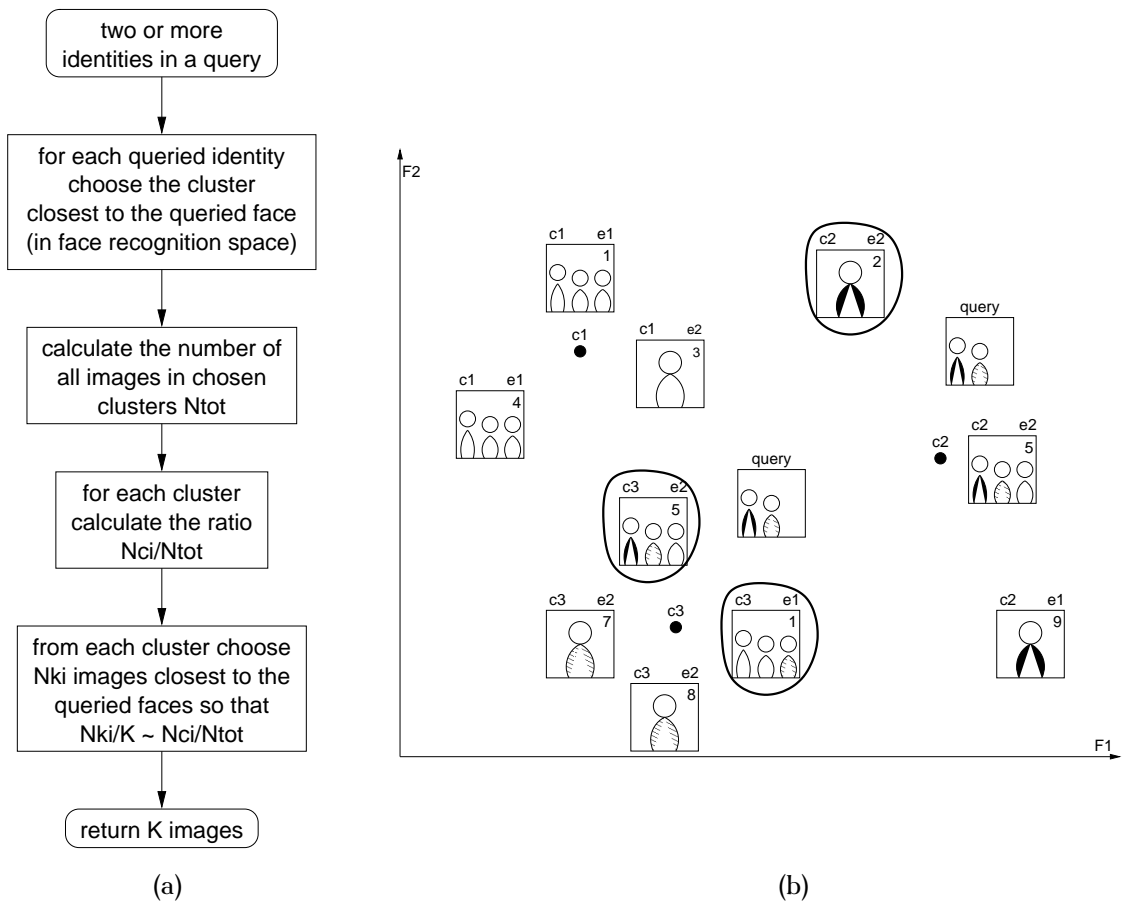


Figure 7.8: (a) the flow chart of the query-based Method 3; (b) sample choice of key images for the query using Method 3.

(right hand side in the query image) and the cluster 2 for the other person in the query image. As the cluster 3 contains more images, two key images from this cluster are chosen and one key image is chosen from the cluster 2. The images are chosen using Method 2, thus the images 1 and 5 from cluster 3 and the image 2 from cluster 2 are obtained (see Figure 7.8(b)). It is possible that in this approach some images can be chosen twice or more times. For example, the image 5 can be chosen twice as both persons from the query image are present in this image.

- **Method 4:** More specific queries like e.g. a query for the particular person in the certain range of time would require getting the clusters of this person and choosing the images that appear in the events that took place in the queried period of time. The flow chart of this method is presented in Figure 7.9(a).

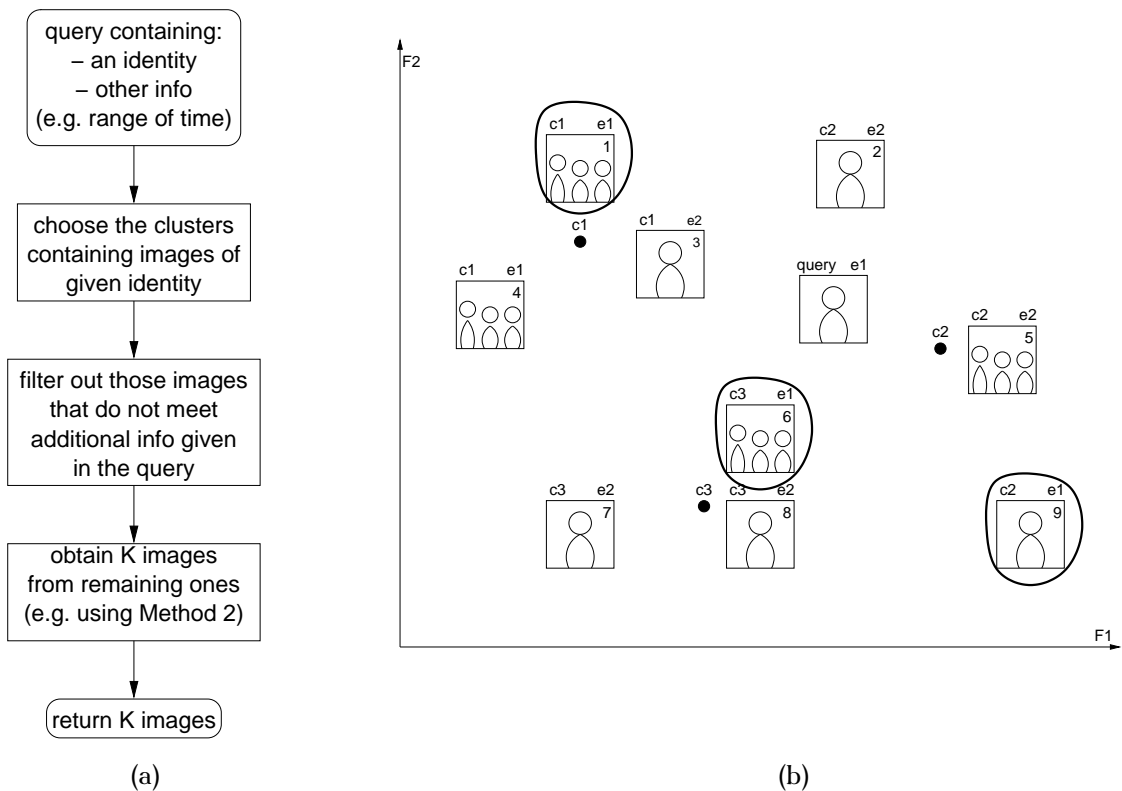


Figure 7.9: (a) the flow chart of the query-based Method 4; (b) sample choice of key images for the query using Method 4.

The example of such a query is presented in Figure 7.9(b). The query image was captured at the event e1 and the user wants to obtain images of this person captured at the same event (let us assume that $K = 3$ images can be shown). Firstly the three nearest clusters are found. Then, the images in those clusters are filtered, so that only images captured at event e1 are left (those are images 1,4,6 and 9). Among those the images are chosen using Method 2, that returns to the user the images 1,6 and 9.

In this section methods for non-query-based and query-based choice of key images are presented. Although some of those methods work in different conditions than others (e.g. single identity query requires different method than multiple identities query), their efficiency must be measured in some way in order to compare them to each other or to any other method. Thus in the next section some evaluation methodologies of the choice of key-frames are discussed.

7.4 Key-frame evaluation

The evaluation of the key-frame selection is not a trivial task as the choice of key-frames can be very subjective since different people may have different needs for representing the same clusters. The evaluation of the key-frame choice can be approached in two ways:

- objective — by a certain value calculated on the selected key-frames, measuring the “goodness” of the key-frame selection
- subjective — based on a user study and a user satisfaction

7.4.1 Objective measurements

Objective measurements are easier to conduct than the subjective ones as they do not require any human interaction. However, such measurements might not necessarily accurately model the human’s perception of key-frame quality, and such modelling might or might not be required.

The measurements that are not intended to model human perception can be based on comparing the statistics mentioned in Section 7.3.1 with the key-frame selection. Another possibility is analysing the statistics of the chosen key-frames such as their spread within the collection, or the variety of features they represent.

Yet another criterion for assessing the key images selection is the ratio of how well the key image represents the given group of data to how well it distinguishes from all the rest of the data. Let us denote by d_{in} the similarity measure of the selected key photograph to the group it represents and by d_{out} the similarity of the key image to all other photographs. One can expect that the key images with the highest d_{in}/d_{out} ratio are more representative than the ones with lower value of this ratio.

7.4.2 User study

There are at least two ways of evaluating the key-frame selection using user inputs:

- the user chooses most relevant in her/his opinion key-frames, then the key frames selected by the system are compared with the key images chosen by the user;
- the user is presented with the system, that uses automatically selected key frames and fills up a questionnaire expressing her/his satisfaction of using the system.

In the situation when the user chooses her/his key images for the representation of personal clusters, one can build a model of user key-frames based on given user's selections. These key images can be compared to the key photographs selected by the system and the measure of how well they correspond to each other can be calculated. However, this measure would be highly biased towards the user's own subjective notion of which photographs should be chosen. Thus such measure would favour key-frames following user's model. This measure, however, might suggest directions for improvement to the method of selecting key photographs.

A straight forward measure in such a scenario is the average distance of automatically selected key-frames to the key images selected by the user $1/K \sum_{k=1}^K d_k(F_a, F_{ua})$, where F_a denotes automatically selected key image, F_{ua} is the closest key image selected by the user, and K is the number of selected key-frames. This measure has a drawback in being dependent on the average distance in clusters, and would be different for clusters of different distances between points. Therefore, the distance between the key images should be normalised, preferably by the number reflecting the average distance within each cluster $\bar{d}_c = 1/N_c^2 \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} d(F_i, F_j)$

The second measure, the user satisfaction measure, also is biased towards the subjective user's notion of key images. However, in this case it seems to be easier to analyse inputs of several users and draw some conclusions from averaging the users' marks. Unless the users are asked explicitly in the questionnaire for additional comments, it is hard to find out the directions for improvement. Users themselves might not be able to suggest any improvement. However, they might be able to choose one of the methods presented to them.

7.5 Conclusion

In this chapter, methods for extracting key images are presented. Two groups of techniques are discussed: non-query-based and query-based. The query-based methods can be used for the non-query-based extraction, since the non-query-based extraction can be seen as the query-based one with a very general (non-specific) query. Some methods for evaluation of extracted key images are discussed, too. The objective (user-independent) techniques are presented and the

discussion on the possibilities of the user-dependent evaluation is included. The evaluation criteria can be also used as the extraction criteria. Then, evaluation can only assess how well a given criterion is fulfilled by the given extraction method.

The next chapter contains the summary of the thesis and two directions for future work. These include another approach to the Transferable Belief Model employed for combining various sources of information and a short discussion on possibilities to exploit the user interaction.

Chapter 8

Summary

8.1 A brief review

This thesis presents a full system for unsupervised grouping of digital photographs that show people who are similar in appearance. The system is based primarily on the human face analysis. However, it also exploits additional information, both content (body patch) and context-based (events, user).

A large amount of effort was put into methods for facial components localisation, especially eyes, as component-based and holistic face recognition methods require accurate locations of facial components. Accurately located eyes are essential, not only for component-based techniques, but also for holistic methods. The latter analyse facial images that are normalised by eye positions. A novel method for facial components extraction and localisation is proposed by the author. This method is based on colour segmentation and the observation that facial components such as eyes, lips and eyebrows differ in colour from the skin area of a human face. Another technique for eye localisation, based on a PCA template, is analysed in depth. This method is simpler and much quicker than the one based on colour segmentation. It is based on the assumption that a bounding box is accurately based around the facial region in an image. Both techniques were tested on low resolution images with limited number of details and different lighting conditions. Results, therefore, are not as good as results presented in the works of others.

A variety of techniques for face recognition is presented in this work. Investigation into which of the methods best meet the needs of unsupervised clustering is presented. The suitability of various recognition methods for the use in unsupervised clustering is analysed. It was decided to use the MPEG-7 face recognition descriptor for further experiments. This is a convenient method that allows the user to extract the facial features used for recognition. There is no need for any training set of faces because the basis vectors are defined by the standard. Faces can be easily compared by means of the distance measure between feature vectors.

Although MPEG-7 standard does not define any pre-processing of the facial image except normalisation (which is based on the locations of eyes) it is investigated in this thesis how pre-processing of the normalised facial image with histogram equalisation (for removing the influence of changes in lighting) and feathering (to avoid the background in the image) might affect the recognition.

Next, the analysis of clustering techniques for grouping of similar persons in an image collection is presented. Experiments with three grouping techniques were conducted by the author:

- modified k-means
- modified single-link
- nearest neighbour-based

All the techniques were modified to work in a totally unsupervised manner, in such a way that they estimate the number and parameters of clusters, and classify faces into the clusters. The k-means and single-link algorithms estimate the number of clusters using the outlier detection technique based on thresholding. The normalised distance was employed for the k-means algorithm in order to take into account the spread of data points within clusters. The nearest neighbour-based technique gives the best results. However, it is hard to find a model for clusters produced by this technique. The estimation of the number of clusters is based on thresholding. However, this is the number of neighbouring clusters that is thresholded, not the distance between data points. A post-processing technique for further enhancing clusters is described. This method detects faces, which are extracted from the same photograph (two or more faces are “present” in the same photograph) and classified to the same cluster (recognised to be different occurrences of the same person in a single photograph). This method also reallocates wrongly classified data points.

In addition to face recognition, sources of information such as body patch, events, and user information are analysed. The mechanisms behind the fusion of information from these sources are investigated. One of the methods proposed for fusing information about events and user with face recognition is the Transferable Belief Model based on Belief Theory. However, the model proposed in this thesis is very computationally complex, therefore some simplifications for implementation are proposed.

Much better results are obtained by the author when event and user information is used for restricting the search space on which clustering is carried out. This greatly increases the precision value, but there is no improvement in regard to recall. The low values of recall can be explained by the strong restriction put on clustering within events, as people are quite often captured in photographs from several events. However, merging clusters at the user and collection levels does

not increase the recall values significantly. This suggests that either the appearance of human face varies between events, or that the merging method should be improved.

Simple probabilistic methods are analysed for the fusion of face recognition features with information about the colour and texture of the body (clothes). Information of events is also used, since the body patch is expected to work well only within events. However, when face recognition is combined with body patch within user collections, across events, the clusters created are not much different than the ones created within events. This suggests that the combination of body patch with face recognition is capable of suggesting which images were captured at the same events.

A discussion on the possibilities for extracting key images for clusters concludes this thesis. Some techniques for extracting non-query-based and query-based key images are discussed. It is observed that the non-query-based extraction can be viewed as a query-based one with a very general (unspecific) query. Some methods for evaluation of the choice of key images are included. Two groups of evaluation methods are presented: objective ones and the ones based on user preferences. This covers the presentation of clustering results to the user on a limited area such as the screen of a PDA device.

8.2 Future work

There are many directions in which current work can be extended. The similarity of human faces is a field in which largest enhancements can be made, as face recognition, in spite of the long time that researchers have spent on this topic, it is still very error-prone when dealing with real life photographs. Other areas that can greatly enhance the process of finding persons of similar appearance are clustering itself and the fusion of information sources.

There is a source of information which was not exploited in this thesis — the user. The user input, although is not necessary for unsupervised clustering, can be used for correcting the output of the clustering process. It can also be fed into the relevance feedback mechanism or used in fusion algorithms as another, very reliable source of information.

In the next subsections, two directions of future work are presented: another implementation of Transferable Belief Model and the user interaction.

8.2.1 Transferable Belief Model on similarity measure

In this approach to TBM for clustering, basic belief assignment (BBA) functions measure the degree of belief that two points are similar. This is unlike the TBM presented in Section 6.4, in which BBA functions measure the degree of belief that

the given point belongs to the certain cluster. The approach proposed for the future work has the advantage that it does not require any initial membership to the clusters. This preserves independency between information sources.

Two hypotheses based on the distance measure between two points can be considered: S (similar) and N (not similar). The hypothesis S tells us that two considered points are similar, i.e. they belong to the same class (in the case of matching similar appearances of people this is the hypothesis, that both data point are instances of the same person). The hypothesis N is considered when two points are not similar at all, i.e. they belong to different classes (represent two different people).

This set of hypotheses produces the powerset $\{\emptyset, S, N, SN\}$, where $SN = \Omega$ represents the doubt or uncertainty whether the given distance can tell us that the given two points are of the same person or different people.

There are three BBA functions that can be defined on the distance measure between two data points:

- $m(S, d)$ - BBA that points are similar
- $m(N, d)$ - BBA that points are not similar at all
- $m(SN, d) = m_{\Omega}(d)$ - represents doubt

These functions are defined as presented in Figure 8.1. If the distance is below T_{min} , then we are sure that the given points are similar, therefore we can assign all weight of belief to the belief $m(S)$. The value of T_{min} is obtained by finding the minimal distance between the two points of different classes among all points in the collection. When the distance exceeds T_{max} , we are certain that the given points are of different classes, therefore we can assign $m(N) = 1$. This threshold can be found by searching for the maximum distance between the two points of the same class among all classes. The values of T_{mean1} and T_{mean2} are mean distances measured between the points of the same class and the points of different classes, respectively. If the distance is between T_{mean1} and T_{mean2} , then we cannot say whether the points are similar or not, thus the full weight of belief is assigned to our doubt $m(SN)$. Table 8.1 shows the values of thresholds obtained for the large testing dataset described in Appendix C.2.

Table 8.1: Thresholds for MPEG-7 FR features (1127 faces, including Unknowns).

T_{max}	1389
T_{min}	216
T_{mean1}	767
T_{mean2}	814

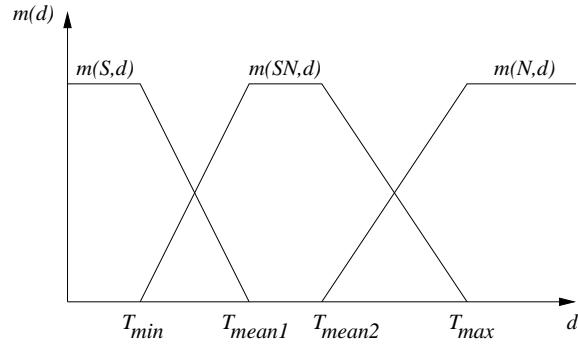


Figure 8.1: BBAs as functions of a distance d .

Table 8.2: Components of sums for Dempster's rule of combination.

m_e	m_f		
	S	SN	N
S	S	S	\emptyset
SN	S	SN	N
N	\emptyset	N	N

Such assignment of BBAs can be carried out for each source of information, and these BBAs can be combined using Dempster's rule of combination. Let us denote by $m_f(S), m_f(SN), m_f(N)$ the BBAs created for the face recognition features and by $m_e(S), m_e(SN), m_e(N)$ the BBAs created for events. Then the combined BBAs are:

$$\begin{aligned}
 m(S) &= \sum_{A \cap B \in S} m_f(A)m_e(B) \\
 &= m_f(S)m_e(S) + m_f(S)m_e(SN) + m_f(SN)m_e(S) \quad (8.1)
 \end{aligned}$$

$$m(SN) = \sum_{A \cap B \in SN} m_f(A)m_e(B) = m_f(SN)m_e(SN) \quad (8.2)$$

$$\begin{aligned}
 m(N) &= \sum_{A \cap B \in N} m_f(A)m_e(B) \\
 &= m_f(N)m_e(N) + m_f(N)m_e(SN) + m_f(SN)m_e(N) \quad (8.3)
 \end{aligned}$$

Similarly, information about ownership and body patch can be combined.

The difficulty that one is faced with, in this approach is the definition of a distance (or similarity measure) between images or faces from the same and different events and of the same or different user.

8.2.2 User interaction

The idea of unsupervised clustering of similar faces, which ideally would be of the same person, is intended to reduce the amount of user interaction. However, still some amount of user interaction is needed. For example, it is more convenient for the user to annotate people in images with their names rather than using automated

abstract cluster labels. However, having clustered faces, the user can annotate the cluster rather than individual pictures with an appropriate name.

Adding new images

Adding new images can be analysed in two scenarios:

- new photographs are of a new event
- new photographs of an event already represented in the collection

Photographs of new events

Adding new images captured at an event or events that are not represented in the collection yet, is quite straight forward. These new images can be easily analysed using the three level clustering (event level, user collection level and full collection level). Firstly the images are pre-organised in new events, then the identified faces are clustered and matched within these new events (or this new event, if one only). Once the clustering within the new events is finished, the created clusters are matched to the existing clusters within the user collection. After that, broader matching and merging is performed on all clusters in the full collection.

Photographs of existing events

When considering three level clustering, adding new images captured at the same event or events as photographs already existing in the collection provides just minor difficulties in comparison to the situation presented in the previous section. The difficulty is in finding the events, which the new images were captured at, then the clustering can be performed only within these events. Another difficulty is then on updating the merging of clusters in user collection level, as existing clusters are changed, which affects previous merging.

Corrections and labelling

Once the automated clustering is done and clusters are available, they can be presented to the user, who should be allowed to make corrections to the created clusters and also to name (label) them. Although, giving names is not that important from the system point of view, as the names are just another label given to the clusters, the system has to handle the user's corrections in an efficient way and use them for improving the clustering of new photographs or those that were not corrected.

The user interaction can be viewed as another source of information, just like facial analysis or body patch analysis. However, the user is the very reliable

source with little uncertainty assigned to this source. Thus this information should be given a priority over other sources.

Bibliography

- [1] A.L. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.
- [2] M. B. Stegmann, B. K. Ersbøll, and R. Larsen. FAME – a flexible appearance modelling environment. *IEEE Trans. on Medical Imaging*, 22(10):1319–1331, 2003.
- [3] C-H. Lin and J-L. Wu. Automatic facial feature extraction by genetic algorithms. *IEEE Transactions on Image Processing*, 8(6), June 1999.
- [4] A.V. Nefian and M.H. Hayes. Hidden markov models for face recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, pages 2721–2724, 1998.
- [5] N. O'Hare, H. Lee, S. Cooray, C. Gurrin, G. Jones, J. Malobabic, N. O'Connor, A. F. Smeaton, and B. Uscilowski. Mediassist: Using content-based analysis and context to manage personal photo collections. In *CIVR2006 - 5th International Conference on Image and Video Retrieval. Springer Lecture Notes in Computer Science*, volume 4071, pages 529–532, Berlin/Heidelberg, Germany, 2006.
- [6] N. O'Hare, H. Lee, S. Cooray, C. Gurrin, G. Jones, J. Malobabic, N. O'Connor, A. F. Smeaton, and B. Uscilowski. Automatic text searching for personal photos. In *SAMT 2006 - Proceedings of The First International Conference on Semantics And Digital Media Technology*, Athens, Greece, 2006.
- [7] A. Kuchinsky, C. Pering, M.L. Creech, D. Freeze, B. Serra, and J. Guizdka. Fotofile: A consumer multimedia organization and retrieval system. In *SIGCHI conference on Human factors in computing systems*, pages 496–503, 1999.
- [8] Platt J.C. Autoalbum: clustering digital photographs using probabilistic model merging. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 96–100, 2000.

- [9] W. Liu, Y. Sun, and H. Zhang. Mialbum: a system for home photo management using the semi-automatic image annotation approach. In *ACM Multimedia*, 2000.
- [10] J. C. Platt, M. Czerwinski, and B. Field. Phototoc: automatic clustering for browsing personal photographs. Technical Report MSR-TR-2002-17, Microsoft Research, 2002.
- [11] Hyunmo Kang and Ben Shneiderman. Visualization methods for personal photo collections: Browsing and searching in the photofinder. In *IEEE Int. Conf. on Multimedia and Expo, ICME*, volume 3, pages 1539–1542, 2000.
- [12] Hu Chen, Li Zhang, and Zhang. Face annotation for family photo album management. *Int. Journal of Image and Graphics*, 3(1), 2003.
- [13] Philippe Mulhem Joo-Hwee Lim, Qi Tian. Home photo content modeling for personalized event-based retrieval. *IEEE MultiMedia*, 10(4):28–37, October/December 2003.
- [14] M. Cooper, J. Foote, and A. Girgensohn. Automatically organizing digital photographs using time and content. In *Int. Conf. on Image Processing, ICIP*, 2003.
- [15] Florent Monay and Daniel Gatica-Perez. On image auto-annotation with latent space models. In *ACM Int. Conf. on Multimedia (ACM MM)*, 2003.
- [16] O’Hare N., Gurrin C., Lee H., Murphy N., Smeaton A.F., and Jones G. Digital photos: Where and when? In *13th ACM International Conference on Multimedia*, 2005.
- [17] O’Hare N, Gurrin C, Jones G, and Smeaton A.F. Combination of content analysis and context features for digital photograph retrieval. In *2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, 2005. London, UK.
- [18] RIYA system, <http://www.riya.com>.
- [19] Neil O’Hare, Hyowon Lee, Saman Cooray, Cathal Gurrin, Gareth Jones, Jovanka Malobabic, Noel O’Connor, Alan F. Smeaton, and Bartlomiej Uszilowski. Mediassist: Using content-based analysis and context to manage personal photo collections. In Hari Sundaram, Milind Naphade, John R. Smith, and Yong Rui, editors, *CIVR2006 - 5th International Conference on Image and Video Retrieval. Springer Lecture Notes in Computer Science*, volume 4071 / 2006, pages 529–532, 2006.

- [20] A. W. Fitzgibbon and A. Zisserman. On affine invariant clustering and automatic cast listing in movies. In *7th European Conference on Computer Vision*, 2002.
- [21] Saman Cooray, Noel O'Connor, Cathal Gurrin, Gareth Jones, Neil O'Hare, and Alan F. Smeaton. Identifying person re-occurrences for personal photo management applications. In *VIE 2006 - IEE International Conference on Visual Information Engineering, Innovation and Creativity in Visual Media Processing and Graphics*, 2006.
- [22] Jain A.K., Murty M.N., and Flynn P.J. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, September 1999.
- [23] R. Gross, J. Shi, and J. Cohn. Quo vadis face recognition? - the current state of the art in face recognition. Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2001.
- [24] *Information technology — Multimedia content description interface — Part 8: Extraction and use of MPEG-7 descriptions*, 2002. ISO/IEC 15938-8:2002.
- [25] Michalski R., Steep R., and Diday E. Automated construction of classifications: conceptual clustering versus numerical taxonomy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 396–409, September 1983.
- [26] B. Uscilowski and T. Curran. Face components extraction via image segmentation. In *Workshop on Signal Processing SP'2002*, Poland, 2002.
- [27] K. Sobottka and I. Pitas. Extraction of facial regions and features using color and shape information. In *Int. Conf. on Pattern Recognition (ICIP)*, Vienna, Austria, August 1996.
- [28] S. Cooray and T. Curran. Region-based facial feature extraction for face detection in color images. In *1st Indian Int. Conference on Artificial Intelligence*, December 2003.
- [29] O. Arandjelovic and A. Zisserman. Automatic face recognition for film character retrieval in feature-length films. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [30] M. Davis, M. Smith, F. Stentiford, A. Bamidele, J. Canny, N. Good, S. King, and R. Janakiraman. Using context and similarity for face and location identification. In *IS&T/SPIE 18th Annual Symposium on Electronic Imaging Science and Technology Internet Imageing VII*, 2006.

- [31] R. Chelappa, C. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5), May 1995.
- [32] M. Kampmann and L. Zhang. Estimation of eye, eyebrow and nose features in videophone sequences. In *International Workshop on Very Low Bitrate Video Coding (VLBV 98)*, Urbana, USA, October 1998.
- [33] R. Linggard, D. J. Myers, and C. Nightingale. *Neural Networks for Vision, Speech and Natural Language*. Chapman & Hall, 1992.
- [34] R.S. Feris, J. Gemmell, K. Toyama, and V. Krüger. Hierarchical wavelet networks for facial feature localization. In *Proceedings of Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 125–30, 2002.
- [35] Cootes T. *An Introduction to Active Shape Models*, chapter 7, pages 223–248. Oxford University Press, 2000.
- [36] Cootes T.F., Edwards G., and Taylor C.J. Comparing active shape models with active appearance models. In *British Machine Vision Conference, BMVC'99*, 99.
- [37] E. Saber and A.M. Tekalp. Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions. *Pattern Recognition Letters*, 19(8):669–680, 1998.
- [38] A. Jacquin and A. Eleftheriadis. Automatic location tracking of faces and facial features in video sequences. In *Proc. Int. Workshop on Automatic Face and Gesture Recognition*, Zurich, Switzerland, June 1995.
- [39] M. Nixon. Eye spacing measurement for facial recognition. In *SPIE Proc.*, volume 575, pages 279–285, 1985.
- [40] J.M. Vincent, D.J. Myers, and R.A. Hutchinson. *Neural Networks for Vision, Speech and Natural Language*, chapter Image Feature Location in Multi-Resolution Images Using a Hierarchy of Multilayer Perceptrons, pages 13–29. 1992.
- [41] C.C. Hand, M.R. Evans, and S.W. Ellascott. *Neural Networks for Vision, Speech and Natural Language*, chapter A Neural Network Feature Detector Using a Multi-Resolution Pyramid. 1992.
- [42] R.M. Debenham and S.C.J. Garth. *Neural Networks for Vision, Speech and Natural Language*, chapter The Detection of Eyes in Facial Images Using Radial Basis Functions. 1992.

- [43] M.J.T. Reinders, R.W.C. Koch, and J.J. Gebrands. Locating facial features in image sequences using neural networks. In *Second Int. Conference on Automatic Face and Gesture Recognition*, pages 230–235, Killington, USA, 1997.
- [44] J. Huang and H. Wechsler. Visual routines for eye location using learning and evolution. *IEEE Transactions on Evolutionary Computation*, 4(1), April 2000.
- [45] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, June 1991.
- [46] S. Spors, R. Rabenstein, and N. Strobel. Joint audio-video object tracking. In *IEEE International Conference on Image Processing*, Greece, October 2001.
- [47] K. Talmi and J. Liu. Eye and gaze tracking for visually controlled interactive stereoscopic displays. 14(10):99–810, 1999.
- [48] K. Takaya and K-Y. Choi. Detection of facial components in a video sequence by independent component analysis. In *Int. Conf. on Independent Component Analysis and Blind Signal Separation*, pages 266–271, San Diego, USA, December 2001.
- [49] P.J.G. Lisboa. *Neural Networks for Vision, Speech and Natural Language*, chapter Image Classification Using Gabor Representations with a Neural Net. 1992.
- [50] Viola P. and Jones M. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.
- [51] E.A. Johnson and C-H. Wu. Real-time fuzzy logic-based neural facial feature extraction technique. In *IEEE International Conference on Fuzzy Systems*, volume 1, pages 268–273, 1994.
- [52] H. Wu, J. Inada, T. Shioyama, Q. Chen, and T. Simada. Automatic facial feature points detection with susan operator. In *Proceedings of 12th Scandinavian Conference on Image Analysis*, pages 257–63, 2001.
- [53] Cooray S, O'Connor N, Marlow S, Murphy N, and Curran T. Semi-automatic video object segmentation using recursive shortest spanning tree and binary partition tree. In *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'01)*, May 2001.

- [54] O'Connor N. *Video Object Segmentation for Future Multimedia Applications*. PhD thesis, Dublin City University, 1998.
- [55] P. J. Mulroy. Video content extraction: Review of current automatic segmentation algorithms. In *Workshop on Image Analysis for Multimedia Interactive Services 1997 (WIAMIS'97)*, June 1997.
- [56] K. Sobottka and I. Pitas. Segmentation and tracking of faces in color images. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 236–241, 1996.
- [57] O.J. Morris, M. de J. Lee, and A.G. Constantinides. Graph theory for the image analysis: an approach based on the shortest spanning tree. *IEE Proceedings*, 133(2):146–152, April 1986. pt. F.
- [58] Cooray S, O'Connor N, Marlow S, Murphy N, and Curran T. Hierarchical semi-automatic video object segmentation for multimedia applications. In *ITCOM'01*, August 2001.
- [59] E. Gose, R. Johnsonbaugh, and S. Jost. *Pattern Recognition and Image Analysis*. Prentice Hall PTR, 1996.
- [60] E. Chalom and V.M. Bove. Segmentation of an image sequence using multi-dimensional image attributes. In *Proceedings IEEE International Conference on Image Processing, ICIP'96*, volume 2, pages 525–528, Lausanne, September 1996.
- [61] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, (10):1090–1104, 10 2000.
- [62] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterisation of human faces. *J. Opt. Soc. Am. A*, 4(3):519–524, 3 1987.
- [63] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.
- [64] Marian S. Bartlett, Javier R. Movellan, and Terrence J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13(6), November 2002.
- [65] G. Shakhnarovich and B. Moghaddam. Face recognition in subspaces. In Stan Z. Li and Anil K. Jain, editors, *Handbook of Face Recognition*. Springer-Verlag, December 2004.

- [66] K. Kucharski and W. Skarbek. Face recognition methods – mpeg-7 perspective. In *International Workshop on Systems, Signals and Image Processing, IWSSIP*, September 2004.
- [67] Xiaofei He and Prtha Niyogi. Locality preseving projection. Technical report, Dept. of Computer Science, the University of Chicago, 2002. TR-2002-09.
- [68] Xiaofei He, Shuicheng Yan, Yuxiao Hu, and Hong-Jiang Zhang. Learning a locality preserving subspace for visual recognition. In *9th Int. Conf. on Computer Vision, ICCV'03*, 2003.
- [69] Tenenbaum J.B., Silva de V., and Langford J.C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290, December 2000.
- [70] Roweis S.T. and Saul L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290, December 2000.
- [71] Belkin M. and Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing System*, Vancouver, Canada, 2001.
- [72] M.-H. Yang. *Face Recognition Using Kernel Methods*, volume 14. T. Diederich and S. Becker and Z. Ghahramani, 2002.
- [73] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *IEEE Computer Vision and Pattern Recognition*, pages 84–91, 1994.
- [74] D.B. Graham and N.M. Allison. Face recognition from unfamiliar views: Subspace methods and pose dependency. In *Proceedings of FG'98*, 1998.
- [75] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33(11):1771–1782, November 2000.
- [76] T. Kanade. *Computer recognition of human faces*. Birkhauser, Basel, Switzerland and Stuttgart, Germany, 1973.
- [77] L. Wiskott, J.-M. Fellous, N. Krueger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):776–779, 1997.
- [78] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 12 2003.
- [79] A. Bronstein, M. Bronstein, and R. Kimmel. Expression-invariant 3d face recognition. In *Audio & Video-based Biometric Person Authentication*, 2003.

- [80] T. Vetter V. Blanz. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, September 2003.
- [81] P. Penev and J. Atick. Local feature analysis: A general statistical theory for object representation. *Netw.: Computat. Neural Syst.* 7, pages 477–500, 1996.
- [82] C. Liu and H. Wechsler. Evolutionary pursuit and its application to face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(6):570–582, 6 2000.
- [83] H. Alam, F. Rahman, Y. Tarnikova, and R. Hartono. A pair-wise decision fusion framework: Recognition of human fases. In *The 6th Int. Conf. on Information Fusion*, 2003.
- [84] *Information technology — Multimedia content description interface — Part 3: Visual*, 2002. ISO/IEC 15938-3:2002.
- [85] S. Cooray and B. Uscilowski. Effect of eye localization on the mpeg-7 face recognition descriptor. In *The 4th IASTED International Conference on Visualization, Imaging and Image Processing*, Marbella, Spain, 2004.
- [86] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, pages 47–60, November 1996.
- [87] Shi J. and Malik J. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [88] Frigui H. and Krishnapuram R. Clustering by competitive agglomeration. *Pattern Recognition*, 30(7):1109–1119, 1997.
- [89] Boujemaa N. On competitive unsupervised clustering. In *International Conference on Pattern Recognition, ICRP'00*, volume 1, pages 631–634, 2000.
- [90] B. Le Saux and N. Boujemaa. Unsupervised robust clustering for image database categorization. In *International Conference on Pattern Recognition, ICRP'2002*, 2002.
- [91] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. Commun.*, COM-28:84–95, January 1980.
- [92] Ball G.H. and Hall D.J. Isodata, a novel method of data analysis and classification. Technical report, Stanford University, Stanford, CA, 1965.

- [93] Zahn C.T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput. C-20*, April 1971.
- [94] Bentley J.L. and Friedman J.H. Fast algorithms for constructing minimal spanning trees in coordinate spaces. *IEEE Trans. Comput. C-27*, pages 97–105, June 1978.
- [95] Wu Z. and Leahy R. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.
- [96] Dempster A.P., Laird N.M., and Rubin D.B. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Stat. Soc. B. 39*, pages 1–38, 1977.
- [97] Zouhal L.M. and Denoeux T. An evidence-theoretic k-nn rule with parameter optimization. *IEEE Trans. on Systems, Man and Cybernetics-part C: Applications and Reviews*, 28(2):263–271, May 1998.
- [98] Zadeh L.A. Fuzzy sets. *Inf. Control 8*, pages 338–353, 1965.
- [99] Bezdek J. C. *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York, 1981.
- [100] G.F. Luger and W.A. Stubblefield. *Artificial Intelligence. Structures and Strategies for Complex Problem Solving*. Addison-Wesley Longman, Inc., 1998.
- [101] Kohonen T. *Self-Organization and Associative Memory*. Springer Information Series, Springer-Verlag, New York, NY, 3 edition, 1989.
- [102] Pal N.R., Bezdek J.C., and Tsao E.C-K. Generalized clustering networks and kohonen’s self organizing scheme. *IEEE Trans. on Neural Networks*, (4):549–557, 1993.
- [103] Mao J. and Jain A.K. A self-organizing network for hyperellipsoidal clustering (hec). *IEEE Trans. on Neural Networks*, (7):16–29, 1994.
- [104] Raghavan V.V. and Birchand K. A clustering strategy based on formalism of the reproductive process in a natural system. In *Proc. of the Second Int. Conf. on Information Storage and Retrieval*, pages 10–22, 1979.
- [105] Bhuyan J.N. and Raghavan V.V. Genetic algorithm for clustering with an ordered representation. In *Proc. of the Fourth Int. Conf. on Genetic Algorithms*, pages 408–415, 1991.

- [106] Jones D. and Beltramo M.A. Solving partitioning problems with genetic algorithms. In *Proc. of the Fourth Int. Conf. on Genetic Algorithms*, pages 442-449, 1991.
- [107] Capelle A.-S., Fernandez-Maloigne C., and Colot O. Segmentation of brain tumors by evidence theory: on the use of the conflict information. In *The 7th Int. Conf. on Information Fusion*, 2004.
- [108] P.F. Singer. The fusion of parametric and non-parametric hypothesis tests. In *The 6th Int. Conf. on Information Fusion*, 2003.
- [109] B. van Ginneken M. Loog. Static posterior probability fusion for signal detection. applications in the detection of intersistitial diseases in chest radiographs. In *The 6th Int. Conf. on Information Fusion*, 2003.
- [110] P. Smets. Data fusion in the transferable belief model. In *ISIF*, 2000.
- [111] Smets P. *Non-standard logics for automated reasoning*, chapter Belief Functions. 1998.
- [112] T.M. Shuck, M. Freisel, and J.B. Hunter. Information properties as a means to define decision fusion methodologies in non-benign environments. In *The 6th Int. Conf. on Information Fusion*, 2003.
- [113] D. Hackerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Tehnology Division, Redmond, WA, USA, 1995.
- [114] Matthew Cooper and Jonathan Foote. Discriminative techniques for keyframe selection. In *IEEE Int. Conf. on Multimedia and Expo, ICME*, 2005.
- [115] A. Girgensohn and J. Boreczky. Time-constrained keyframe selection technique. In *IEEE Int. Conf. on Multimedia Computing and Systems*, 1999.
- [116] D. Diklic, D. Petkovic, and Danielson R. Automatic extraction of representative keyframes based on scene content. In *Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems & Computers*, 1998.
- [117] S. Hasebe, M. Nagumo, S Muramatsu, and H. Kikuchi. Video key frame selection by clustering wavelet coefficients. In *European Signal Processing Conference, EUSIPCO*, 2004.
- [118] Tianming Liu, Hong-Jiang Zhang, and Feihu Qin. A novel video key-frame-extraction algorithm based on perceived motion energy model. *IEEE Trans. on Circuits and Systems for Video Technology*, 13(10), October 2003.

- [119] Marcus J. Pickering and Stefan R ger. Evaluation of key frame-based retrieval techniques for video. *Computer Vision and Image Understanding*, 92:217–235, 2003.
- [120] Yueting Zhuangt, Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Int. Conf. on Image Processing, ICIP*, pages 866 – 870, 1998.
- [121] *Overview of the MPEG-7 Standard*, July 2002. ISO/IEC JTC1/SC29/WG11 N4980.

Appendix A

Derivations required for Expectation-Maximisation algorithm

A.1 Maximum Likelihood estimation

The representation of the object and its components proposed in Section 3.2.4 is very elegant and convenient, however there are some issues that have to be considered for the efficient use of this representation. The main problem is how to find the modes that would best represent the object or the components. There are many classification techniques which can be used for searching for the proper modes parameters such as statistical classifiers, nearest neighbour classifiers, clustering techniques or artificial neural networks [59]. Since the probabilistic representation of the object is chosen, the most convenient solution is probabilistic. A sophisticated and convenient method is based on the estimation of the Maximum Likelihood (ML) using Bayes theory. The iterative approach to such an estimation is called the Expectation-Maximisation (EM) algorithm since it maximises the expectation of the likelihood function.

The model of the image consisting of a mixture of PDFs was presented in Section 3.2.4. Let us use this model of an image and assume that each PDF is a likelihood function for each object (or region within the object), so the likelihood function $l_x(\theta)$ with the parameters θ can be denoted as [54]:

$$l_x(\theta) = \text{pdf}_m(x, \theta), \tag{A.1}$$

where $\text{pdf}_m(x, \theta)$ is the probability that the x th pixel belongs to the m th object with the parameters θ . The logarithmic form of the likelihood function $L_x(\theta) = \ln l_x(\theta)$, naturally named the *log-likelihood* function is commonly used instead of the likelihood function, since the logarithmic function is monotonic and the maximum

of the log-likelihood function appears at the same point as the likelihood function and any constant value is removed [54].

The calculation of the Maximum Likelihood requires the complete set of data to be available. However, one is very unlikely to capture the complete dataset and calculate initial PDFs, which would give the Maximum Likelihood for the image. Thus the Maximum Likelihood is estimated through maximising the expectation of the likelihood function or the expectation of the log-likelihood function. The initial set of PDFs representing the image can be adjusted iteratively to the data of the image using the criterion of the maximum expectation of the log-likelihood function, what is denoted as:

$$Q(\theta|\theta_k) = E [L_x(\theta), \theta_k], \quad (\text{A.2})$$

where θ_k denotes the parameters of the PDF in the k th iteration. The calculation of the likelihood function expectation is called the expectation step (E-step). However, the set of parameters maximising the expectation $Q(\theta|\theta_k)$ is to be found. Thus the set of the parameters for the next iteration should maximise the value of the expectation of the likelihood function [86]:

$$\theta_{k+1} = \arg \max_{\theta} Q(\theta|\theta_k). \quad (\text{A.3})$$

This step is called the Maximisation Step (M-step). Both steps the E-step and the M-step are repeated while the model converges towards the Maximum Likelihood at each iteration. It can be assumed that the convergence is met when the change of the parameters is very small [86]:

$$\|\theta_k - \theta_{k+1}\| < \varepsilon. \quad (\text{A.4})$$

In the case of image processing, the iterations may be stopped when a very small number of pixels change their object membership.

A.2 Maximum Likelihood estimation for the mixture of Gaussian distributions

Let us consider the mixture of M multimodal Gaussian PDFs

$$\text{pdf}_m(x, \theta), \quad m = 1 \dots M, \quad (\text{A.5})$$

where each of these PDFs is obtained with the Equation 3.15 as the sum of G multivariate Gaussian distributions $\text{pdf}_g(x, \theta)$, $g = 1 \dots G$. The multivariate Gaussian

PDF is described with Equation 3.13 and depends on the parameters vector

$$\theta = [\theta_1 \ \theta_2], \quad (\text{A.6})$$

where

$$\theta_1 = [m_1 \ m_2 \ \dots \ m_k]^T \quad (\text{A.7})$$

is a vector of mean values of the k features, and θ_2 is a covariance matrix, which if the components are independent becomes a diagonal matrix of variances as in Equation 3.14. The means and variances of the features are the parameters, which are to be calculated in order to find the estimation of the maximum likelihood.

A.2.1 The likelihood maximisation of the unimodal multivariate Gaussian PDF

Given the initial set of data representing the modes and the object (which are not complete due to the limited knowledge about the overall distribution) the initial parameters maximising the likelihood function can be calculated. Let us firstly calculate the mean values θ_1 . The maximum likelihood regarding the θ_1 meets the condition [86, 54]:

$$\frac{\partial}{\partial \theta_1} L_x(\theta) = 0. \quad (\text{A.8})$$

The log-likelihood function $L_x(\theta)$ for the N observations of the unimodal multivariate Gaussian distribution with k components is [54]:

$$\begin{aligned} L_x(\theta) &= \ln(\text{pdf}_g(x, \theta)) \\ &= \ln \left(\frac{1}{(2\pi)^{\frac{Nk}{2}} |\theta_2|^{\frac{N}{2}}} e^{-\frac{1}{2} \sum_{n=1}^N (x_n - \theta_1) \theta_2^{-1} (x_n - \theta_1)} \right), \end{aligned} \quad (\text{A.9})$$

where $x_n = [x_{n1} \ x_{n2} \ \dots \ x_{nk}]^T$ is a vector of k components of the n th observation, and $\theta_1 = [m_1 \ m_2 \ \dots \ m_k]^T$ denotes the vector of mean values of the features. If the components are independent the θ_2 is defined with Equation 3.14 and the log-likelihood function becomes:

$$\begin{aligned} L_x(\theta) &= \ln \left((2\pi)^{-\frac{Nk}{2}} \left(\prod_{l=1}^k \sigma_l^{-N} \right) e^{-\frac{1}{2} \sum_{l=1}^k \sum_{n=1}^N \frac{(x_{nl} - m_l)^2}{\sigma_l^2}} \right) \\ &= -\frac{Nk}{2} \ln(2\pi) - N \sum_{l=1}^k \ln(\sigma_l) - \frac{1}{2} \sum_{l=1}^k \sum_{n=1}^N \frac{(x_{nl} - m_l)^2}{\sigma_l^2} \end{aligned} \quad (\text{A.10})$$

In order to obtain the Maximum Likelihood estimation the l th component of the vector θ_1 fulfils the condition (according to A.8):

$$\begin{aligned} \frac{\partial}{\partial m_l} L_x(\theta) &= 0 \\ -\frac{1}{2} \frac{-2}{\sum_{n=1}^N \sigma_l^2} \sum_{n=1}^N (x_{nl} - m_l) &= 0 \\ \sum_{n=1}^N x_{nl} - Nm_l &= 0 \end{aligned} \quad (\text{A.11})$$

what gives eventually

$$m_l = \frac{1}{N} \sum_{n=1}^N x_{nl}. \quad (\text{A.12})$$

Assuming that the components are independent, the l th component of the covariance matrix θ_2 fulfils the condition:

$$\frac{\partial}{\partial \sigma_l} L_x(\theta) = 0. \quad (\text{A.13})$$

Using Equation A.10 in the condition given by Equation A.13, the σ_l is obtained:

$$\begin{aligned} -\frac{N}{\sigma_l} + \frac{2}{2} \frac{1}{\sigma_l^3} \sum_{n=1}^N (x_{nl} - m_l)^2 &= 0 \\ \sigma_l^2 &= \frac{1}{N} \sum_{n=1}^N (x_{nl} - m_l)^2 \end{aligned} \quad (\text{A.14})$$

A.2.2 The likelihood estimation of the multimodal PDF

Once the initial parameters are obtained, the multimodal distribution is calculated with Equation 3.15. The weighting factor π_{gn} is considered as the *prior* probability that the observation x_n , $n = 1, 2, \dots, N$ belongs to the g th class (in the case of an image, the g th region). Since the parameters of each region's PDF have been changed due to the calculation of the parameters maximising the Maximum Likelihood, we must obtain the probability (in terms of the likelihood) that the given observation belongs to the certain class. This probability is called the *posterior* probability and is obtained using the Bayes rule [54, 59]:

$$\tau_{gn} = \frac{\pi_{gn} \text{pdf}_g(x_n, \theta_g)}{\sum_{g=1}^G \pi_{gn} \text{pdf}_g(x_n, \theta_g)}, \quad (\text{A.15})$$

The estimated log-likelihood function of the multimodal PDF is obtained as the weighted sum of the log-likelihood functions of the unimodal distributions [54]:

$$L_x(\theta) = \sum_{g=1}^G \sum_{n=1}^N \tau_g \ln(\text{pdf}_g(x_n, \theta_g)) \quad (\text{A.16})$$

where τ_g is the *posterior* probability that the n th observation x_n belongs to the class (region) g . The variable G denotes the number of modes of which the PDF consists, and N is the number of observations belonging to the g th mode. τ_{gn} is a probability that the n th observation belongs to the g th mode described with the parameters θ_g .

A.2.3 The likelihood maximisation of the multimodal PDF

The likelihood of the multimodal PDF is maximised when the derivative of the log-likelihood function from the Equation A.16 becomes zero:

$$\frac{d}{d\theta} L_x(\theta) = 0. \quad (\text{A.17})$$

The region log-likelihood function is given by the equation:

$$\ln(\text{pdf}(x_n, \theta_g)) = \left[-\frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln|\theta_{g2}| - \frac{1}{2} (x_n - \theta_{g1})^T \theta_{g2} (x_n - \theta_{g1}) \right], \quad (\text{A.18})$$

where $\theta_{g1} = [m_{g1} \ m_{g2} \ \dots \ m_{gk}]^T$ is the vector of the mean values of the components representing the g th region and the θ_{g2} is the covariance matrix of region g . Using Equations A.17, A.16 and A.18, the value of the l th component of the mean vector can be calculated (with the assumption that the components are independent):

$$\begin{aligned} \sum_{n=1}^N \tau_{gn} \frac{\partial}{\partial m_{gl}} \left[-\frac{1}{2} \sum_{j=1}^k \frac{(x_{nj} - m_{gj})^2}{\sigma_{gj}^2} \right] &= 0 \\ - \sum_{n=1}^N \tau_{gn} x_{nl} + m_{gl} \sum_{n=1}^N \tau_{gn} &= 0. \end{aligned} \quad (\text{A.19})$$

This leads to the value of the m_{gl} :

$$m_{gl} = \frac{\sum_{n=1}^N \tau_{gn} x_{nl}}{\sum_{n=1}^N \tau_{gn}}. \quad (\text{A.20})$$

Similarly the values of the variances σ_{gl}^2 maximising the likelihood function of the g th region can be calculated as follows (again assuming that the components are independent):

$$\begin{aligned} \sum_{n=1}^N \tau_{gn} \frac{\partial}{\partial \sigma_{gl}^2} \left[-\ln \left(\prod_{j=1}^k \sigma_{gj} \right) - \frac{1}{2} \sum_{j=1}^k \frac{(x_{nj} - m_{gj})^2}{\sigma_{gj}^2} \right] &= 0 \\ -\sigma_{gl}^2 \sum_{n=1}^N \tau_{gn} + \sum_{n=1}^N \tau_{gn} (x_{ni} - m_{gl})^2 &= 0. \end{aligned} \quad (\text{A.21})$$

This gives the variance value:

$$\sigma_{gl}^2 = \frac{\sum_{n=1}^N \tau_{gn} (x_{ni} - m_{gl})^2}{\sum_{n=1}^N \tau_{gn}}, \quad (\text{A.22})$$

Appendix B

MPEG-7 FaceRecognition Descriptor

The MPEG-7 standard provides tools for the description of the multimedia content. The Face Recognition Descriptor is a descriptor that can be used for searching for the face images which contain a face similar to the face on the query image [121].

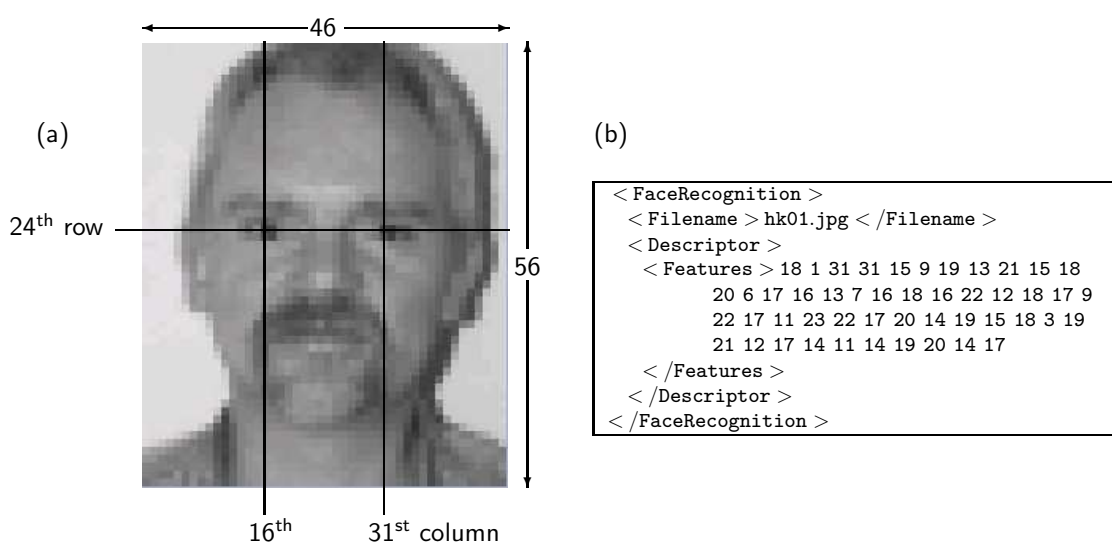


Figure B.1: (a) An example of the normalised facial image and (b) an example of the description scheme containing an extracted descriptor represented in the XML format.

The Face Recognition descriptor is extracted from the normalised facial image which consists only of the luminance values. The normalised facial image has 56 rows and 46 columns with the face placed such that the eyes are located on the 24th row and the 16th and 31th column. This image is then scanned column-wise, what gives the one-dimensional vector of the luminance values. The one-dimensional vector is then projected onto the space of the basic matrix defined by the standard [121, 84]. The descriptor itself is a vector containing 48 coefficients,

each coefficient stored as a 5 bit integer, thus the size of the descriptor is 30 bytes[84]. The representation of the descriptor can be either binary or textual (the XML format). The example of the normalised facial image is presented in Figure B.1. The example of the description scheme containing the Face Recognition descriptor stored in the XML format is shown on the same figure.

The distance measure used for the matching and searching for the similar faces is the weighted Manhattan distance (the Minkowski distance with $p = 1$). The weights are described in the part 8 of the MPEG-7 standard [84].

Appendix C

Evaluation

C.1 Analysis of clusters

A method for analysing clusters obtained by clustering techniques described in this thesis is presented in this section. The clusters obtained in each experiment were analysed using the following criteria:

- the number of all clusters
- the number of valid clusters
- the average precision
- the average recall.

These criteria are best described by the example presented below and by the detailed definitions in Appendixes C.4 and C.3.

Let us consider sample clustering results shown in Figure C.1. There are six clusters in total. However, four of them contain only one image each. Therefore, these clusters are seen as containing outliers and are discarded from further analysis. The remaining two clusters (C3 and C4) contain more than one face each, therefore they are “valid” clusters. Further analysis is carried out on valid clusters only, because they are able to provide valuable information. The detailed definition of valid clusters is presented in Appendix C.4.

The average precision and recall are used for analysing the structure of clusters. The values of precision and recall should be obtained only from clusters that do not contain outliers, in our example these are clusters C3 and C4. Otherwise, the outliers would affect the analysis, leading to the poor analysis. In the cluster C4, the images of John occur at most, therefore John’s identity is regarded as the label of this cluster. Because there are three images of John, out of four images in total in this cluster, the precision for this cluster is 0.75. The total number of the images of John in the collection is 5, therefore the value of recall for this cluster is 0.6. Similarly, in the cluster C4 Mary’s identity is most frequent and the values












C1						
	Unknown					
C2						
	Mary					
C3						precision=0.667 recall=0.667
	Mary	Mary	John			
C4						precision=0.75 recall=0.600
	John	John	John	Unknown		
C5						
	Unknown					
C6						
	John					

Figure C.1: Sample clusters — the example for analysing results

Table C.1: Annotated features

user	owner of a photograph
time	year, month, day, hour, minute, second
GPS location	longitude, latitude,
exposition	aperture, shutter speed, flash on/off
name of place	based on GPS location
weather	based on GPS location and time
time of day	down, noon, dusk, night
indoor/outdoor	based on GPS location and exposition
event	based on location and time (auto)
building	present/not present (manual)
no of people	person/group/crowd (manual)
face region	a bounding box (manual)
identity	associated with a face (manual)

of precision and recall are both 0.667. The average precision for this clustering example is thus 0.709 and the average recall is 0.634. The detailed definitions of precision and recall are presented in Appendix C.3.

C.2 Dataset

The dataset used for experiments consists of 11288 photographs captured by 16 different people, of all sort of content and taken at various locations and events, i.e. holiday photos, party photos, landscapes, cityscapes, crowd, sport events, etc. The subset of the photographs, which were manually annotated, containing 1127 facial images in various poses, expressions, lighting conditions and resolutions was chosen for experiments. Usually, experiments for face recognition take into account just one of the mentioned above factors. However, in real life photographs all these factors are mixed, therefore the recognition can be expected to be poor, even when using the most sophisticated techniques that give excellent results in a controlled environment [23].

The photographs were already organised and classified into events. This was carried out using the algorithm by O'Hare *et al* [16] based on the location and time of capturing the image. Time information is stored by modern photo cameras in the EXIF format in a digital photograph file. The GPS aware photo cameras also save, in the EXIF format, the geographic position of the camera at the moment of taking a picture. Therefore both sources of information might be available. The photographs in the database where stamped with the location coordinates obtained from external GPS devices, which were used with photo cameras not equipped with a built-in GPS receiver.

The images used by the author for our experiments were manually annotated with identities and manually located faces (each face was marked with a bounding

Table C.2: Statistics on the large dataset

total no of facial images	1127
no of identities	54
total no of outliers	649
frontal images	1127
no of outliers in frontal	649
smallest face	21x20
largest face	824x852
average face size	123x122
number of users	15
number of events	434

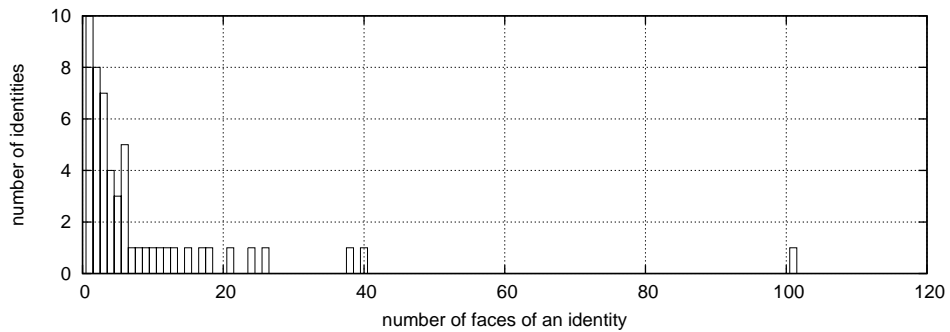


Figure C.2: The histogram of the occurrences of identities in the large dataset

box). The annotations were used as the ground truths for the evaluation of clustering results. This collection of annotated facial images contains 649 outliers, i.e. the faces annotated as “unknown” persons. Such outliers are not uncommon in personal photographs and provide more of a challenge to the clustering algorithm, as they might cause noisy and highly inaccurate results. Therefore, there is a need to identify them and remove from clusters during the clustering process. There are 54 identities in total including the “unknown” identity, therefore one can expect ideally 53 valid clusters (when excluding the clusters of outliers). The statistics of this collection and the histogram of the number of the faces of each identity are presented in Table C.2 and Figure C.2 respectively.

Some initial experiments were carried out by the author using a private, very small collection. These experiments lead to the observation that narrowing the search space improves the clustering accuracy. This private collection is the small part of the test collection. Statistics of the private collection are presented in Table C.3 and the histogram of the number of the occurrences of each identity is shown in Figure C.3.

Table C.3: Statistics on a small private dataset

total no of facial images	95
no of identities	7
total no of outliers	30
frontal images	68
no of outliers in frontal	13
smallest face	35x25
largest face	442x429
average face size	118x121
number of users	1
number of events	13

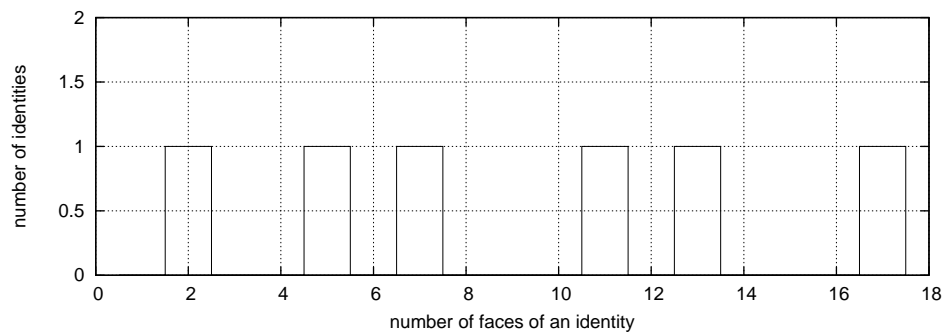


Figure C.3: Histogram of occurrences of identities in the small dataset

C.3 Precision and recall

The measures used for evaluating the “goodness” of clustering process are precision and recall. Let us consider a cluster with N faces of K identities, $K < N$. Let there be $N_k, k = 1, \dots, K$ occurrences of the k th identity in the cluster. Let us sort them in such a way that $N_1 > N_2 > \dots > N_K$, or find an identity with the highest number of occurrences N_h in the cluster such as $N_h > N_k \forall k = 1, \dots, K, k \neq h$, and $h = 1$. Let us also assume that there are M_k occurrences of the k th identity in the whole collection. The precision for the cluster is defined as:

$$p = N_h/N \quad (\text{C.1})$$

and the recall

$$r = N_h/M_h. \quad (\text{C.2})$$

For the whole collection the mean values of precision \bar{p} and recall \bar{r} are calculated:

$$\bar{p} = \frac{1}{C} \sum_{c=1}^C p_c \quad (\text{C.3})$$

$$\bar{r} = \frac{1}{C} \sum_{c=1}^C r_{c\cdot}, \quad (\text{C.4})$$

where C denotes the number of valid clusters in the collection.

The precision p indicates how good a cluster is in terms of the consistency of identities within the cluster, e.g. if the cluster consists of eight images of one person and two images of two other people, than the precision is 0.8. The recall measure r indicates how many instances of all instances of the given person were captured within the cluster. In other words this is the measure of the spread of the images of the same person across all clusters, e.g. if there are ten images of the person within the collection and the cluster contains five of them, than the recall value is 0.5. Ideally, the clustering process should produce clusters consisting only of the faces of one person (precision= 1) containing all images of this person available in the collection (recall= 1).

C.4 Definition of valid clusters

All calculations of precision and recall values were carried out using only the “valid” clusters. By definition the “valid” clusters are:

Definition: Valid cluster is a cluster that:

- consists of at least two members (faces)
- the identity of all occurrences in the cluster is not an “Unknown” identity.

In consequence of such a definition, all clusters containing just a single member are not considered to be valid. The clusters with only one member are considered as outliers. Therefore, should not be taken into account for further analysis.

Similarly, if the identity of all elements in a cluster is the “Unknown” person, than such a cluster should be removed from the analysis as many different persons, which are not interesting for users, were annotated with this identity. The number of persons in the cluster named as “Unknown” does not mean that just a person of this name is classified to this cluster, it is more likely that the faces of several different people labelled as “Unknown” were classified to this cluster. There is no way to evaluate this, thus such clusters should be removed from the analysis and are not viewed as the valid clusters.

Appendix D

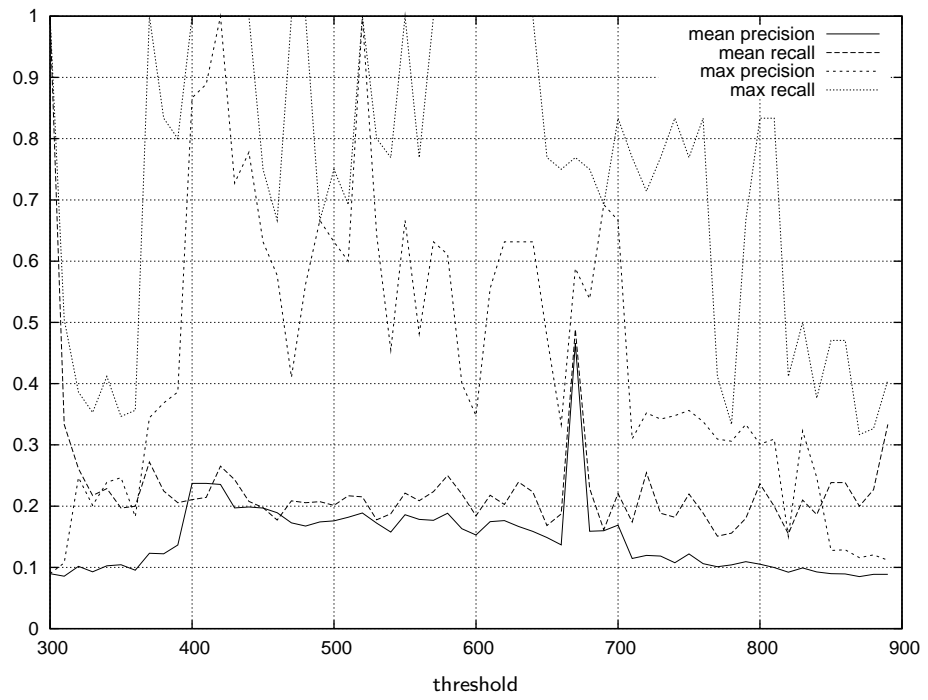
Detailed results — clustering

Detailed graphs for analysis of clustering methods presented in Chapter 5 are presented in this appendix.

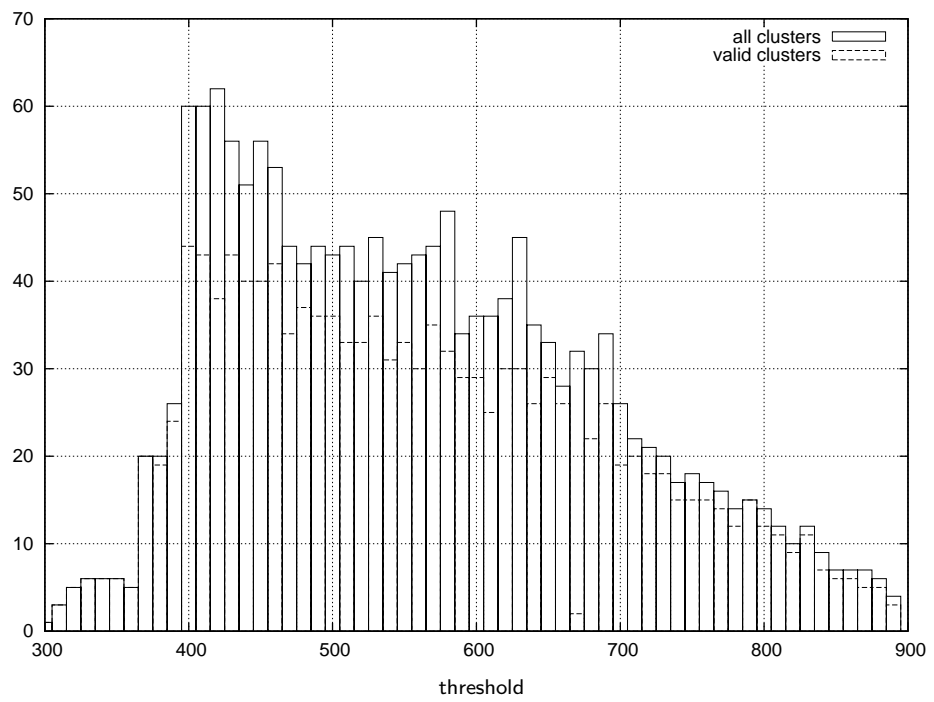
D.1 K-means approach

D.1.1 Classic similarity measure

Figures D.1 - D.8 present analysis of clustering results using a modified k-means algorithm with a MPEG-7 FR similarity measure. The graphs show values of average and maximum precision, average and maximum recall, numbers of all created clusters and numbers of valid clusters for different scenarios and values of thresholds.

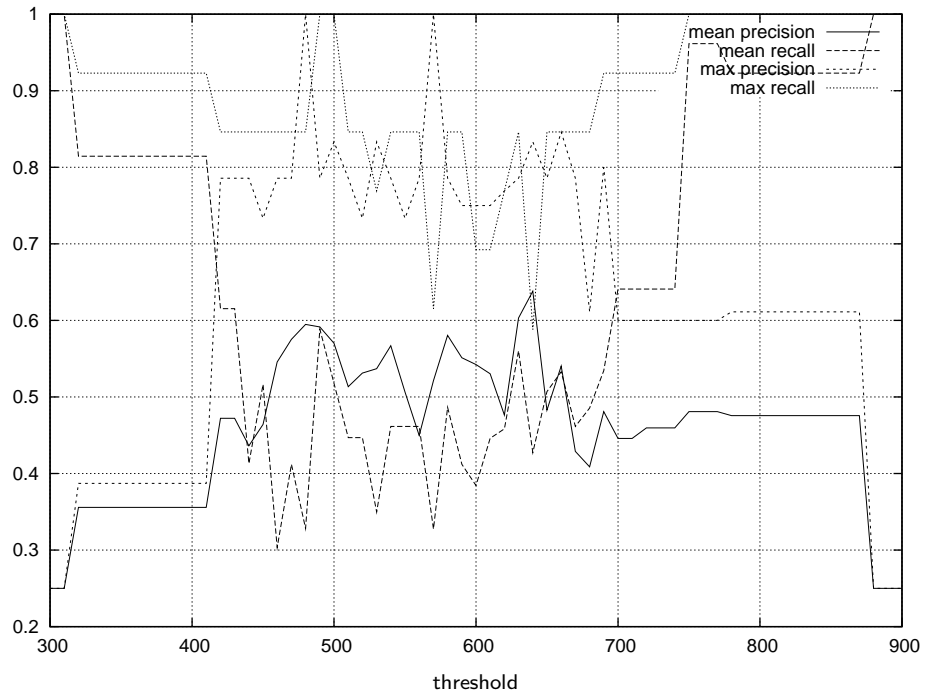


(a)

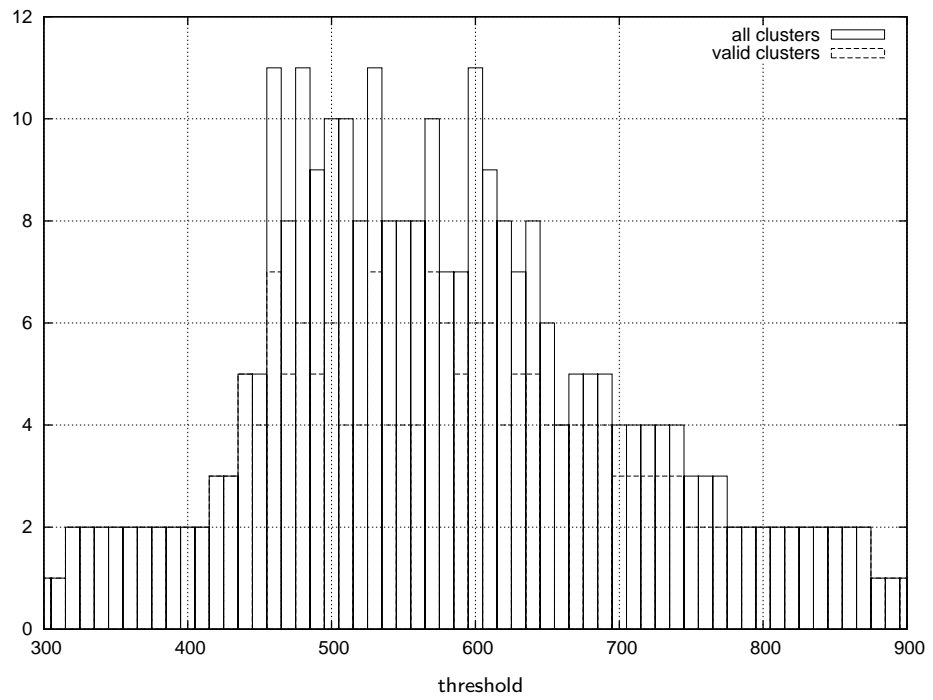


(b)

Figure D.1: Precision and recall for clustering with modified k-means algorithm using data set of 1127 faces with outliers; (a) precision and recall, (b) number of clusters.

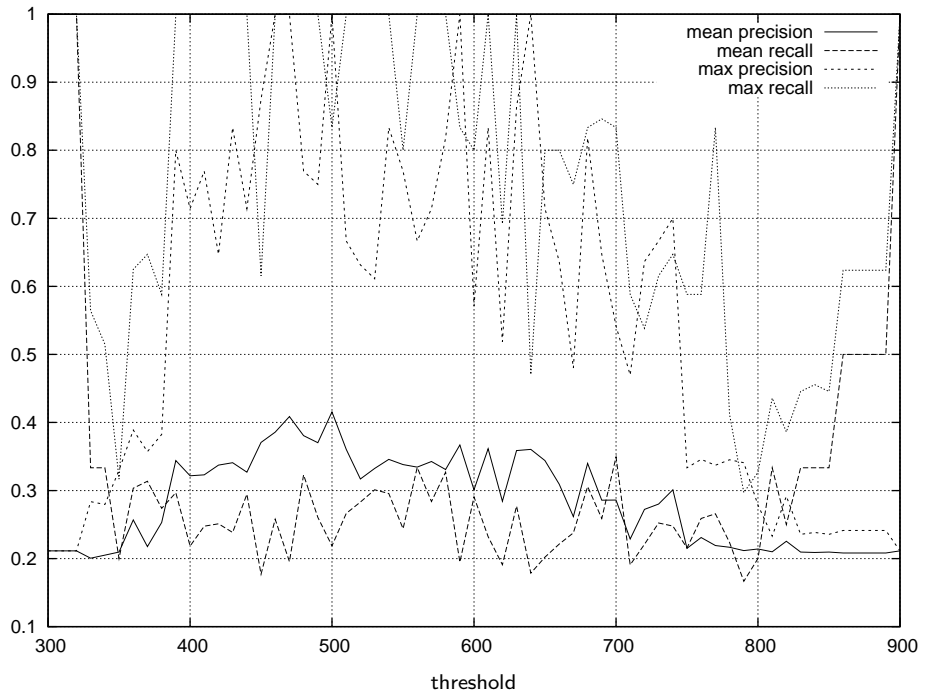


(a)

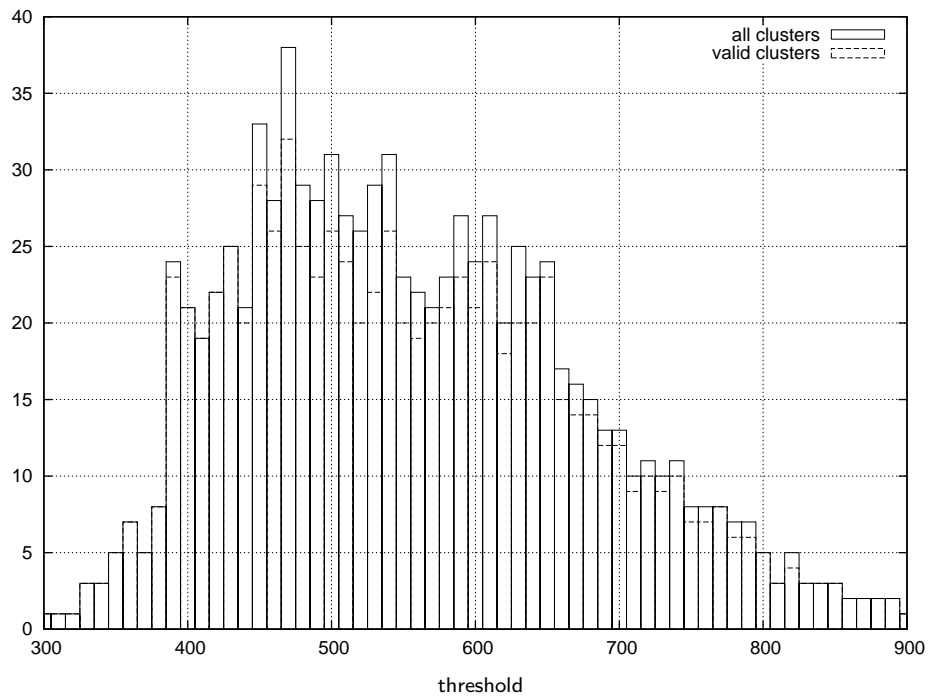


(b)

Figure D.2: Precision and recall for clustering with modified k-means algorithm using small data set of 68 faces with outliers; (a) precision and recall, (b) number of clusters.

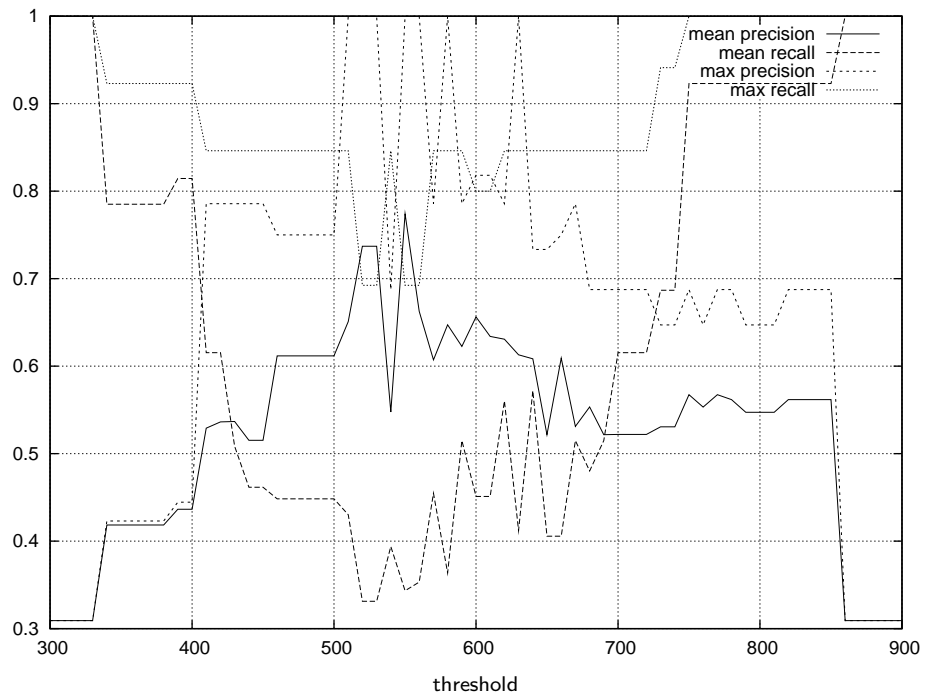


(a)

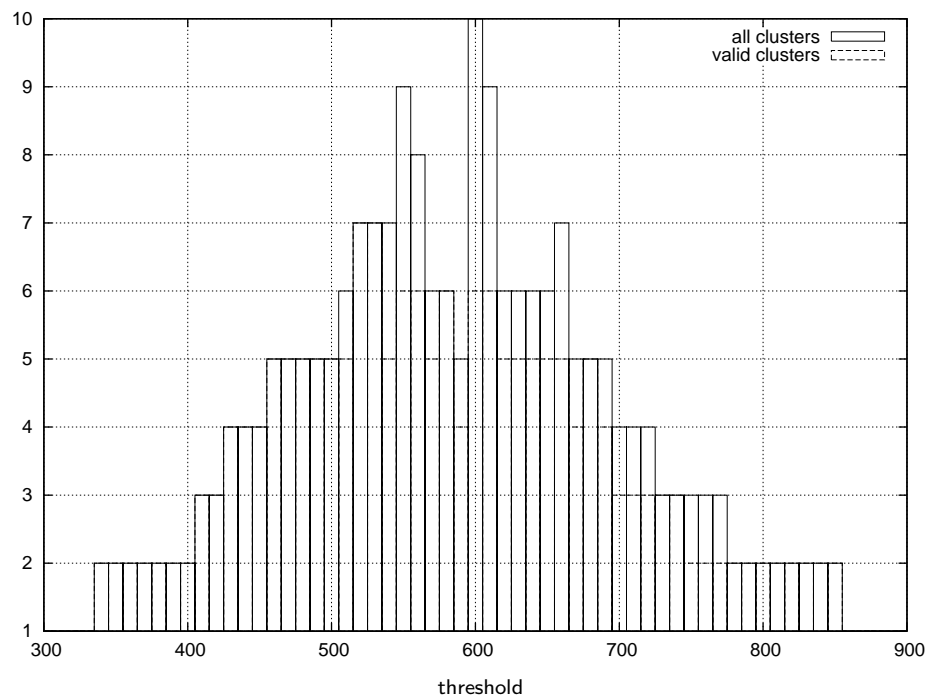


(b)

Figure D.3: Precision and recall for clustering with modified k-means algorithm using data set of 478 faces without outliers; (a) precision and recall, (b) number of clusters.

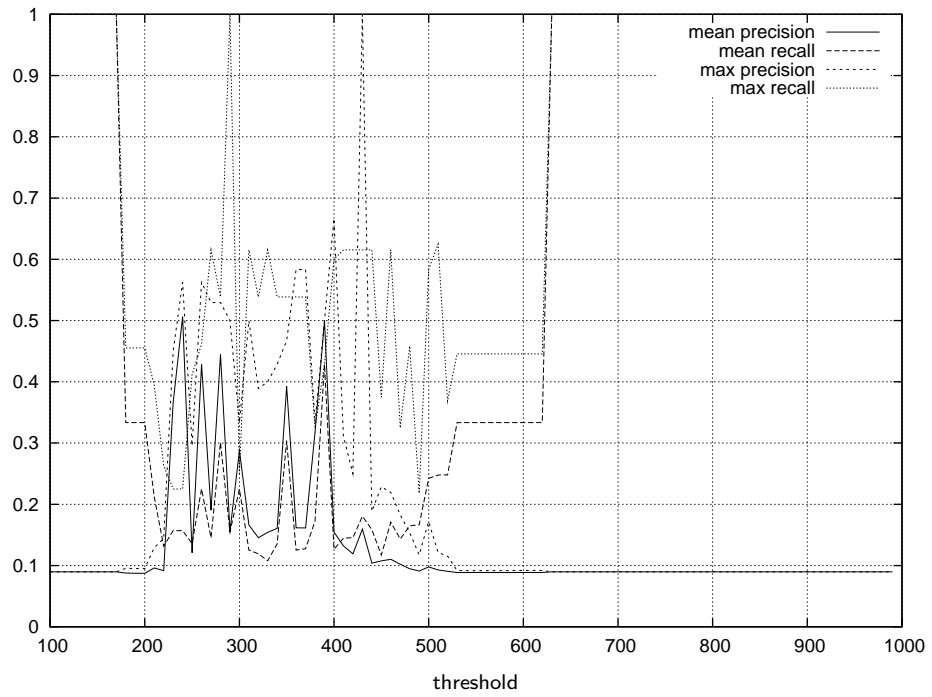


(a)

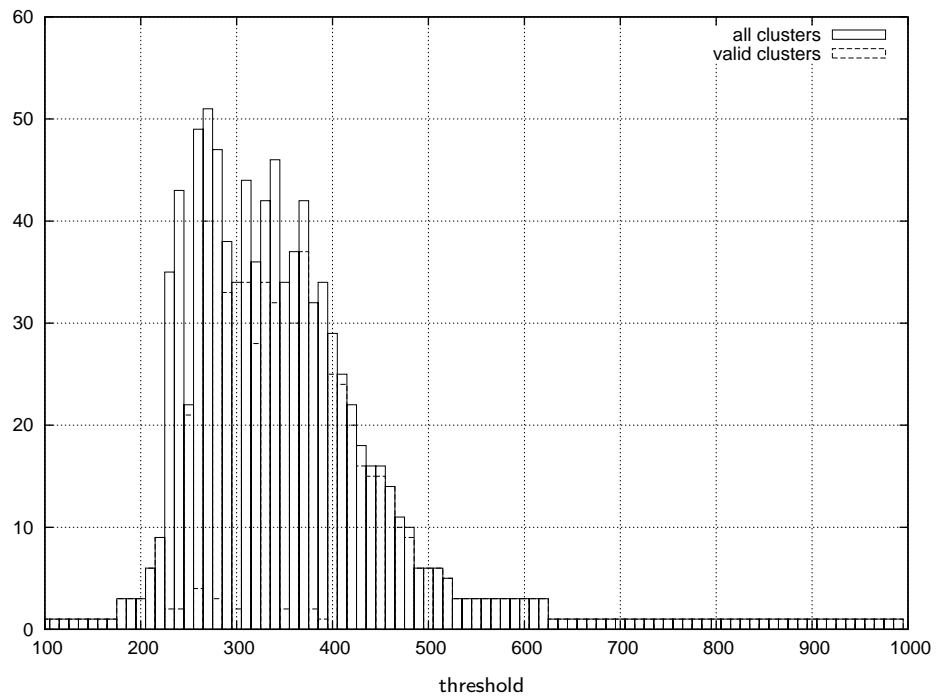


(b)

Figure D.4: Precision and recall for clustering with modified k-means algorithm using small data set of 55 faces without outliers; (a) precision and recall, (b) number of clusters.

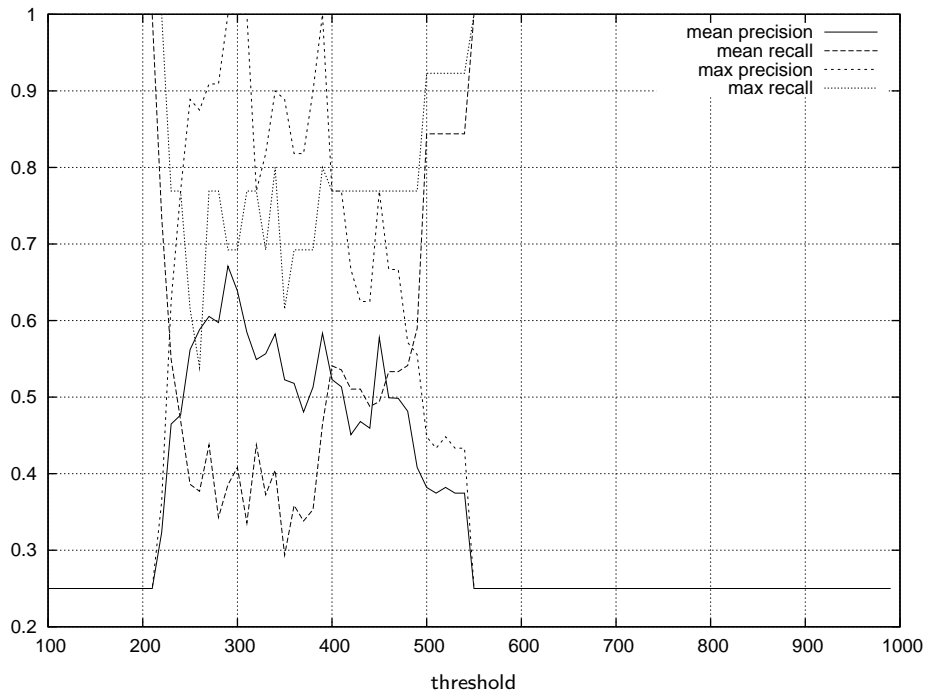


(a)

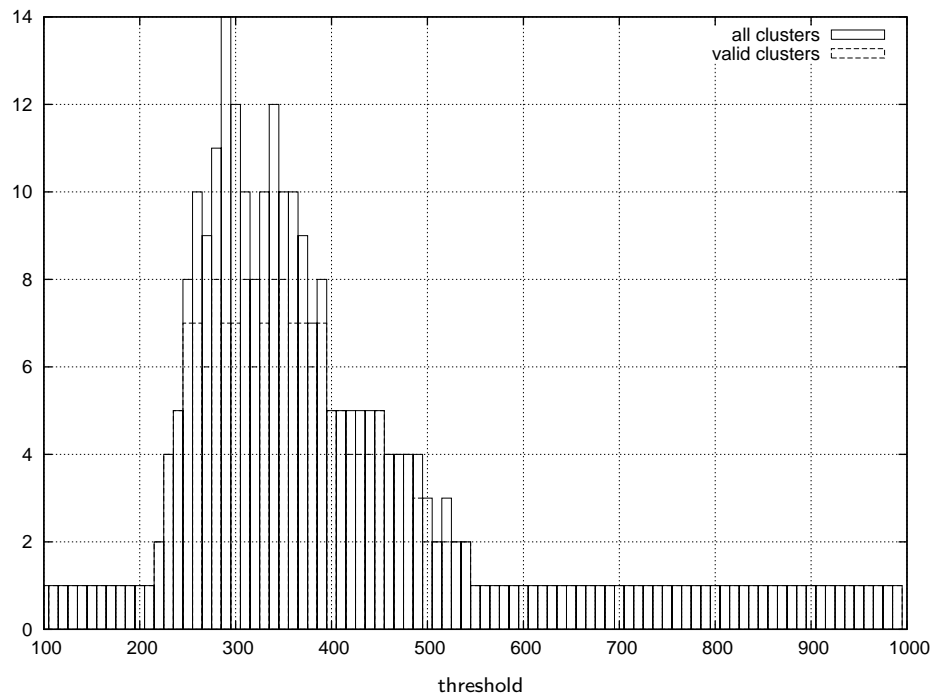


(b)

Figure D.5: Precision and recall for clustering with modified k-means algorithm and automatically located eyes using data set of 1127 faces with outliers; (a) precision and recall, (b) number of clusters.

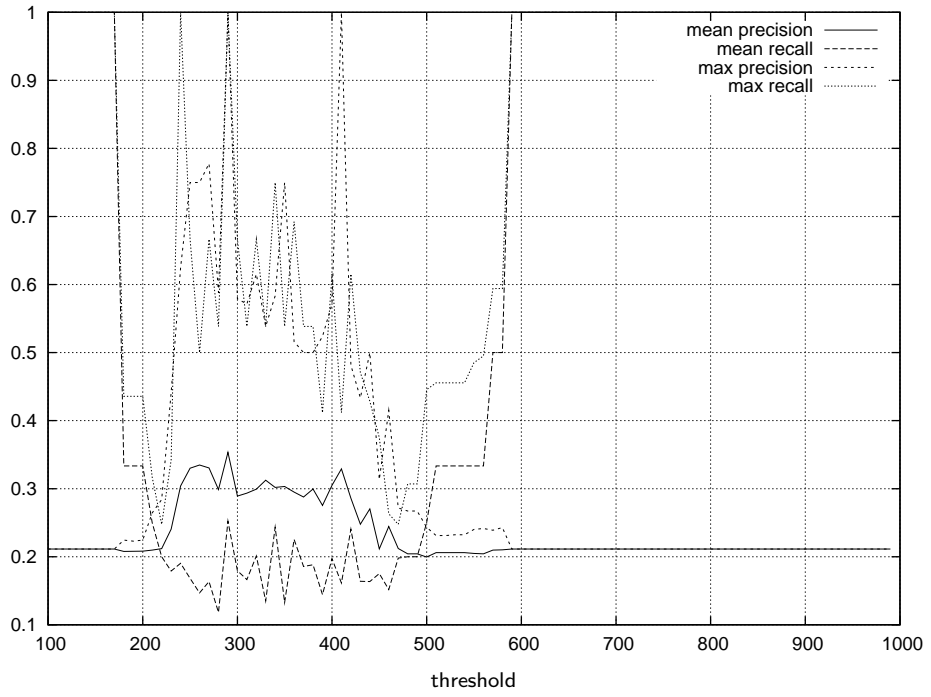


(a)

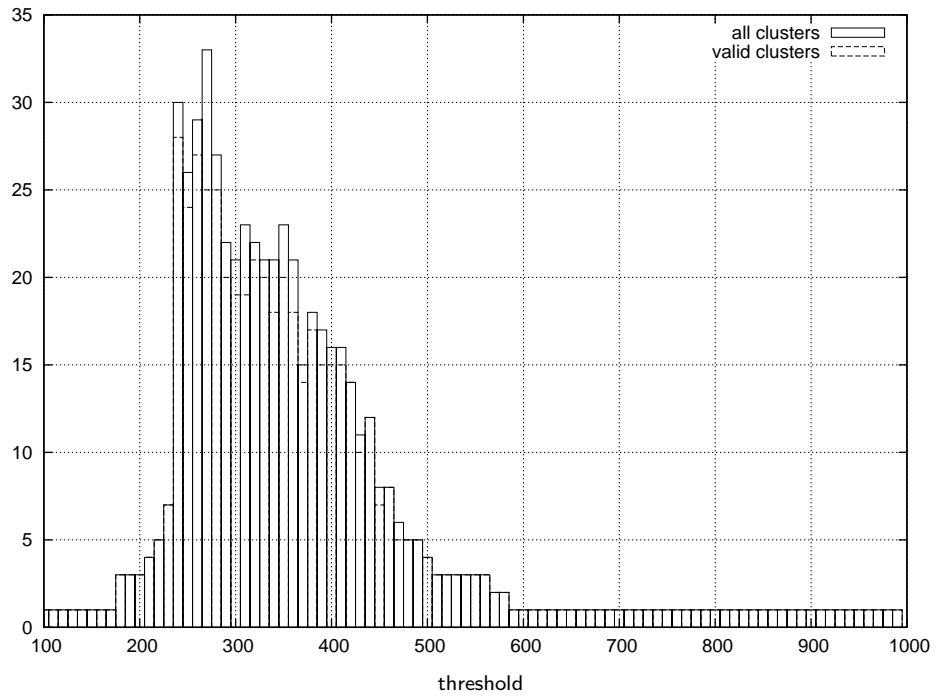


(b)

Figure D.6: Precision and recall for clustering with modified k-means algorithm and automatically located eyes using small data set of 68 faces with outliers; (a) precision and recall, (b) number of clusters.

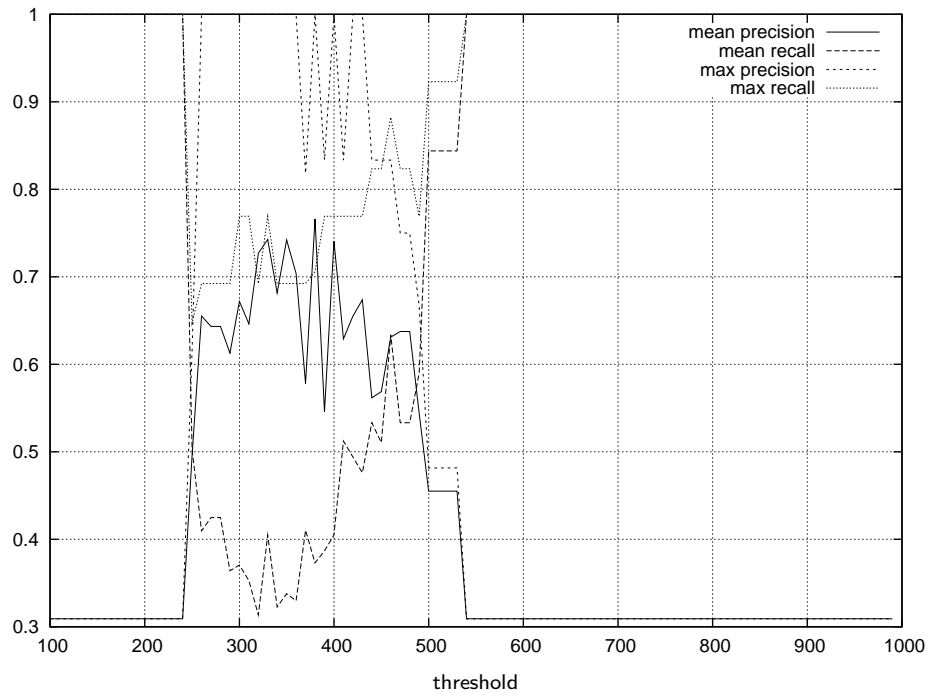


(a)

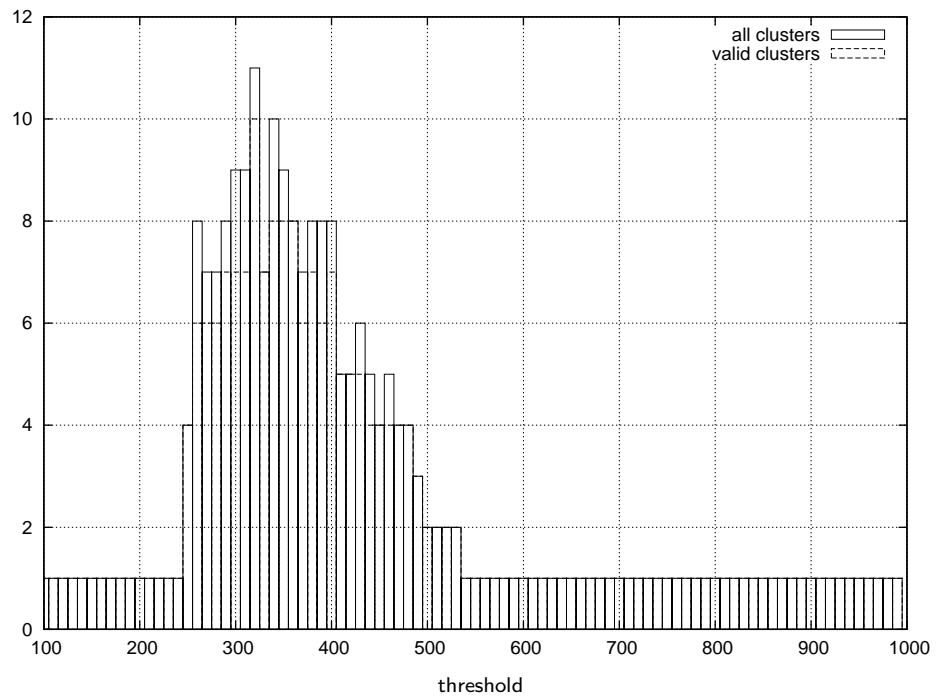


(b)

Figure D.7: Precision and recall for clustering with modified k-means algorithm and automatically located eyes using data set of 478 faces without outliers; (a) precision and recall, (b) number of clusters.



(a)

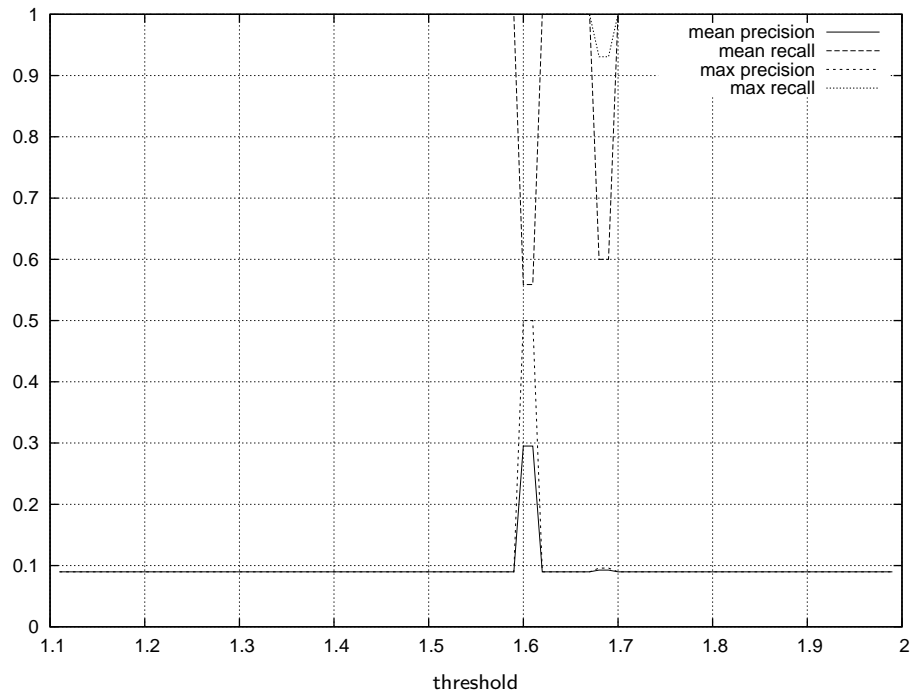


(b)

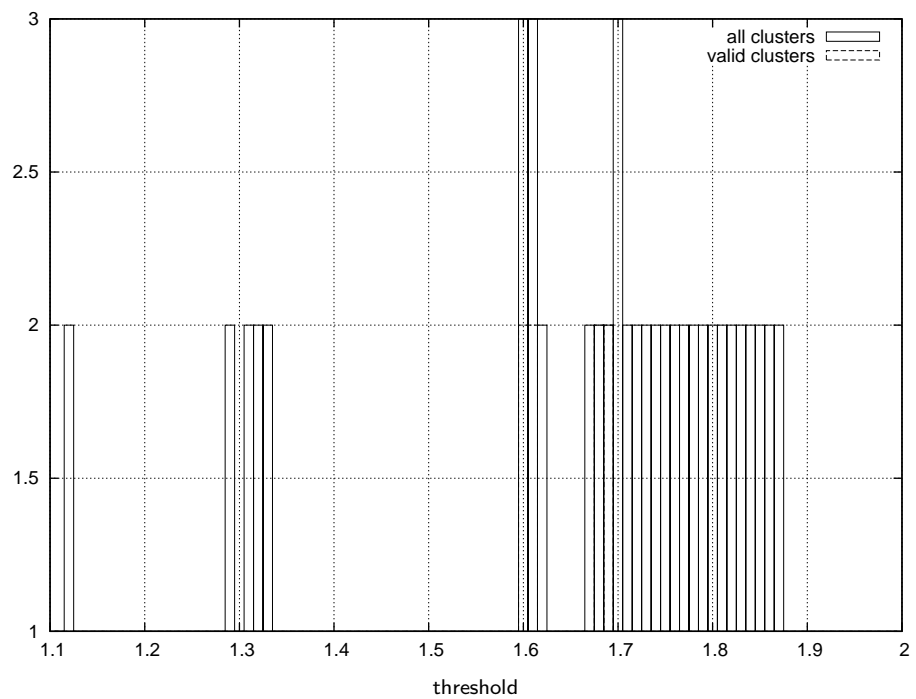
Figure D.8: Precision and recall for clustering with modified k-means algorithm and automatically located eyes using small data set of 55 faces without outliers; (a) precision and recall, (b) number of clusters.

D.1.2 Normalised similarity measure

Figures D.9 - D.16 present analysis of clustering results using a modified k-means algorithm with a normalised similarity measure. The graphs show values of average and maximum precision, average and maximum recall, numbers of all created clusters and numbers of valid clusters for different scenarios and values of thresholds.

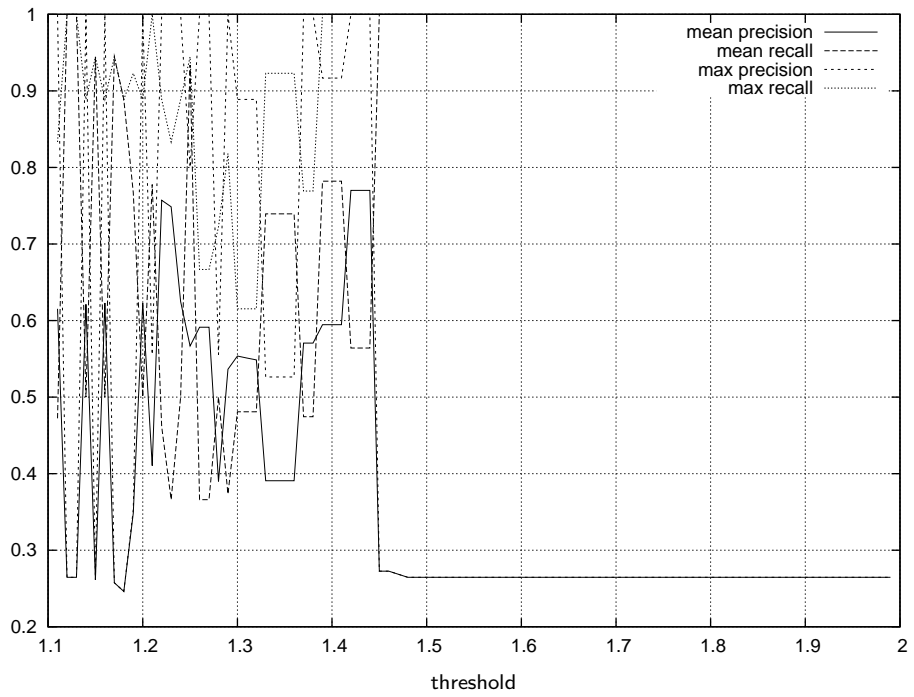


(a)

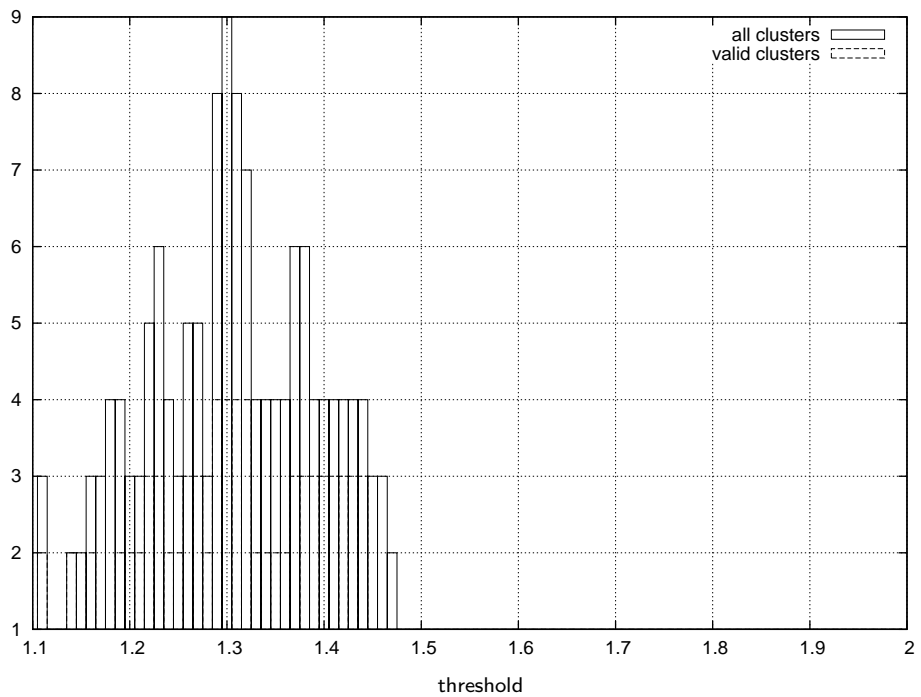


(b)

Figure D.9: Clustering with modified k-means algorithm and normalised distance using large data set with outliers and manual eye locations; (a) precision and recall, (b) number of clusters.

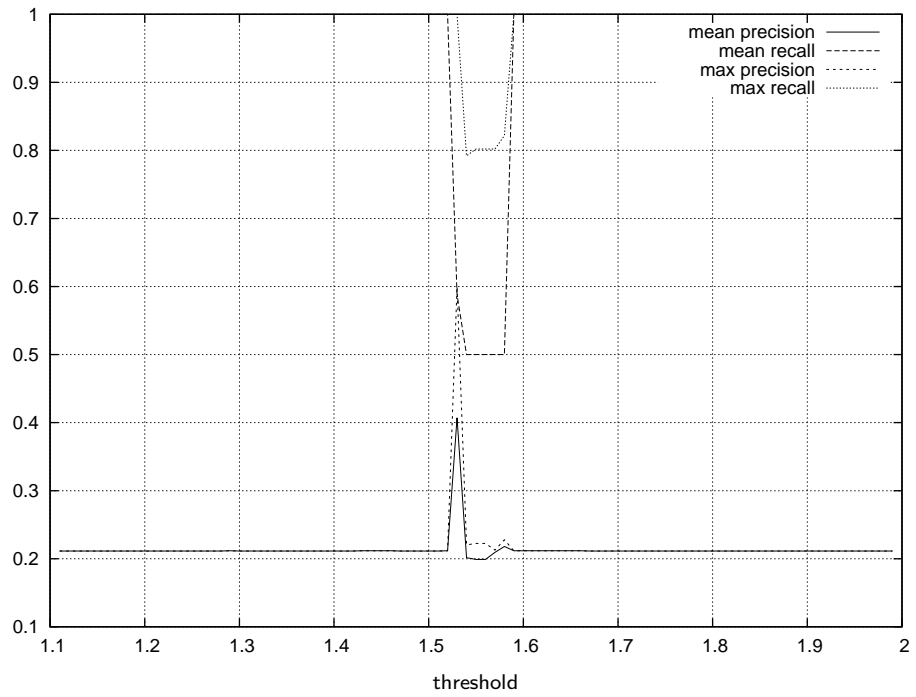


(a)

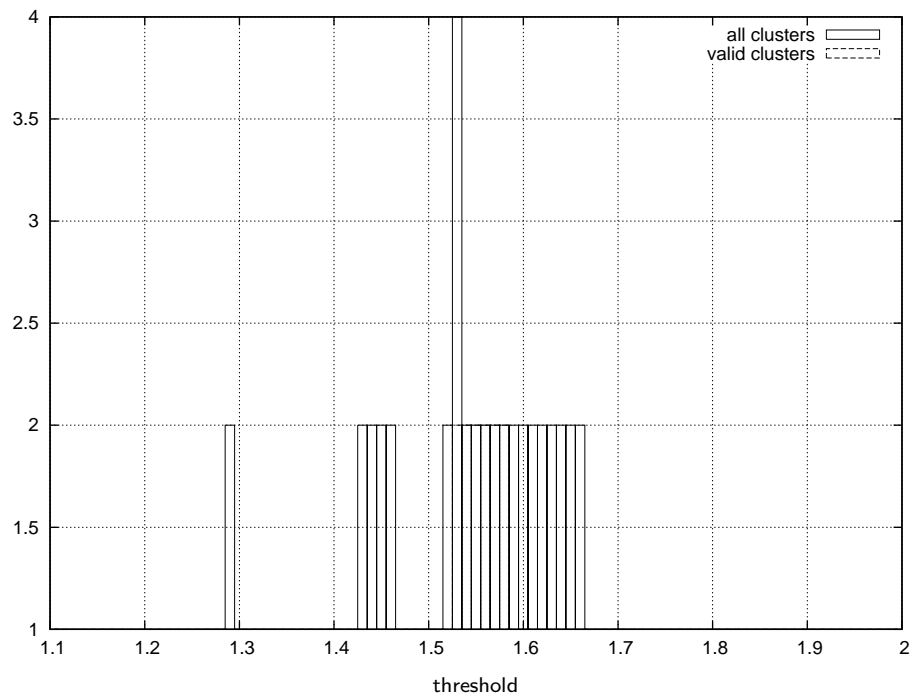


(b)

Figure D.10: Clustering with modified k-means algorithm and normalised distance using small data set with outliers and manual eye locations; (a) precision and recall, (b) number of clusters.

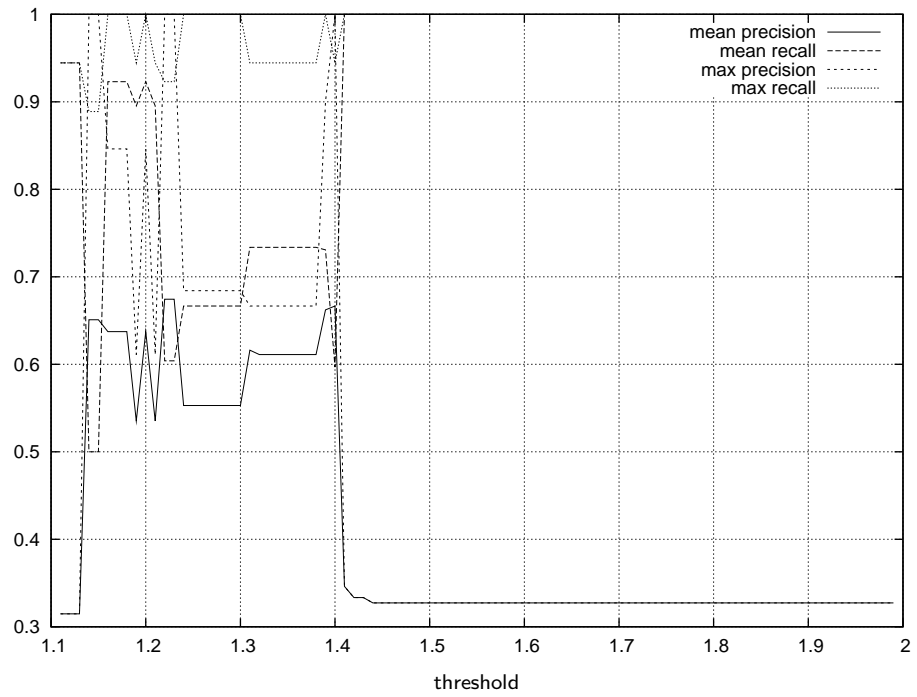


(a)

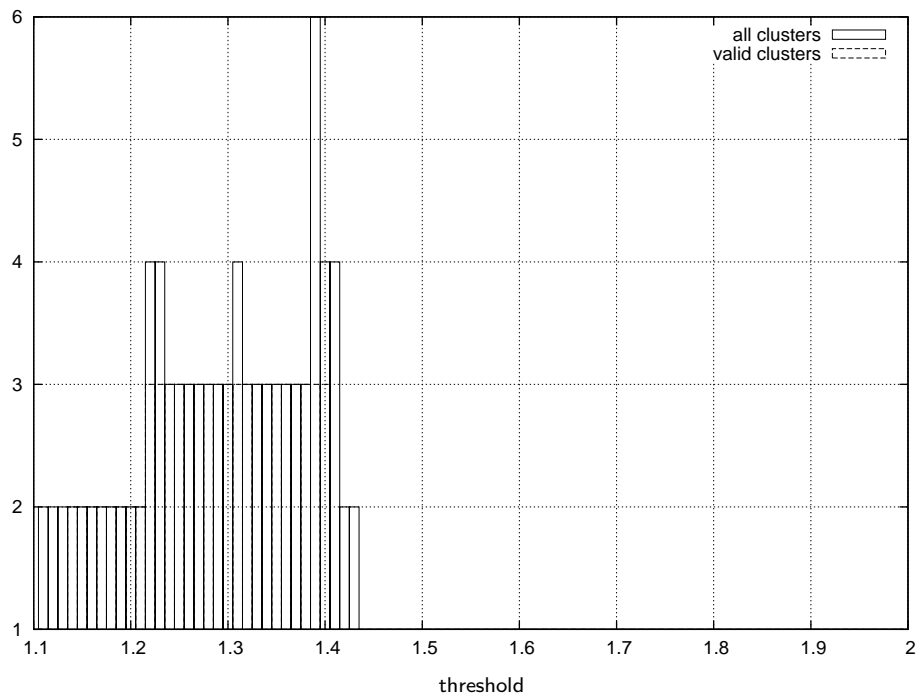


(b)

Figure D.11: Clustering with modified k-means algorithm and normalised distance using large data set without outliers and manual eye locations; (a) precision and recall, (b) number of clusters.

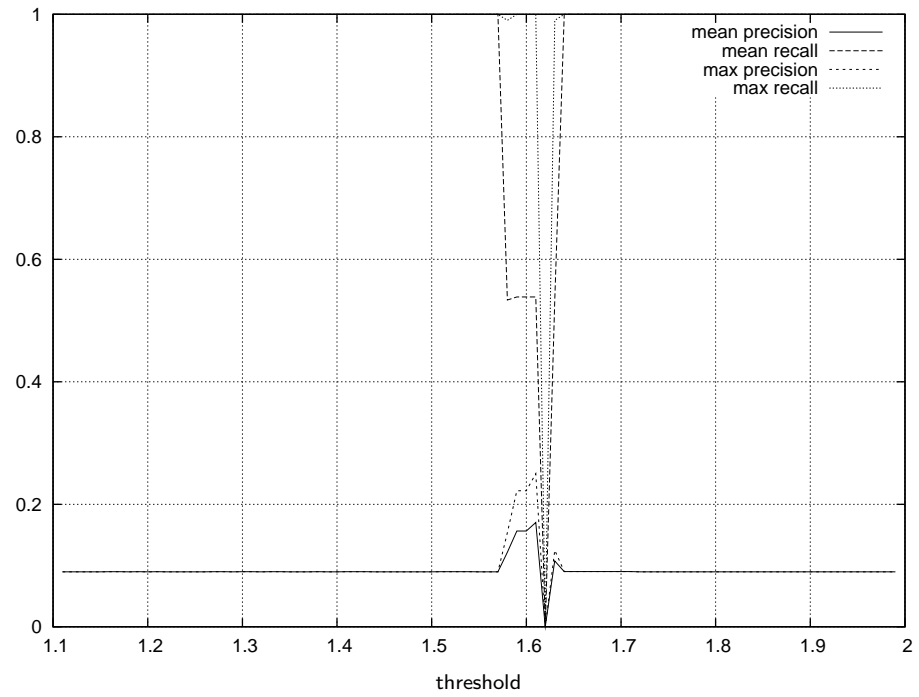


(a)

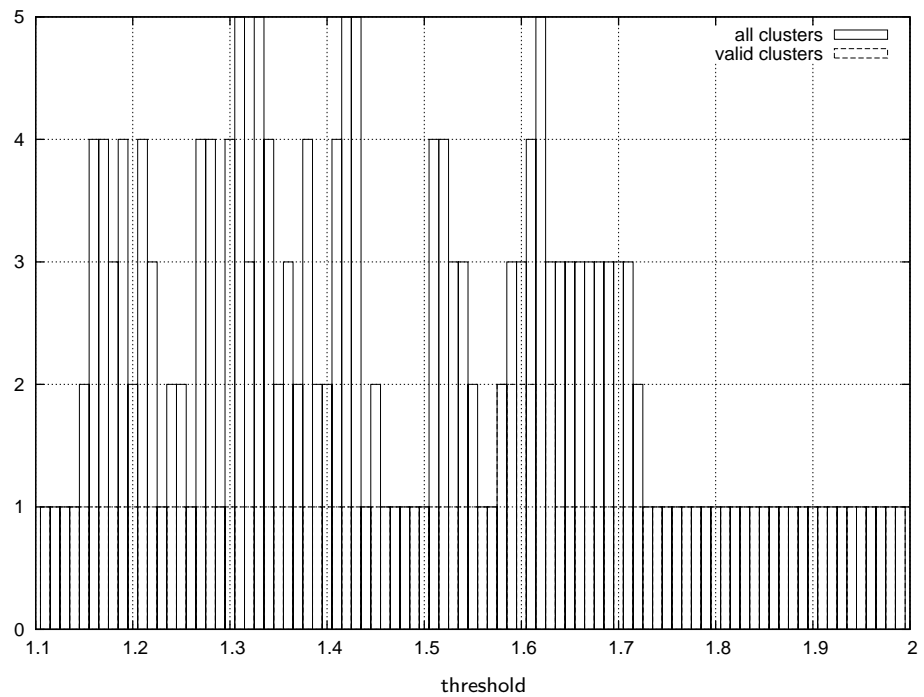


(b)

Figure D.12: Clustering with modified k-means algorithm and normalised distance using small data set without outliers and manual eye locations; (a) precision and recall, (b) number of clusters.

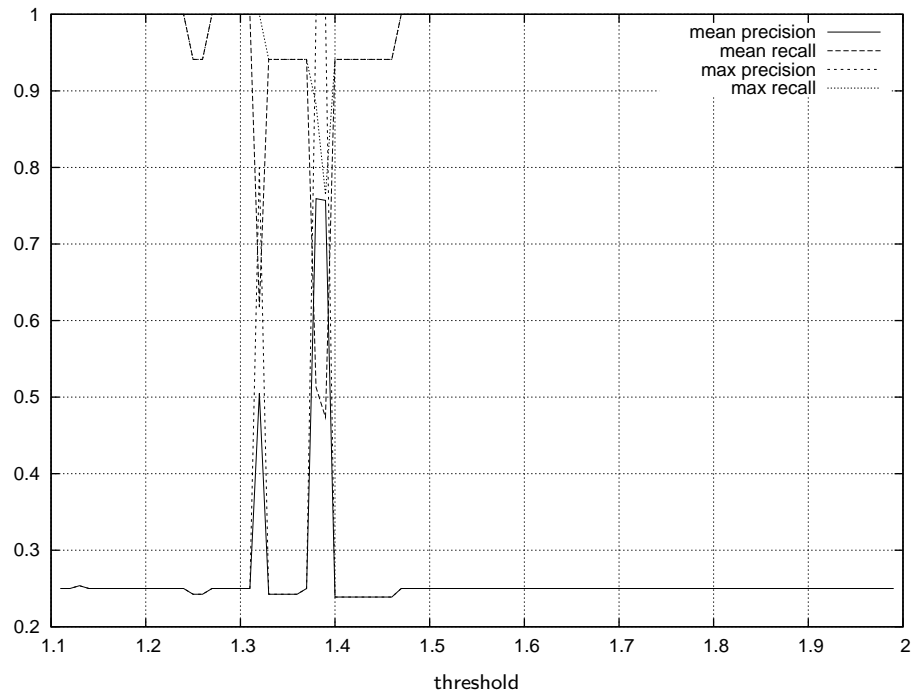


(a)

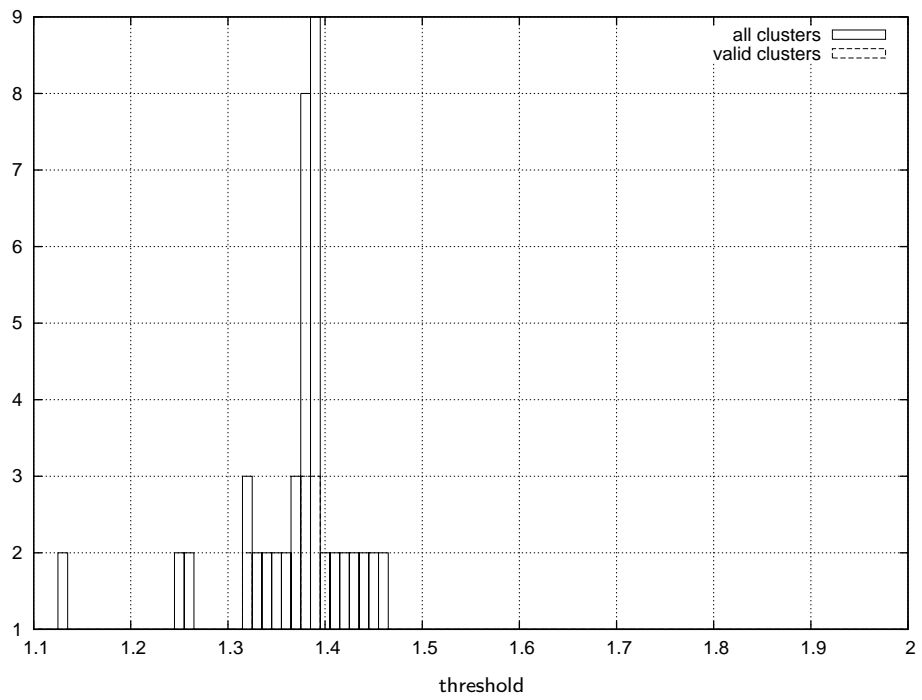


(b)

Figure D.13: Clustering with modified k-means algorithm and normalised distance using large data set with outliers and automated eye locations; (a) precision and recall, (b) number of clusters.

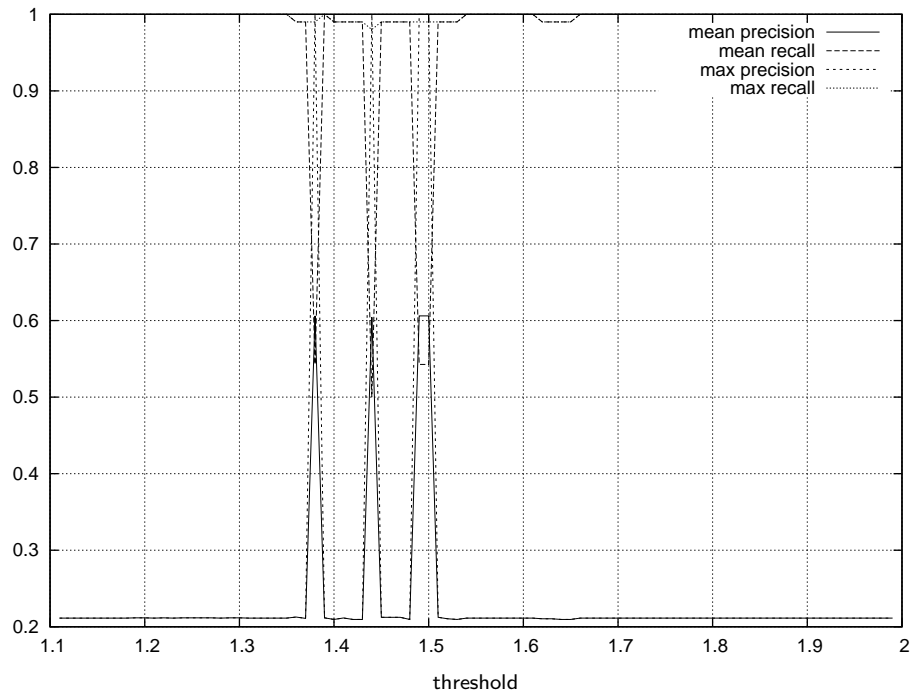


(a)

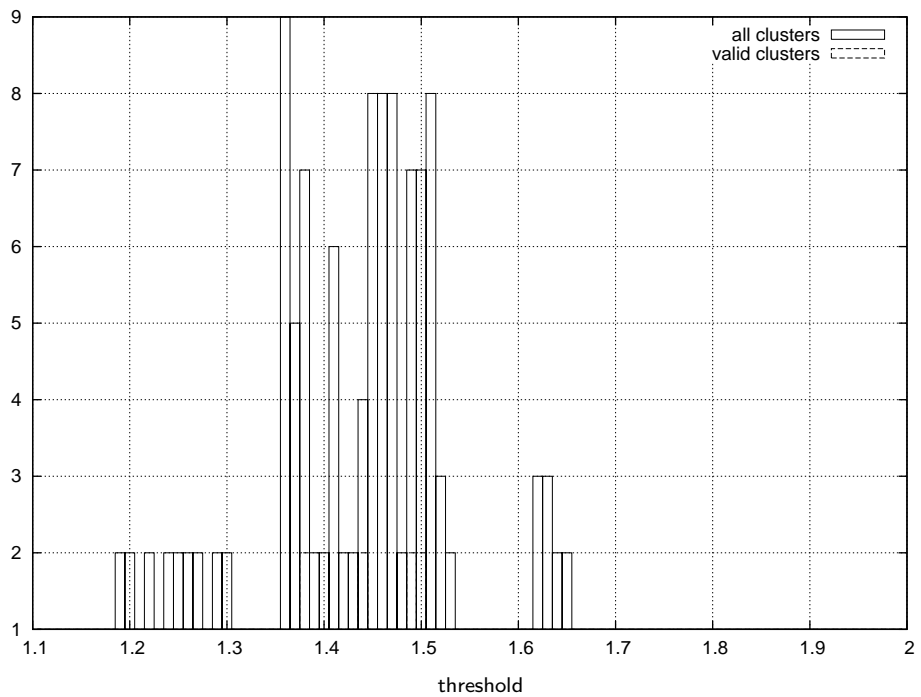


(b)

Figure D.14: Clustering with modified k-means algorithm and normalised distance using small data set with outliers and automated eye locations; (a) precision and recall, (b) number of clusters.

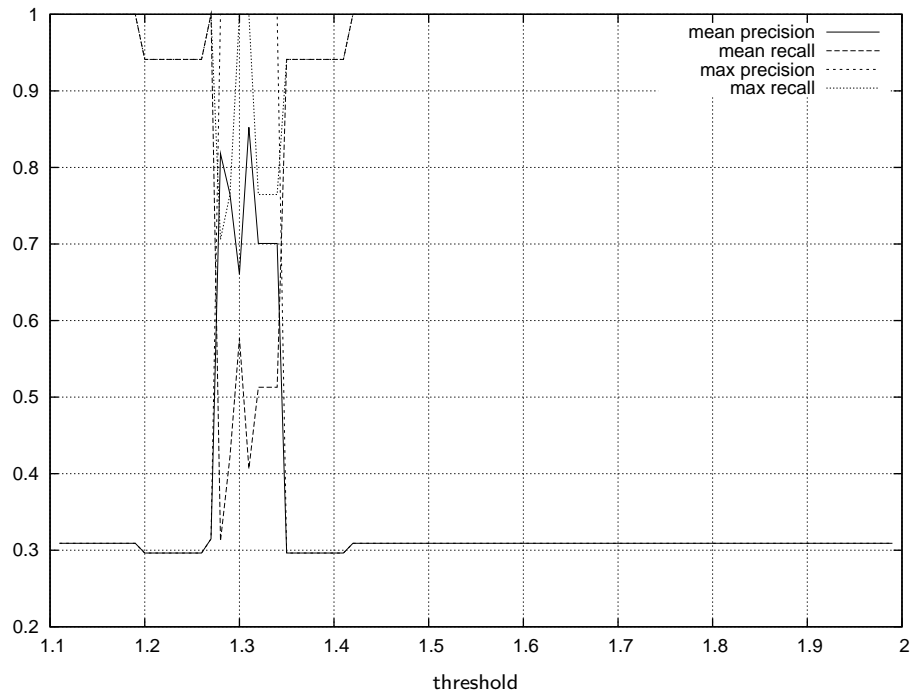


(a)

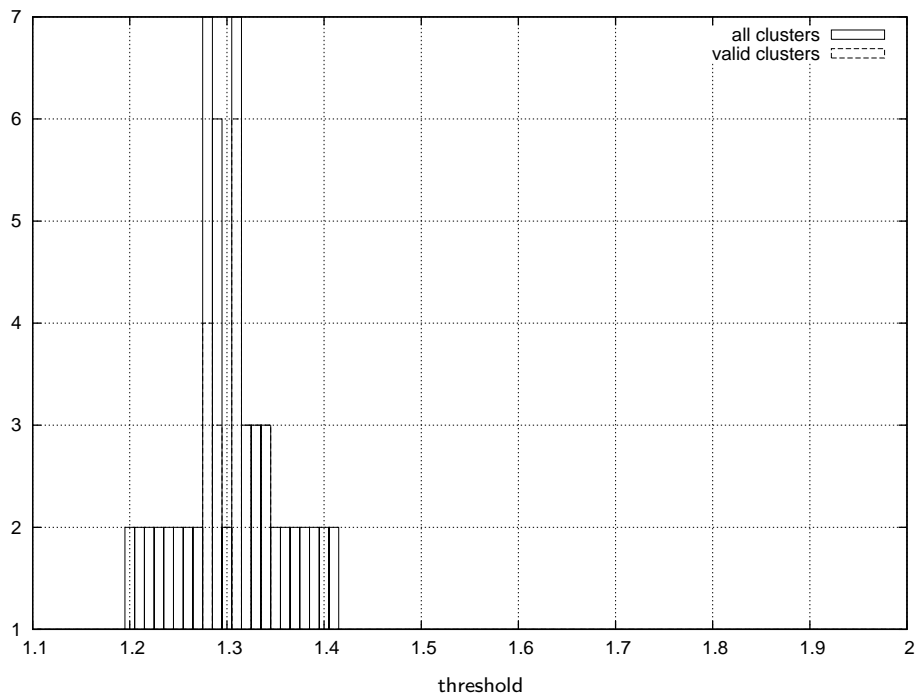


(b)

Figure D.15: Clustering with modified k-means algorithm and normalised distance using large data set without outliers and automated eye locations; (a) precision and recall, (b) number of clusters.



(a)

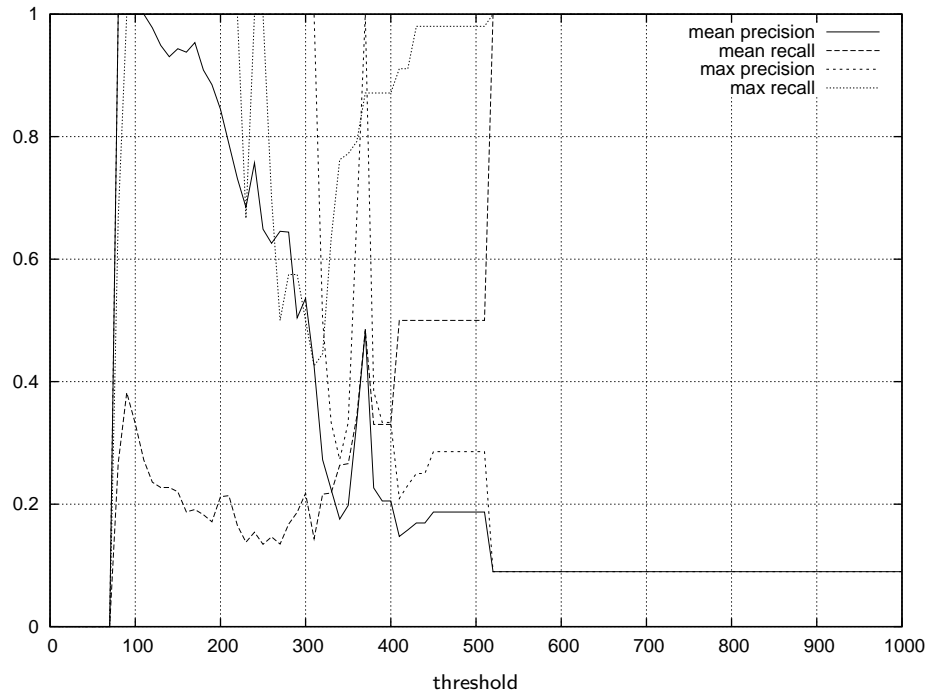


(b)

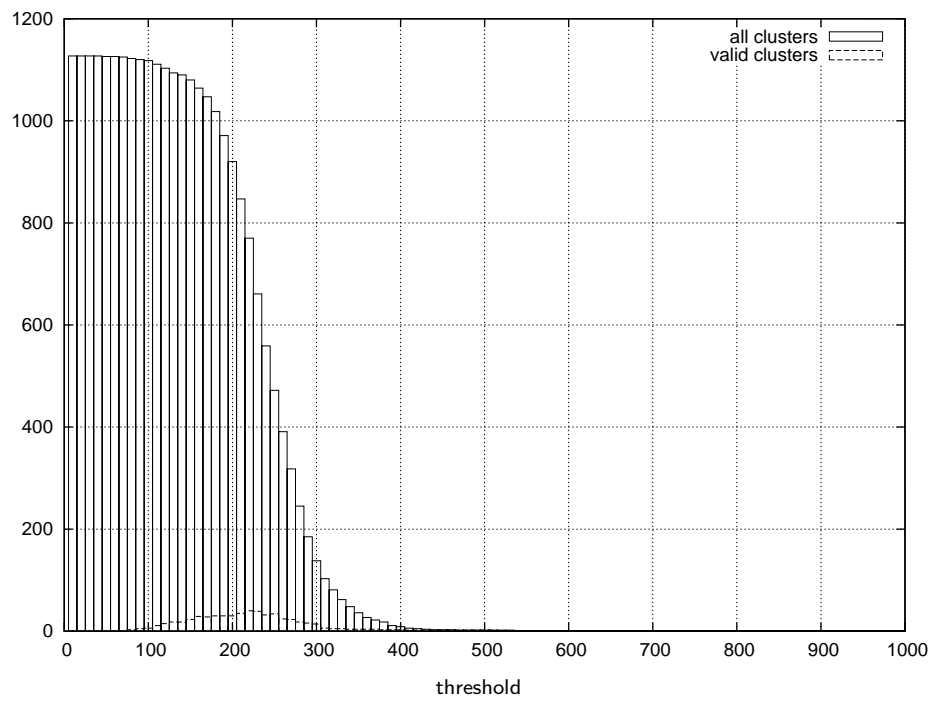
Figure D.16: Clustering with modified k-means algorithm and normalised distance using small data set without outliers and automated eye locations; (a) precision and recall, (b) number of clusters.

D.2 Single-link approach

Figures D.17 - D.24 present analysis of clustering results using a modified single-link algorithm. The graphs show values of average and maximum precision, average and maximum recall, numbers of all created clusters and numbers of valid clusters for different scenarios and values of thresholds.

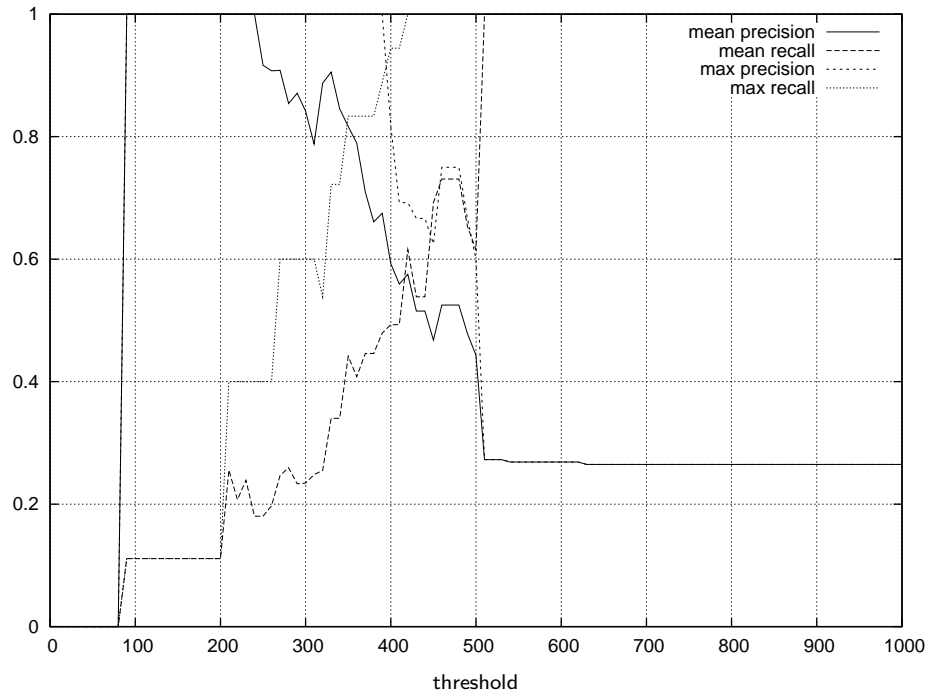


(a)

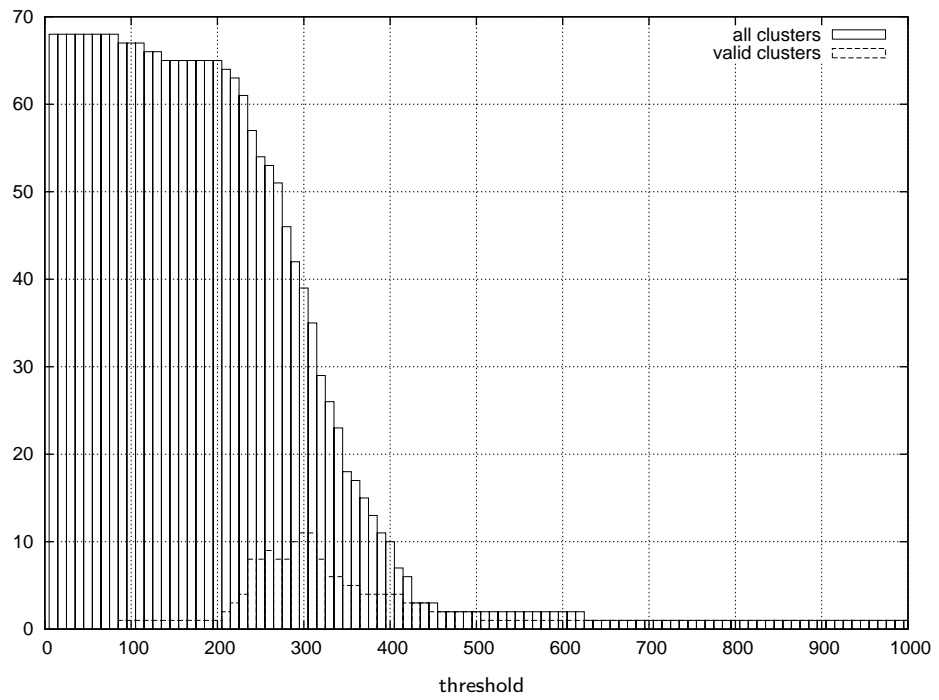


(b)

Figure D.17: Clustering with modified single-link algorithm using large data set containing outliers, manually located eyes; (a) precision and recall, (b) number of clusters.

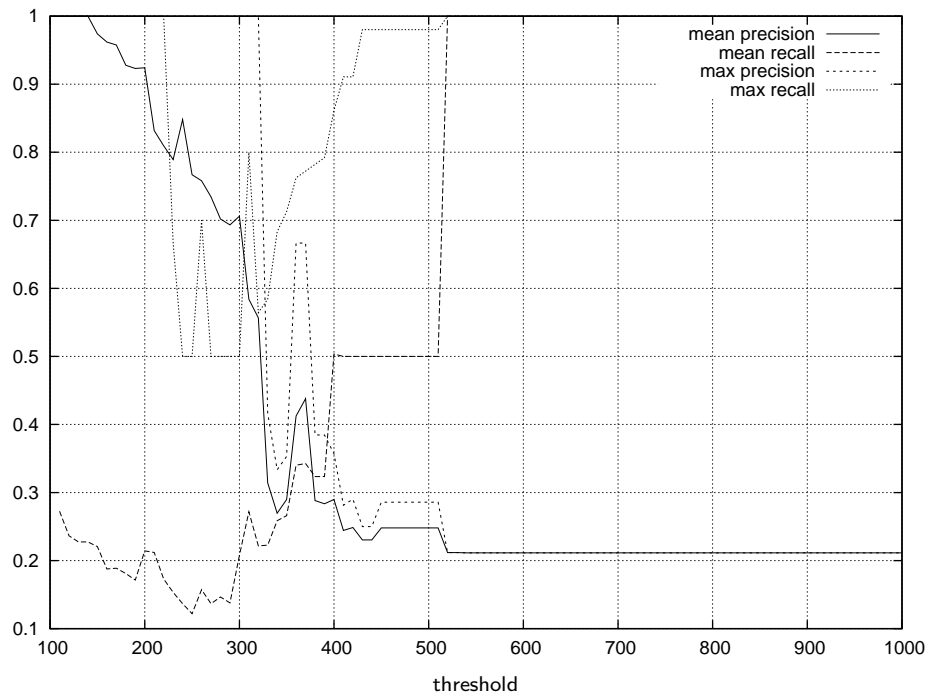


(a)

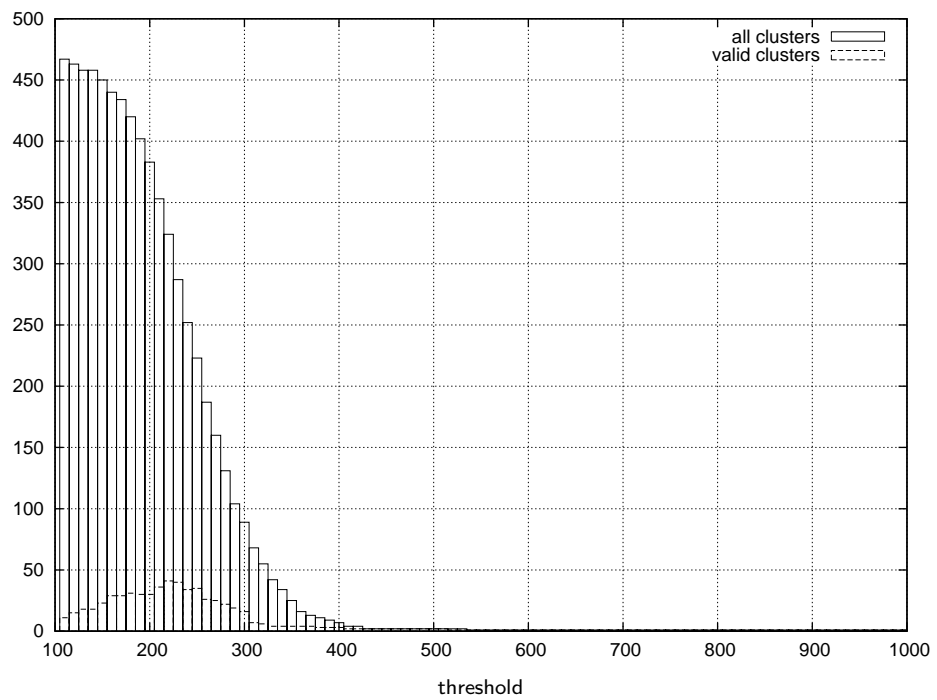


(b)

Figure D.18: Clustering with modified single-link algorithm using small data set containing outliers, manually located eyes; (a) precision and recall, (b) number of clusters.

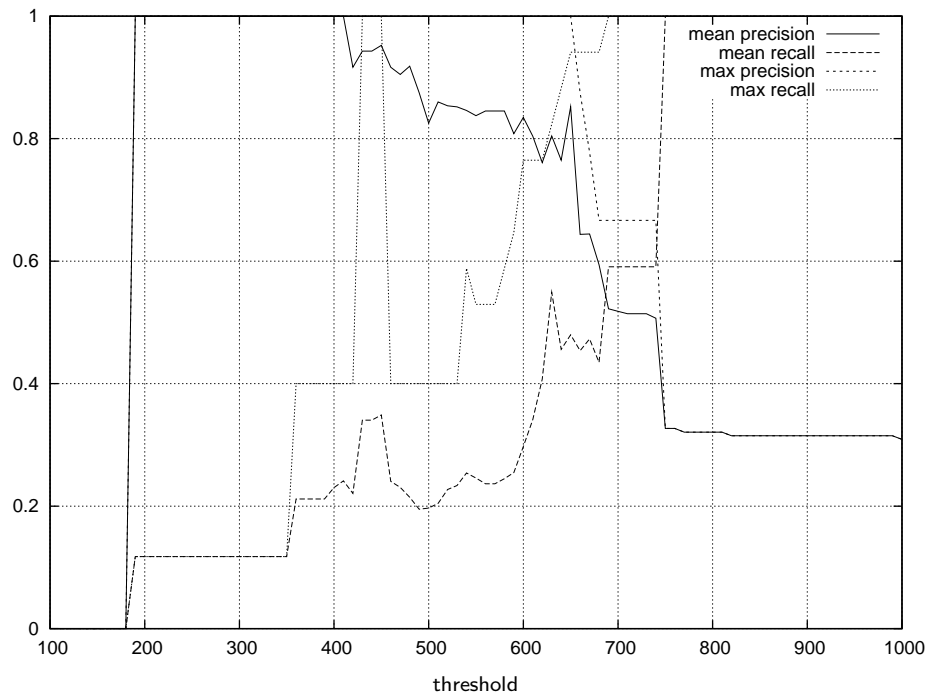


(a)

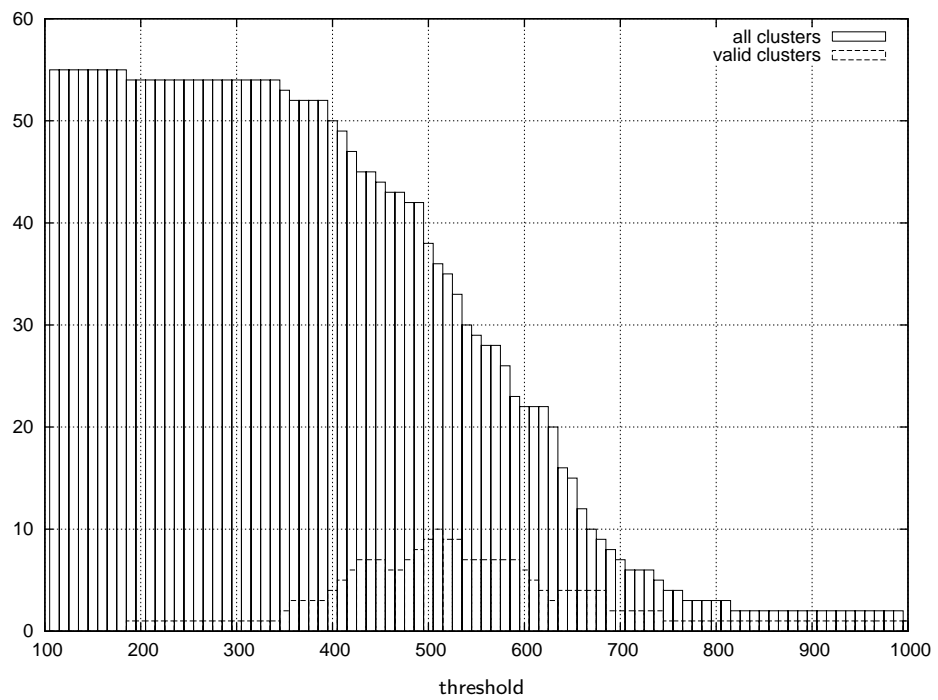


(b)

Figure D.19: Clustering with modified single-link algorithm using large data set without outliers with manually located eyes; (a) precision and recall, (b) number of clusters.

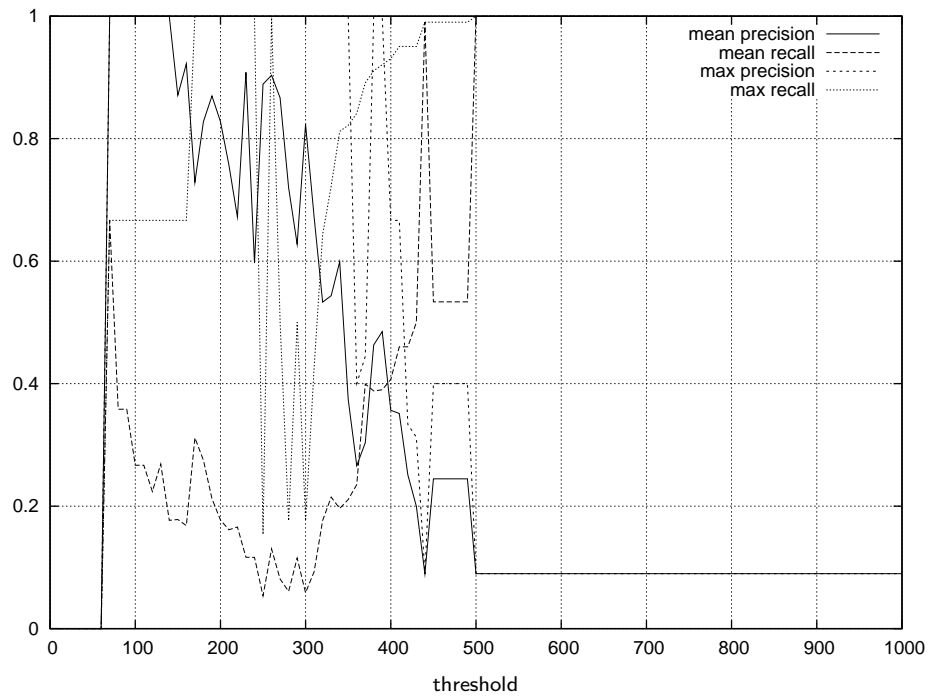


(a)

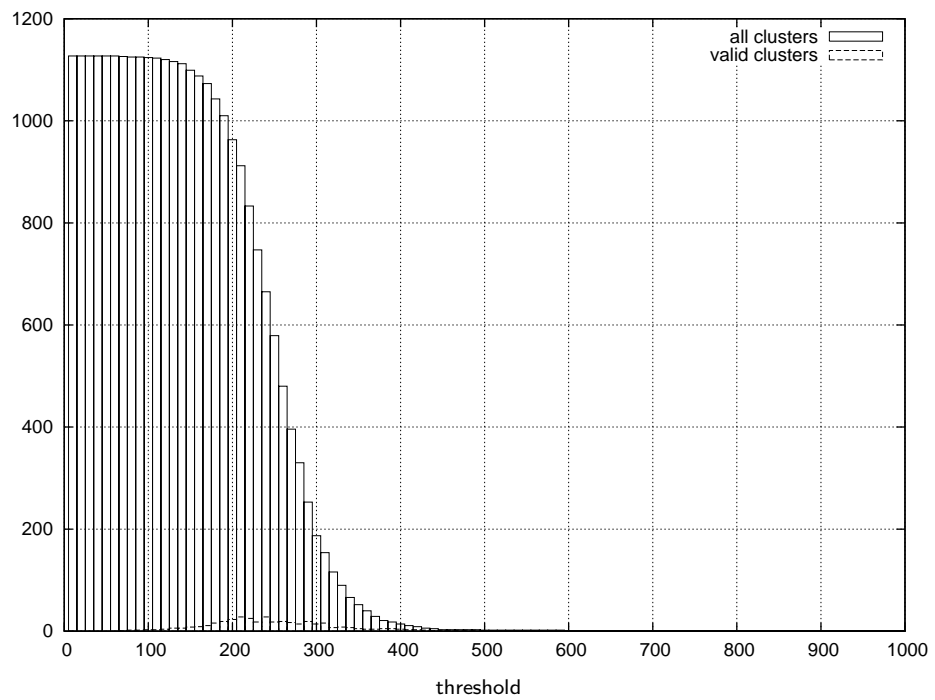


(b)

Figure D.20: Clustering with modified single-link algorithm using small data set without outliers with manually located eyes; (a) precision and recall, (b) number of clusters.

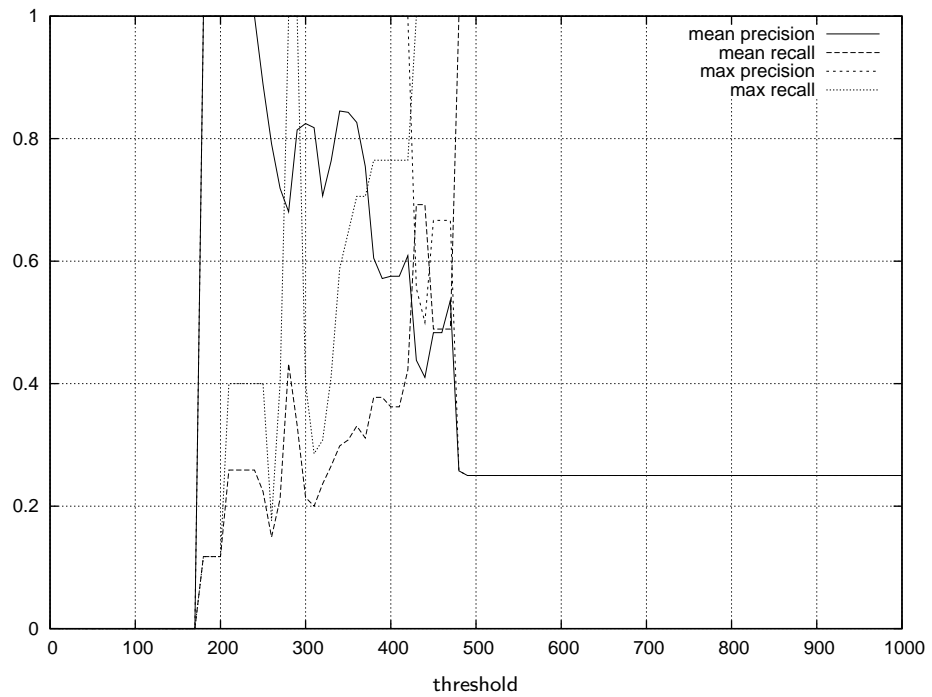


(a)

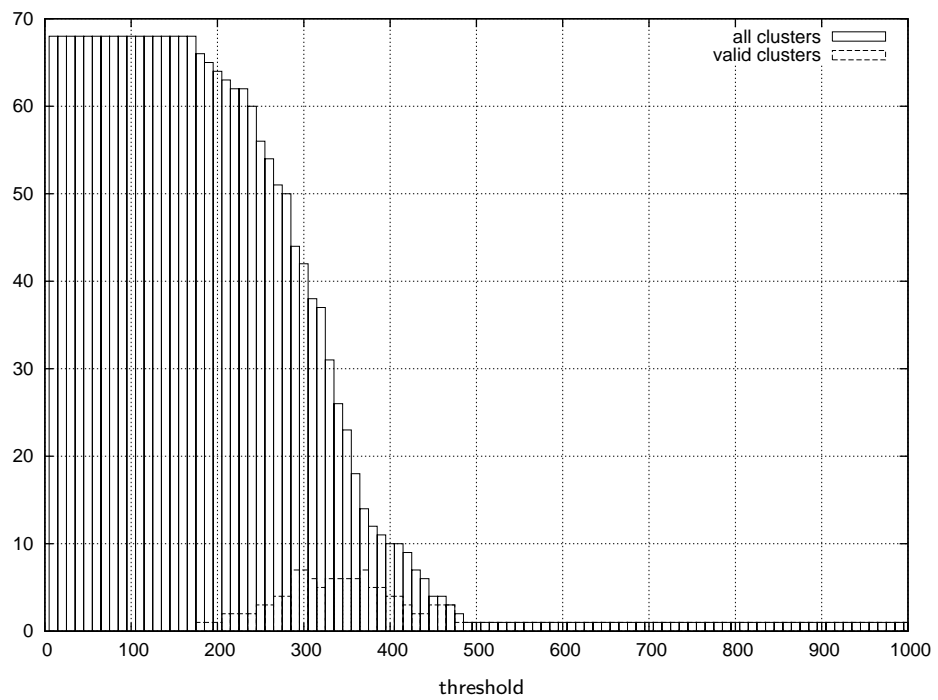


(b)

Figure D.21: Clustering with modified single-link algorithm using large data set containing outliers, automatically located eyes; (a) precision and recall, (b) number of clusters.

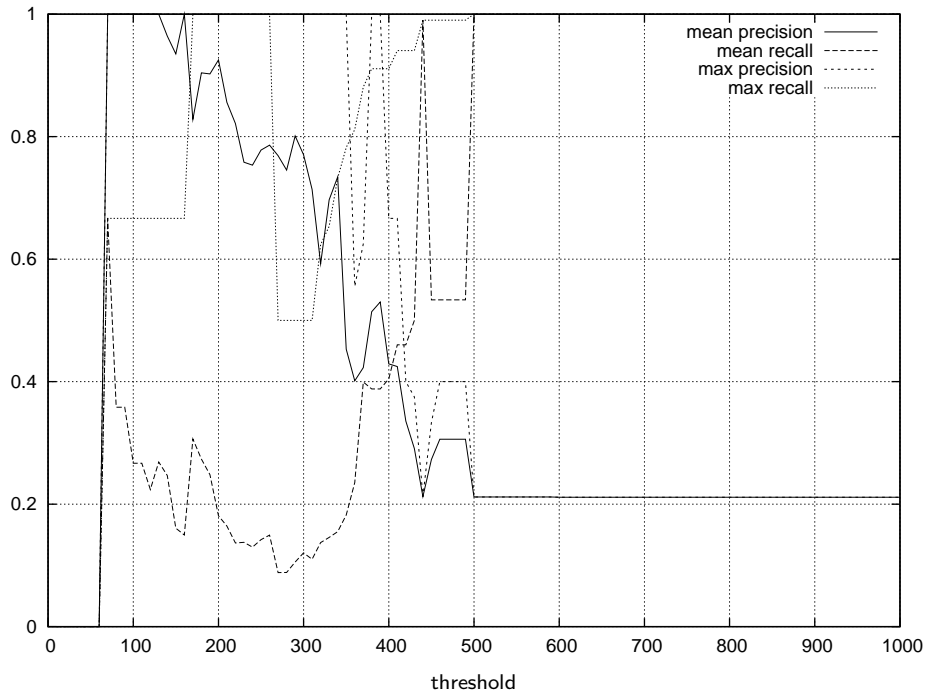


(a)

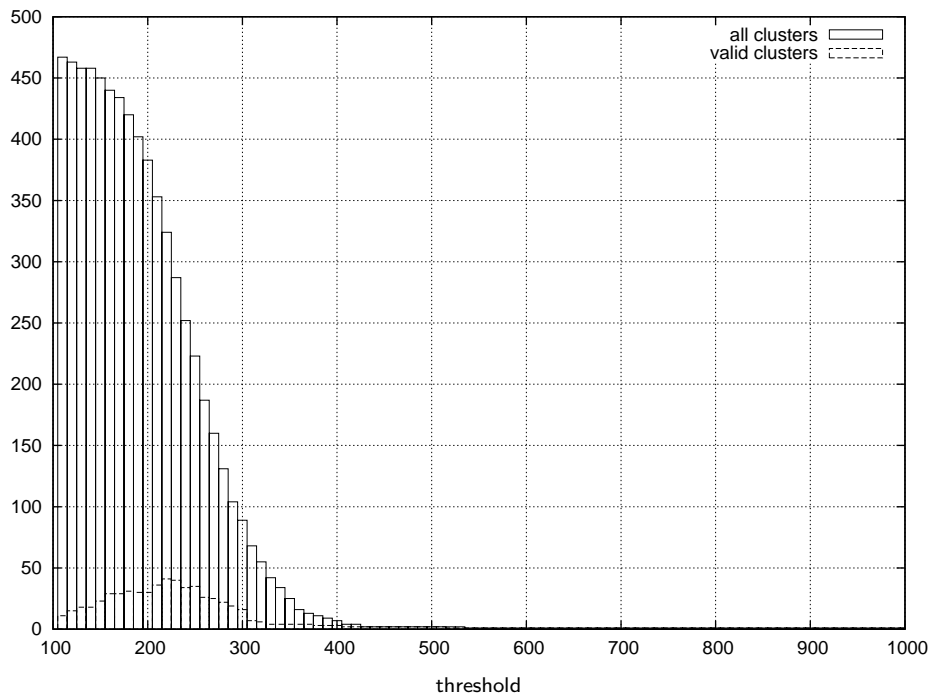


(b)

Figure D.22: Clustering with modified single-link algorithm using small data set containing outliers, automatically located eyes; (a) precision and recall, (b) number of clusters.

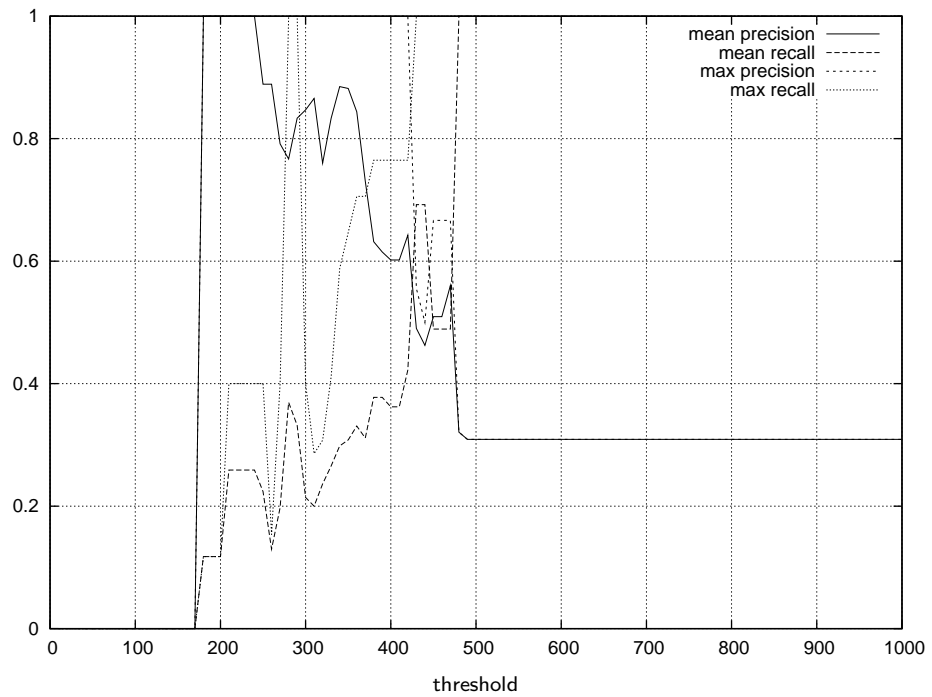


(a)

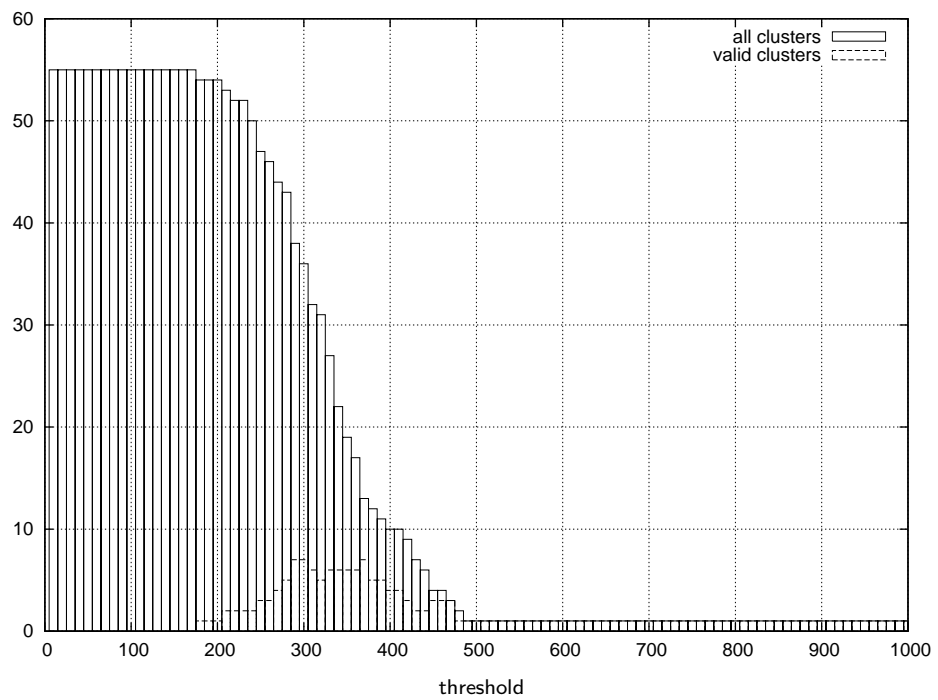


(b)

Figure D.23: Clustering with modified single-link algorithm using large data set without outliers, automatically located eyes; (a) precision and recall, (b) number of clusters.



(a)



(b)

Figure D.24: Clustering with modified single-link algorithm using small data set without outliers, automatically located eyes; (a) precision and recall, (b) number of clusters.