Helsinki University of Technology Laboratory of Computational Engineering Publications
Teknillisen korkeakoulun Laskennallisen tekniikan laboratorion julkaisuja
Espoo 2007 REPORT B68

# OBJECTS EXTRACTION AND RECOGNITION FOR CAMERA-BASED INTERACTION: HEURISTIC AND STATISTICAL APPROACHES

Hao Wang

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Helsinki University of Technology Laboratory of Computational Engineering Publications
Teknillisen korkeakoulun Laskennallisen tekniikan laboratorion julkaisuja
Espoo 2007 REPORT B68

# OBJECTS EXTRACTION AND RECOGNITION FOR CAMERA-BASED INTERACTION: HEURISTIC AND STATISTICAL APPROACHES

Hao Wang

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Electrical and Communications Engineering, Helsinki University of Technology, for public examination and debate in Auditorium D at Helsinki University of Technology (Espoo, Finland) on the 14th of December, 2007, at 12 noon.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Laboratory of Computational Engineering

Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Laskennallisen tekniikan laboratorio

Distribution:
Helsinki University of Technology
Laboratory of Computational Engineering
P.O.Box 9203
FI-02015 TKK
FINLAND

Tel.   +358 9 451 5726
Fax   +358 9 451 4833
http://www.lce.hut.fi

Online in PDF format: http://lib.tkk.fi/Diss/2007/isbn9789512291342/

E-mail: hao.ui.wang@nokia.com

ISBN 978-951-22-9137-3 (printed)
ISBN 978-951-22-9134-2 (PDF)
ISSN 1455-0474
Picaset Oy
Helsinki 2007

| | |
|---|---|
| ABSTRACT OF DOCTORAL DISSERTATION | HELSINKI UNIVERSITY OF TECHNOLOGY<br>P.O. BOX 1000, FI-02015 TKK<br>http://www.tkk.fi |

Author Hao Wang

Name of the dissertation

OBJECTS EXTRACTION AND RECOGNITION FOR CAMERA-BASED INTERACTION: HEURISTIC AND STATISTICAL APPROACHES

| | |
|---|---|
| Manuscript submitted 7. 9.2007 | Manuscript revised 28.11.2007 |

Date of the defence 14.12.2007

| | |
|---|---|
| ☐ Monograph | ☒ Article dissertation (summary + original articles) |

| | |
|---|---|
| Department | Department of Electrical and Communications Engineering |
| Laboratory | Laboratory of Computational Engineering |
| Field of research | Computational Engineering |
| Opponent(s) | Prof. Matti Pietikäinen |
| Supervisor | Prof. Jouko Lampinen |
| Instructor | |

Abstract

In this thesis, heuristic and probabilistic methods are applied to a number of problems for camera-based interactions. The goal is to provide solutions for a vision based system that is able to extract and analyze interested objects in camera images and to use that information for various interactions for mobile usage. New methods and new attempts of combination of existing methods are developed for different applications, including text extraction from complex scene images, bar code reading performed by camera phones, and face/facial feature detection and facial expression manipulation.

The application-driven problems of camera-based interaction can not be modeled by a uniform and straightforward model that has very strong simplifications of reality. The solutions we learned to be efficient were to apply heuristic but easy of implementation approaches at first to reduce the complexity of the problems and search for possible means, then use developed statistical learning approaches to deal with the remaining difficult but well-defined problems and get much better accuracy. The process can be evolved in some or all of the stages, and the combination of the approaches is problem-dependent.

Contribution of this thesis resides in two aspects: firstly, new features and approaches are proposed either as heuristics or statistical means for concrete applications; secondly engineering design combining seveal methods for system optimization is studied. Geometrical characteristics and the alignment of text, texture features of bar codes, and structures of faces can all be extracted as heuristics for object extraction and further recognition. The boosting algorithm is one of the proper choices to perform probabilistic learning and to achieve desired accuracy. New feature selection techniques are proposed for constructing the weak learner and applying the boosting output in concrete applications. Subspace methods such as manifold learning algorithms are introduced and tailored for facial expression analysis and synthesis. A modified generalized learning vector quantization method is proposed to deal with the blurring of bar code images. Efficient implementations that combine the approaches in a rational joint point are presented and the results are illustrated.

Keywords camera-based interaction, text extraction, bar code, facial expression, boosting, manifold learning

| | |
|---|---|
| ISBN (printed) 978-951-22-9137-3 | ISSN (printed) |
| ISBN (pdf) 978-951-22-9134-2 | ISSN (pdf) 1455-0474 |
| Language English | Number of pages 68p. + app. 72p. |

Publisher Laboratory of Computational Engineering

Print distribution Laboratory of Computational Engineering

☒ The dissertation can be read at http://lib.tkk.fi/Diss/

# Preface

The work of this thesis for the degree of Doctor of Technology has been carried out in the Visual Interaction System research team of Nokia Research Center (NRC), Beijing, China between the years 2001-2007. During these years, the Laboratory of Computational Engineering at Helsinki University of Technology provided me with careful instructions and supervision.

I would like to first express my sincere gratitude to Prof. Jouko Lampinen for excellent guidance and support, and to Dr. Jari Kangas who introduced the opportunity and encouraged me all the way for my postgraduated study.

Sincere thanks are also due to my line manger in NRC, Dr. Wang Kongqiao, for his kindly support and advice. I am grateful to my colleagues and management team members of my department and projects, Zou Yanming, Wang Xia, Ma Jian, Niu Xin, Seppo Pienimaa, Roope Takala, Ramalingam Hariharan, Seppo Hamalainen, Juha Iso-Sipila, Jari Alhonen, who have helped me by providing good suggestions and given me grate supports with time, process and constructive solutions.

My deepest gratitude goes to my wife and my parents who had steadfastly support my studies during all the time in spite of difficult circumstances.

*Hao Wang*

# List of Publications

This thesis includes the following seven publications.

[P1] Hao Wang, Jari Kangas, Text location in color scene images for information acquisition by mobile terminals, *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, Vol. 6, pp. 436-441, Orlando, Florida, 2001. IIIS.

[P2] Hao Wang, Jari Kangas, Character-like region verification for extracting text in scene images, *Proceedings of 6th International Conference on Document Analysis and Recognition*, pp. 957-962, Seattle, WA, 2001. IEEE.

[P3] Kongqiao Wang, Yanming Zou, Hao Wang, 1D bar code reading on camera phones, *International Journal of Image and Graphics*, vol. 7, no. 3, pp. 529-550, 2007. World Scientific Publishing, ISSN: 0219-4678.

[P4] Hao Wang, Yanming Zou, 2D Bar Codes Reading: Solutions for Camera Phones, *International Journal of Signal Processing*, Vol. 3, No. 3, pp. 164-170, 2006. World Academy of Science, Engineering and Technology, ISSN: 1304-4478.

[P5] Hao Wang, Kongqiao Wang, Facial feature extraction and image-based face drawing, *Proceedings of 6th International Conference on Signal Processing*, Vol. 1, pp. 699-702, Beijing, China, 2002. IEEE.

[P6] Hao Wang, Image-based face drawing using active shape models and parametric morphing, *Proceedings of International Conference on Neural Networks and Signal Processing*, Vol. 2, pp. 1017-1020, Nanjing, China, 2003. IEEE.

[P7] Hao Wang, Konqiao Wang, Affective Interaction Based on Person-Independent Facial Expression Space, *accepted by Neurocomputing, Special Issue for Vision Research*. Elsevier, ISSN: 0925-2312.

# Abbreviations

| | |
|---|---|
| AAM | Active Appearance Model |
| ASM | Active Shape Model |
| AUs | Action Units |
| BAG | Block Adjacency Graph |
| EM | Expectation-Maximization |
| FACS | Facial Action Coding System |
| FAR | False Acceptance Rate |
| FRR | False Rejection Rate |
| GLVQ | Generalized Learning Vector Quantization |
| HCI | Human-Computer Interaction |
| ICA | Independent Component Analysis |
| LBP | Local Binary Pattern |
| LDA | Linear Discriminant Analysis |
| LLE | Locally Linear Embedding |
| LPP | Locality Preserving Projections |
| MLP | Multi-Layer Perceptron |
| MRASM | Multi-Resolution Active Shape Model |
| OCR | Optical Character Recognition |
| PCA | Primary Component Analysis |
| PDM | Point Distribution Model |
| RBF | Radial Basis Function |
| ROI | Region of Interest |
| SLLE | Supervised Locally Linear Embedding |
| SLPP | Supervised Locality Preserving Projections |
| SNR | Signal-to-Noise Ratio |
| SVM | Support Vector Machine |
| UI | User Interface |
| ZC2 | Zero Crossings of the second derivatives |

# Contents

# Chapter 1

# Introduction

As mobile hardware configurations evolve, camera is becoming more and more important interaction modality for information collection from the environment in many ways. Some of the proposed methods, such as optical character recognition (OCR) and bar code reading from images captured by a mobile camera, readily produce textual information. And it is also possible to use the camera in many other user interface (UI) solutions to realize natural interactions from it, for example, human faces and facial expressions can be used as identification approaches and emotion activators for human-computer interaction (HCI); and hand gestures can be used for UI navigation especially in some specific scenarios such as entertainment and gaming.

The mobile visual interaction systems are remarkably driven by concrete applications and the capability of imaging devices. Multidisciplinary enabling techniques are required to deal with the problems from many use cases, and no single paradigm is available for all the solutions. Although it is a broad umbrella under which fall diverse aspects of optics, electronics, photography, computer technology, and usability, there is possibility to identify some common questions and a generalized framework underline to be easily extended for new problems.

The first preliminary question focuses on the objects we are interested in: what kind of information contained in a natural image is the most important and valuable. There are infinite answers considering different scenarios, however, textual information and occurrence of human beings including human activities will have high priorities that attract people's attention in common sense. Revelation of this point helps to define the objectives of our camera-based interaction solutions.

After getting agreement on the objects of interests, the next question is what we should do with such objects for visual interactions. By giving the pixel arrays of any input image, one of the basic tasks is to detect and extract the interested objects from their complex background, though it seems to be very easy for human observation. Pixel grouping for high-level image understanding by unsupervised learning methods has attracted considerable attention in recent decades (Yu and Shi, 2003; Zoller et al., 2002; Prasad and Sowmya, 2004). By adding the priors of the objects to be detected, it would be more efficient to convert the clustering problem to a "black and white" classification problem in a supervised manner, which benefits from the definition of the objects of interests. Then the consequent operations of the objects extracted from the image are based on practical requirements, but recognition of the objects into some pre-defined categories is usually a must step. For example, in bar code reading application it is demanded to recognize the patterns of basic units after a bar code is located and segmented, in order to convert the camera-readable code into textual information. The algorithms adopted for object recognition are highly application-oriented.

There had been a lot of research works in the fields of text detection from images, bar code reading, and facial feature analysis. However, there are still quite many problems to be solved in order to implement the camera-based interaction as mentioned above. For text detection from images, existing methods have limitations with specific application domains. Even though a large number of algorithms have been proposed in the literature, no single method can provide satisfactory performance in all the applications due to the large variations in character font, size, texture, color, etc. For bar code reading, algorithms were mainly designed for the traditional laser scanners and there were few attempts to implement a system based on a low quality camera equipped mobile phone. Facial feature analysis had attracted intensive attentions in the past decades. Various algorithms had been proposed for face detection, facial feature extraction and facial expression analysis. Our intention is to develop practical solutions for mobile interactions that utilize the advances of facial image processing.

As the motivation of designing such a visual interaction system is mainly for mobile usage, there are some technical challenges caused by the limitation of hand held equipments. The biggest problem comes from the image acquisition stage where user variability tends to degrade the robustness of the system. Most users are not well-trained photo takers, and in many cases it is very difficult to obtain an ideal image for object extraction and recognition, the proposed solutions must be capable to deal with complex imaging environment, e.g., noise, blurring, illumination variance, shadows, extreme exposure, pose and viewpoint variations. The computation efficiency is another concerned problem. Real time operation of visual interaction systems requires a quick response from the system to a visual input, however, the computation power of mobile devices is limited and the memory is relatively small. Simple but effective on-line implementation of the visual interaction system is certainly desirable.

Then basic components of the camera-based interaction system have been identified: the focus is concentrated on objects of textual information and human being related facts contained in natural images; the task is to extract the interested objects from their complex background and probably classify them into pre-defined categories; and the context is that variability of image acquisition and computation efficiency must be addressed. Consequently, the following question is how to achieve these objectives under contextual assumptions. Feature selection for increasing the robustness of visual object detection and recognition against complex environmental interferences is an open question, and in many applications empirical knowledge of human observation can be applied to provide acceptable performance. Recently developed statistical learning methods will be used to further select most salient features in an adaptive form, and apply them in classification stage.

In another word, a hybrid system that combines heuristic and statistical methods is proved to be effective to solve problems bursting from real applications. However, there is no guideline indicating how to find an optimal joint-point of the empirical and theoretical approaches. The art of combination is again an application-driven and empirical problem.

In this thesis, we will look into applying a hybrid method to visual object extraction and recognition as defined above by case study, where combination of heuristic and statistical approaches is employed and rational joint-point is attempted. In the following chapters, we define objectives in detailed forms and corresponding solutions are manifested, which takes the combined approaches as a preferred course.

In the second chapter the general concept of the combination of heuristic and statistical methods for visual object extraction and recognition is introduced. First we discuss the structural information of the visual objects to be processed and try to find some common features as the source of the original heuristic approach. Afterwards possibilities of using statistical training method to select optimal features and approximate the final goals are presented. Background of the concrete applications is briefly specified.

In Chapter 3, text extraction from natural scene images is discussed, where heuristics of the identified features are reviewed and parameterization is performed by empirical training. Main difficulties of the problem come from the nature of the text or characters of different scripts that there lacks of obvious discriminance to distinguish the characters from complex background if high level knowledge of the scripts is unavailable.

Chapter 4 looks at a well defined scenario of bar code reading by using common phone cameras. Because bar codes, either 1D or 2D, contain dominant textural information that is relatively easy to be identified from the background image, direct heuristics can be applied to detect bar codes with acceptable accuracy. However, the performance of bar code character recognition is always degraded due to the blurring caused by out-of-focus problem, which is inevitable in a low quality camera-based picturing system. Statistical model is used to describe the problem in a more generalized form, and deblurring problem is discussed and implemented to improve the robustness of the system.

Chapter 5 focuses on the development of face manipulation system that can detect human faces from complex images, analyze facial expressions, and interact with human faces by expressional transformation and other facial feature manipulation. Recognizing a face has been crucial for our survival for millions of years, whereas it is not an easy task for machines because of the different interpretations used by human brain and machine perception. Subspace techniques are applied to deal with such kind of high-dimensional problems, also many heuristics are used for manipulation to achieve good performance.

A summary of publications and the author’s contribution to these publications are outlined in Chapter 6. Finally, the conclusions are drawn in Chapter 7.

# Chapter 2

# Combination of Heuristic and Statistical Approaches

People learn by experience rather than by theoretical approaches. Thus in a world which we do not understand, we develop relatively simple mental models so that we can make decision. Real life problems are known to be messy, dynamic and multi-objective, and involve high levels of uncertainty and constraints. Because traditional well defined models are no longer capable of handling this level of complexity, heuristic methods have attracted increasing attention in recent years for solving such problems. Inspired by nature, biology, statistical mechanics, physics and neuroscience, heuristic techniques are used to solve many problems where traditional methods have failed.

However, a heuristic is an algorithm that usually finds pretty good solutions, but there is no proof the solutions could not get arbitrarily bad; or it usually runs reasonably quickly, but there is no argument that this will always be the case. Often, one can find specially crafted problem instances where the heuristic will in fact produce very bad results or run very slowly; however, these instances might never occur in practice because of their special structure. Therefore, the use of heuristics is very common in real world implementations. For many practical problems, a heuristic algorithm may be the only way to get good solutions in a reasonable amount of time.

The question is what we could do on the basis of the heuristics to guarantee the performance of a solution, which minimize the possibility of worst conditions. And the question should be put into the context of visual interactions based on mobile devices with ordinary camera. Statistical mechanics provides a lot of choices to refine the heuristics and re-organize the combined scheme. General speaking, for the problems such as visual recognition, our design efforts may in fact be influenced by knowledge of how these are solved in nature, both in algorithms we employ and in the design of special-purpose application, and what we want the computer to do is to optimize and select the heuristics that can lead to good solutions by statistical learning.

Figure 2.1 shows the components of a typical pattern recognition system (Duda et al., 2001). The input to a pattern recognition system is often some kind of a transducer, such as a phone camera in the work of this thesis. Segmentation is one of the deepest problems in pattern recognition, which usually requires that the target objects can be segmented from the images before they have been categorized. Closely related to the problem of segmentation is the problem of recognizing or grouping together the various parts of a composite object. The traditional goal of the feature extractor is to characterize an object to be recognized by measurements whose values are very similar for objects in the same category, and very different for objects in different categories. This leads to the idea of seeking distinguishing features that are invariant to irrelevant transformations of the input. As with segmentation, the task of feature extraction is much more problem- and domain-dependent than is classification proper, and thus requires knowledge of the domain. However, some of the principles of pattern classification can be used in the design of the feature extractor. The task of the classifier component proper of a full system is to use the feature vector provided by the feature extractor to assign the object to a category. The conceptual boundary between feature extraction and classification proper is somewhat arbitrary: an ideal feature extractor would yield a representation that makes the job of the classifier trivial; conversely, an omnipotent classifier would not need the help of a sophisticated feature extractor. The distinction is forced upon us for practical rather than theoretical reasons. Anyhow, the abstraction provided by the feature-vector representation of the input data enables the development of a largely domain-independent theory of classification. Finally, the post-processor uses the output of the classifier to decide on the recommended action.

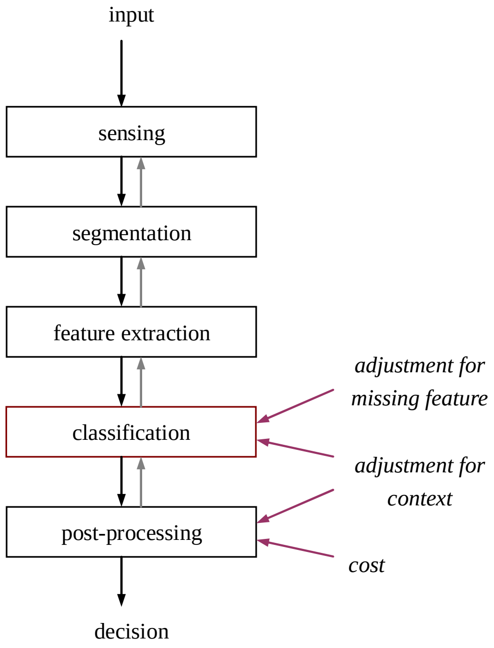


Figure 2.1. Components of a typical pattern recognition system.

The choice of the distinguishing features is a critical design step and depends on the characteristics of the problem domain. We obviously would like to find features that are simple to extract, invariant to irrelevant transformations, insensitive to noise, and useful for discriminating patterns in different categories. Heuristics can help to select the most valuable features from a large set of candidate features, where prior knowledge plays a major role. If we take the segmentation task as a two-category classification problem, we could look at the system from the same view point that the design of a pattern recognition system entails the repetition of a number of different activities: data collection, feature selection, model selection, training, and evaluation. Using of prior knowledge enables a probable good start with kind of heuristics, and statistical training normally decides the selected features and models to be used in classification.

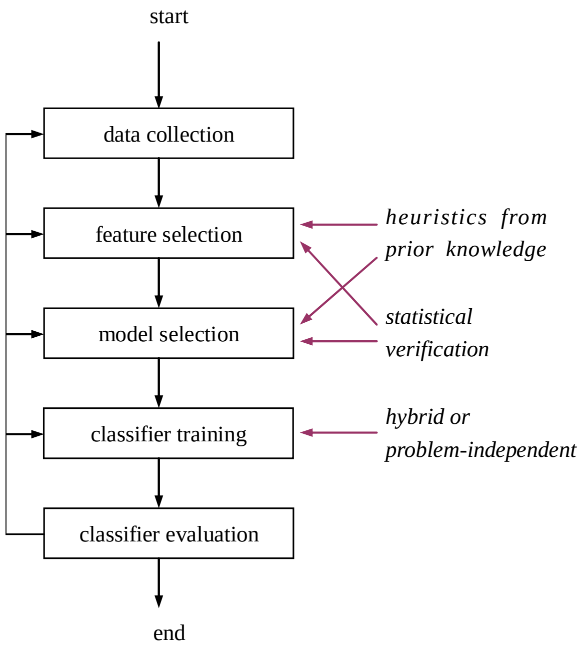


Figure 2.2. A design cycle of a pattern recognition system: combination of heuristic and statistical approaches is applied.

Figure 2.2 illustrates the diagram of the combination that could be commonly used in the design of a pattern recognition system. Heuristics often are inspired by the abstraction of structural information that is learned by human observation and perception. As can be seen in the following chapters, a set of heuristic methods were developed to solve the problems raised by the visual interaction systems, which can be implemented within the constrains presented by a mobile phone. For instance, characters or texts of a script are typical man-made symbols that are easy to be identified from an image with human knowledge. It is easy to notice that a significant feature of most text regions is that there are uniform drawing-like stroke groups with certain alignment and often with uniform color and size. In Chapter 3, we utilized this feature to develop a multi-group decomposition method which extracted text in different image layers based on connect component analysis. In Chapter 4, texture analysis was applied to detect bar codes from their background, which was based on the heuristic that both 1D and 2D bar codes contain certain patterns of textures. In Chapter 5, horizontal component extraction was used as the heuristic method to perform a fast search of face candidates in the coarse search step. The structural information and features such as color, shape, and many kinds of texture are usually employed as candidate features for object segmentation and thus affect model selection.

To minimize the risks to produce very bad results, statistical approaches are introduced to sift the features and help to decide the model and create the classifier. The purpose of statistical verification of the heuristics is to control the cost under an acceptable level. Classifier training is often a problem-independent stage, however, the performance of a classifier can still benefit from the previous steps of feature selection and model selection. Again statistical training methods or knowledge trees, or any hybrid versions can be used for classifier design. In this thesis, boosting algorithm was used to extract text, detect faces, and select features of facial expression analysis, which was proved to be effective to improve the system performance in cooperation with the heuristic methods applied previously. Subspace methods, such as manifold learning algorithms, were used to analyze and synthesize facial expressions. Generalized lerning vector quantization method was applied to recognize blurred barcodes. It should be kept in mind that in any practical problem the combination of heuristic and statistical approaches has no definite forms and should be put into the context of applications.

The reason to address the question on the combination of heuristic and statistical approaches is that it is a preferred solution of image processing applications for visual interactions based on mobile camera phones. As we have mentioned before, there are many technical issues caused by the limitation of hand held equipments, e.g., pool imaging quality, limited computation power and memory. Start with some heuristics and then use some statistical methods to fine tune the algorithms for different applications could probably be an appropriate way to guarantee both the real-time interaction and the desired accuracy. Contribution of this thesis resides in two aspects: firstly, new features and approaches were proposed either as heuristics or statistical means for concrete applications; secondly engineering design combining seveal methods for system optimization was studied and desired performance was achieved.

# Chapter 3

# Text Extraction from Natural Scene Images

Text detection in images with complex backgrounds is a novel research area that main findings happened in the recent fifteen years. Other than normal object detection problem, text in the complex background has significant within-class variance, especially for different fonts and language scripts. What we are interested in is to find a universal solution of text extraction problem, which means it should not only be able to deal with different types of images (e.g., book covers, web pages, natural scenes, etc.), but also have good generality on different types of characters and languages.

## 3.1 Related work

From application point of view, the current research work on text extraction can be classified into two categories, namely, text detection with specific preconditions and text detection with no limitation. Based on the algorithms adopted, there can be also two typical categories: region-based approach and texture-based approach. In this section we will review the research work from both viewpoints.

### 3.1.1 Text detection with specific preconditions

Most existing research efforts were put on text detection in a well controlled environment, where preconditions are added to the algorithms so that a series of application-dependent priors can be applied. In these cases, only the specific type of text that has salient characteristics is concerned. Table 3.1 lists some work in this category.

Some algorithms do not require strong assumptions explicitly, however, they still use pre-defined rules and hypothesis. Normally these kinds of algorithms can achieve good detection results because application-dependent prior knowledge has contributed a lot, which manifests that text detection for specific applications has already made commercial value.

Table 3.1. Some work of text detection with specific preconditions

| Authors | Preconditions (application domain) |
|---|---|
| Ohya et al., 1994 | Scene images; only including signboard and car plates; language: English, Japanese |
| Zhong et al., 1995 | Complex color images; only including CD and book cover images; language: English |
| Bunke and Kronenberg, 1999 | Book and journal cover images; language: English |
| Zhong et al., 2000 | Caption detection from video frames; language: English |
| Jain and Yu, 1998 | Book and Magazine covers and caption from video frames; language: English |
| Kim, 1999 | Journal cover images; language: English |
| Suen and Wang, 1997 | Scanned color prints like CD covers; language: English, Chinese |
| Chen and Chen, 1998 | Technical journal's cover images; language: English |
| Li et al., 2000 | Scene text and graphic text (e.g., captions) in video; language: English |
| Piccioli et al., 1996 | Road sign images; language: English |
| Zhou and Lopresti, 1997 | WWW images containing text; language: English |

### 3.1.2 Non-limited text detection

Comparing with preconditioned text detection, research on the universal solution of text detection without deliberate limitation is still in the preliminary stage. These years researchers tried to find efficient algorithms that have good performance and do not require well controlled environment, however, from a few open tests, e.g., ICDAR2003 (Lucas et al., 2003), the detection rate was not more than 50% that can not be used in real applications.

Even though a large number of algorithms have been proposed in the literature, no single method can provide satisfactory performance in all the applications due to the large variations in character font, size, texture, color, etc, referred to (Jung et al., 2004) for an excellent survey. Another issue is that there is a clear need for a public domain and representative test database for objective benchmarking. The lack of a public test set makes it difficult to compare the performances of competing algorithms.

### 3.1.3 Region-based text detection

Whether there are preconditions or not, the algorithms of text detection can be classified into two types: region-based approach and texture based approach. The region-based approach utilizes the features of text regions such as color, gray-level, and the differences between the corresponding features of the background to extract text in images. This approach employs so called bottom-up strategy, which first extracts the bottom structures such as connected components and edges from an image, then groups those bottom structures together to form the outline block of the text regions. Figure 3.1 shows a typical flowchart of text extraction based on connected component analysis.

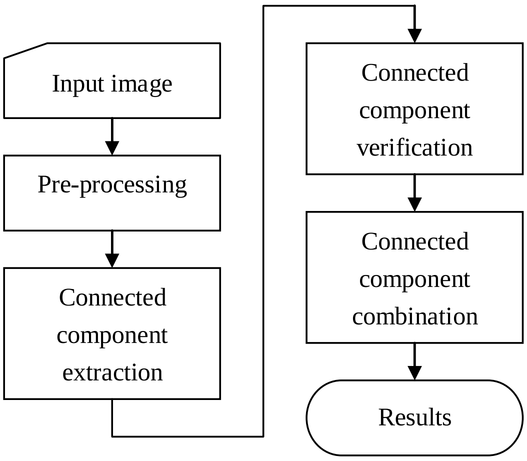

Input image
Pre-processing
Connected component extraction
Connected component verification
Connected component combination
Results

Figure 3.1. Flowchart of text detection based on connected components.

Jain and Yu proposed a typical text detection method based on connected component analysis (Jain and Yu, 1998), which utilizes color information of the text and its background image. First bit-drop and color quantization will be applied to the input color image to compress the color scale, then the original image is decomposed into several foreground images corresponding to different colors. Block Adjacency Graph (BAG) algorithm is employed in each foreground image to extract connected components and finally, all the connected components detected from different foreground images are combined to obtain the ultimate result. The reported text detection rate was about 72%. Ye and Cheriet (Ye and Cheriet, 2001) further proposed a stroke model which enables detecting thin connected components selectively, while ignoring relatively large backgrounds that appear complex.

The connected-component-based algorithms have relatively lower computation complexity, and are easy to be implemented especially for mobile platforms, meanwhile they can usually achieve good performance in many applications. However, this kind of algorithms is sensitive to background noise, and more seriously, there are many rules and thresholding processes during the extraction, verification, and combination of connected components that restricts the capability of generalization.

Another region-based approach utilizes edge information, which supposes that the contrast between text region and its background normally gets high enough to be distinguished. This method first extracts edges in the image using kinds of edge detectors, e.g., Canny detector (Canny, 1986). Afterwards, the non-text regions will be removed according to a series of rules, and then the edges of text will be verified and combined based on morphological or smoothing filters. It can be seen that this method largely depends on the edge detection performance, whereas different edge detectors have their specific properties and use cases, it is not easy to find a universal solution based on edge detectors. Secondly, sometimes the edges of text and the edges of the background regions can not be discriminated easily, therefore the edge-based algorithm can only fit in with some specific applications.

### 3.1.4 Texture-based text detection

Algorithms in this category take totally different ways from region-based approach. It is assumed that text regions have distinctive texture characteristics and can be separated from the background based on some texture analysis methods.

Kim *et al.* (Kim et al., 2003) proposed a texture-based text detection algorithm that introduces Support Vector Machine (SVM) to train the text samples and non-text samples so as to get the texture classifier to distinguish text regions. The algorithm uses the texture classifier to scan all the windows in the image, and based on the classification result of all the scanning windows, the Text Probability Image (TPI) is generated. Finally the text detection result is obtained by the search of TPI.

Texture-based text detection algorithms inspect the neighbors of pixels, so they will not be affected by the background noise if the noise did not change the texture characteristics of the neighborhood regions. Statistical trainings are introduced into texture classifier design, which brings robustness and generality and makes advantages. However, the main problem of this kind of methods is the computational complexity because all of the regions in the image will be searched, and the texture analyzer itself will cost much time. Another concern is about the accuracy of text location. Since texture analysis is based on pixel neighbors, it is difficult to achieve the same accuracy level as what can be done by the connected-component-based algorithms.

## 3.2 Text extraction based on multi-group decomposition

Though different types of text in images have common properties, they also have significant variations that make text detection problem difficult. Table 3.2 gives a classification list of different types of text based on different criterion, and the types concerned in this thesis are marked with grey background.

Traditional text detection algorithms that are based on connected components will have degraded performance if the background gets complex or illumination varies, mainly because the method of extracting connected components is not robust enough to deal with high complexity. Meanwhile the verification method used for identify the real text region does not consider the information contained inside the connected component regions, therefore the verification result is not satisfied, especially for high false acceptance rate.

In this thesis, a multi-group decomposition scheme is presented [P1, P2], which integrates multiple solutions for connected component extraction that make the algorithm have good generality, especially for complex background and illumination variations. The proposed algorithm also introduces a hybrid method for text region verification, which combines geometrical information, texture, and recognition confidence to achieve higher accuracy, especially for non-letter and isolated text. Figure 3.2 shows the flowchart of the approach based on multi-group decomposition.

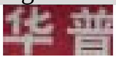
As shown in Figure 3.2, there are four groups of sub-images used in the algorithm. The first group is based on the results of color clustering in hue space. Since hue component is insensitive to illumination variance and shadows, an unsupervised clustering approach is applied to find clusters of similar colors based on the hue values, and the pixels with same color cluster label will be decomposed into the same image layer. In order to overcome the segmentation difficulty in hue space when the saturation is low, other two groups are introduced according to the segmentation results of weak (monochrome-like) color and gray-level respectively. These two groups have great benefit for extracting small characters in scene images using coordinate-dependent thresholding method. The last group contains edge images, which is helpful for text verification. To identify whether the located block candidates contain true text or falseness, some heuristics are employed, such as the size, eccentricity, foreground saturation, edge saturation, edge density variation, and align values. Setting of the parameters and combination of the heuristics could be a problem that attracts attention. Bootstrap algorithm for statistical learning (Duda et al., 2001) is applied to examine a proper parameter set, where random selection process is independently repeated *B* times, and the bootstrap estimate of the parameters is merely the mean of the *B* estimates on the individual bootstrap data sets. The feedback from recognition result is also very helpful for verifying a candidate text region because a false character generally has low recognition confidence. The recognition confidence derives from the error distance of the first two candidates in the recognition result list, as following

$$C_i = \left| \frac{D_i^1 - D_i^0}{D_i^0} \right| , \qquad (3\text{-}1)$$

where $C_i$ denotes the recognition confidence of the region *i*, $D_i^0$ and $D_i^1$ stand for the error distances of the first two recognition candidate, respectively.

Because the output of text extraction is for character recognition and other post processings, and most recognition engines use binary image as input, it is beneficial if the text extractor can give binary information of the text regions, i.e., to identify the text as foreground rather than only give the position. In [P2] a Priority Adaptive Segmentation (PAS) algorithm is proposed to obtain accurate foreground pixels of the characters in the located text regions.

Table 3.2. Classification of the text types

| Criteria | Type #1 | Type #2 |
|---|---|---|
| Language script | Letter<br> | Non-letter<br> |
| Number of characters and their layout | Clustered text<br> | Isolated character<br> |
| Importance and focus level | Foreground text<br> | Background text<br> |
| Degree of deformation | Regular text<br> | Irregular text<br> |
| Background distribution | Uniform background<br> | Varied background<br> |

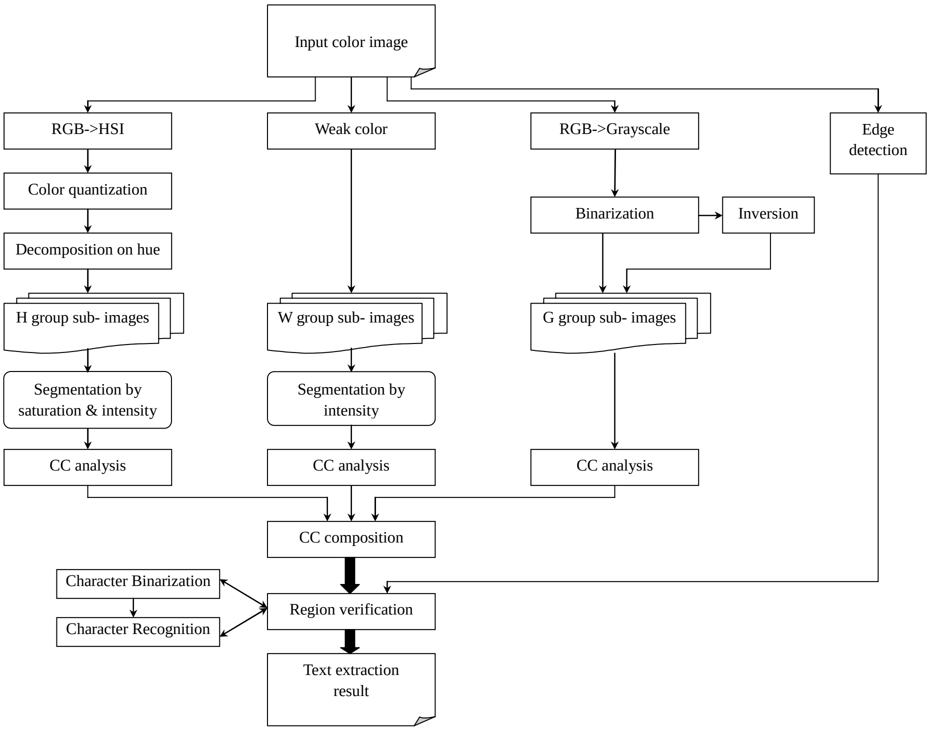


Figure 3.2 Flowchart of text extraction based on multi-group decomposition and connected component analysis.

# 3.3 Texture verification based on AdaBoost

The adoption of heuristics achieved promising performance in verification of text regions. However, it is still difficult to find a set of parameters that can adapt to various conditions. Texture-based text detection algorithms provide a solution that can investigate parameters of classifiers with statistical trainings, which will be more robust and have better generality for most situations.

## 3.3.1 Histogram of sub-patterns

A human observer usually can identify text from complex background without efforts even if he/she does not recognize the characters. In this case, the judgment is made based on the fact that the occurrence frequency of some sub-patterns in text regions is often much higher than that of non-text regions. Figure 3.3 shows the histograms of the output of a sub-pattern filter (vertical edge inspector) for text sample and non-text sample respectively, from which we can see obviously that the distribution of non-text samples converges to the peak near the zero point, whereas the distribution of text samples has larger spread. The observation indicates that the frequency of such sub-pattern in text regions is much higher than that of the non-text regions. Therefore we can choose the histograms of a series of sub-patterns as the features for text verification.

We collected four groups of sub-pattern histograms that have higher frequency appearing in text regions, Figure 3.4 illustrates some examples of the sub-patterns.

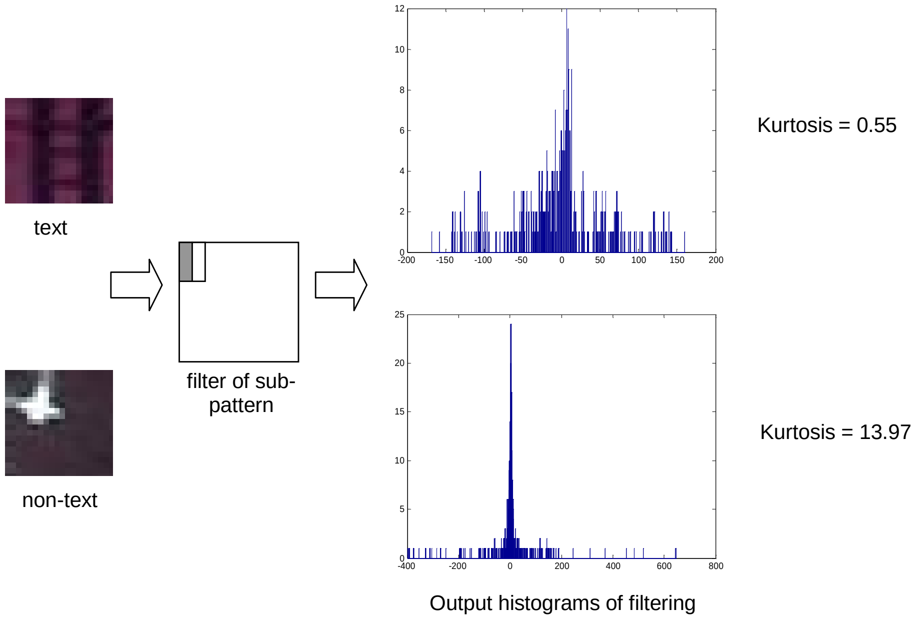


Figure 3.3. The output histograms of corresponding sub-pattern for text and non-text samples.

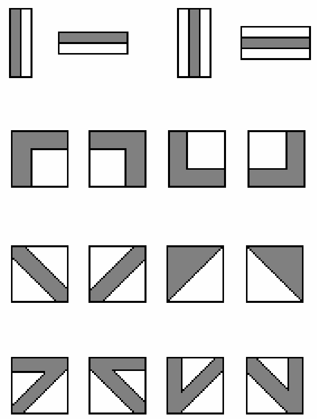


Figure 3.4. Examples of the sub-pattern filters.

In implementation, we use multiscale sampling in the regions to be verified because the text in images can have large variance in scale. In each scale, a fixed size sliding window (20×20 pixels) is used to calculate the histogram of each sub-pattern. Because the histograms of sub-patterns are insensitive to the types of the characters (letter, non-letter, clustered, isolated, regular, irregular, etc.), it is robust to adopt the histograms as text verification features.

### 3.3.2 AdaBoost algorithm

Text has very large inner-class variance, and the variance of non-text patterns is even larger. It is very difficult to devise a strong classifier directly to distinguish text and non-text regions. On the contrary, it is much easier to obtain a weak classifier whose accuracy is slightly better than random guess. Boosting learning algorithm introduced in the last ten years is one of the most powerful learning methods which attempts to boost the accuracy of any given learning algorithm and to produce new classifiers that are better able to predict examples for which the current ensemble's performance is poor. Thus we considered using Boosting algorithm to construct a strong classifier that combines the current weak classifiers which are based on the histograms of sub-patterns.

Freund and Schapire proposed AdaBoost algorithm (Freund and Schapire, 1997) that had been used in many applications for statistical learning. Assume that a sequence of *N* training examples $(x_1, y_1),...,(x_N, y_N)$ is drawn randomly according to distribution *P*, the basic idea of AdaBoost is to find a final hypothesis with low error relative to a given distribution *D* over the training examples. Unlike the distribution *P* which is set by "nature", the distribution *D* is only over the instances in the training set and is controlled by the learner. Ordinarily, this distribution will be set to be uniform so that $D(i) = 1/N$. The algorithm maintains a set of weights $\boldsymbol{w}^t$ over the training examples. On iteration *t* a distribution $\boldsymbol{p}^t$ is computed by normalizing these weights. This distribution is fed to weak learner **WeakLearn** which generates a hypothesis $h_t$ that (we hope) has small error with respect to the distribution. Using the new hypothesis $h_t$, the boosting algorithm generates the next weight vector $\boldsymbol{w}^{t+1}$, and the process repeats. After *T* such iterations, the final hypothesis $h_f$ is output. The hypothesis $h_f$ combines the outputs of the *T* weak hypothesis using a weighted majority vote. The AdaBoost algorithm is described in Figure 3.5.

### 3.3.3 Text verification based on AdaBoost

First of all, we need to construct a "weak" learning algorithm. From Figure 3.3 we can see that the output histogram of the sub-pattern filtering for the non-text sample has strongly positive kurtosis (calculated after normalization to unit variance), whereas the histogram of the text sample is only slightly supergaussian. Thus using the kurtosis measurement as a selected feature could be a suitable choice. In addition, the values of the bins near the zero zone and far away from zero point can be also collected as features of "weak" classifiers.

```
**Algorithm AdaBoost**
**Input**: sequence of $N$ labeled examples $\langle (x_1, y_1), ..., (x_N, y_N) \rangle$
        distribution $D$ over the $N$ examples
        weak learning algorithm **WeakLearn**
        integer $T$ specifying number of iterations
**Initialize** the weight vector: $w_i^1 = D(i)$ for $i = 1, ..., N$ .
**Do for** $t = 1, ..., T$
    1. Set   $\boldsymbol{p}^t = \frac{\boldsymbol{w}^t}{\sum_{i=1}^{N} w_i^t}$
    2. Call **WeakLearn**, providing it with the distribution $\boldsymbol{p}^t$; get back a hypothesis $h_t : X \to [0,1]$ .
    3. Calulate the error of $h_t : \varepsilon_t = \sum_{i=1}^{N} p_i^t \mid h_t(x_i) - y_i \mid$ .
    4. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$ .
    5. Set the new weights vector to be
           $w_i^{t+1} = w_i^t \beta_t^{1 - |h_t(x_i) - y_i|}$
**Output** the hypothesis
    $h_f(x) = \begin{cases} 1 & if \sum_{t=1}^{T} (\log \frac{1}{\beta_t}) h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \log \frac{1}{\beta_t} \\ 0 & otehrwise \end{cases}$ .
```

Figure 3.5. The adaptive boosting algorithm.

For each feature, we can compute the distribution of it for positive (text) samples and negative (non-text) samples in the training set, and use a simple binary tree classifier to make preliminary categorization:

$$h_i(x) = \begin{cases} +1 & if \quad (f_i(x) - \lambda_i) \cdot L_i > 0 \\ -1 & otherwise \end{cases} \tag{3-2}$$

where $x$ is the training sample, $f_i(x)$ is the value of the $i$th feature for the training sample, $\lambda_i$ is the decision threshold, and $L_i \notin \{-1, +1\}$ stands for the comparison direction of the feature value and the threshold. $h_i(x)$ is the classification output of the $i$th feature.

To form the final strong classifier, we can directly apply the AdaBoost algorithm presented in Figure 3.5. There is another option for the output of the classifier: rather than a “Yes or No” hypothesis, it is more beneficial that the output can remain some senses of probability which category the input sample is likely to be. In this case, we can create Text Probability Image (TPI) or compute the average probability of a region to be verified, and make final decision.

## 3.4 Hybrid approach and experimental results

Combined with the texture verification based on AdaBoost, a hybrid approach for text extraction from complex scene background can be built, as shown in Figure 3.6.

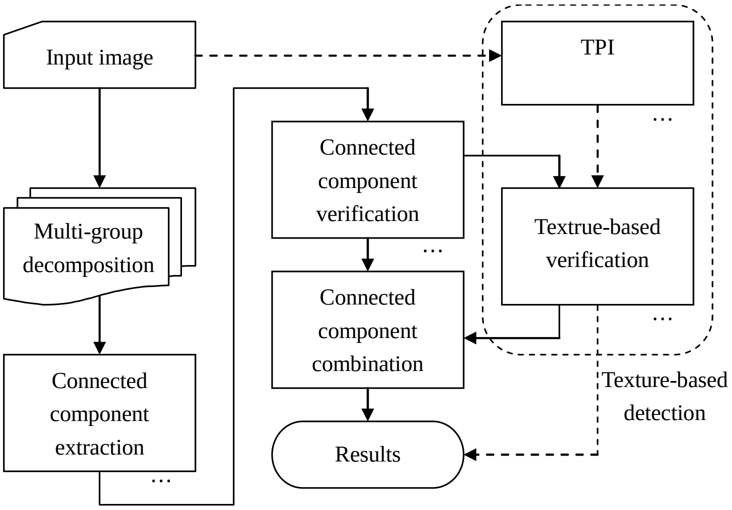

Input image
Multi-group decomposition
Connected component extraction
…
Connected component verification
…
Connected component combination
Results
TPI
…
Textrue-based verification
…
Texture-based detection

Figure 3.6. The hybrid approach of text extraction from complex scene background.

Because creation of TPI and search for text regions in TPI ("pure" texture-based approach) are time consuming, we composite the algorithm in a more effective manner: first, multi-group decomposition and connected component analysis in each image layer are performed to extract candidate regions; secondly the heuristic verification of the extracted connected components is applied to remove the obvious non-text regions; then TPI is created only for the remaining candidate regions in the corresponding scale, and the average confidence values of each region and possible alignment of the regions are calculated; finally the passed regions from different layers and scales are composed to get the results.

Our experiment was carried out on the database which includes more than 300 scene images that contain different kinds of characters. For practical reasons, we only considered the nearly 3,600 valid characters whose sizes are more than 16×16 pixels and can be recognized by OCR engines. The experimental result is shown in table 3.3. Only using geometrical properties as heuristics to verify the candidate text regions has achieved desirable result, and the introduction of recognition confidence both enhanced the detection rate and reduced the false alarm. The hybrid method which utilizes the texture-based probabilistic classifiers for text verification has obtained the best performance, which effectively reduced the false acceptance rate (FAR) meanwhile achieved least false rejection rate (FRR).

Table 3.3. Text detection performance (calculated in character level)

| Methods | Geometrical property | Geometrical property<br>+<br>Recognition confidence | Hybrid* |
|---|---|---|---|
| Detection rate (%) | 89.5 | 91.4 | 93.6 |
| False acceptance rate (%) | 12.3 | 10.6 | 5.4 |

* The test for hybrid approach was conducted after publications P1 and P2.

### Comparisons with the state of the art

The contribution of research work presented in publications P1 and P2, comparing with the state of the art at the time of publications, is as following:

- We proposed a multi-group decomposition scheme that split the original image into multiple layers. The novelty of this idea is to use the HSI color space instead of RGB to perform unsupervised color clustering, and use edge and region-based binarization as complement. This implementation structure improved the detection capability in various conditions.
- We adopted a run length smearing operation followed by BAG algorithm for connected component extraction. Comparing with the traditional line adjacency graph or other CC analysis methods, our CC extraction was much faster.
- We introduced comprehensive character verification method, esp. for the grouping of Chinese characters. Because the structure of Chinese characters is totally different from western letters (the components and radicals bring difficulties), specific processing is demanded. It is worthy of noting that our system produced character-level accuracy of text localization, and binarized characters can be directly recognized by OCR system. However, many systems just gave out rough bounded rectangles of text lines.

There had been very few attempts that quantitatively evaluate the text extraction algorithms, reported in the literature. Even for text detection, the performance measure is difficult because detection rate can not accurately reflect the precision of the algorithm. The ground truth data for text localization is usually marked by bounded rectangles that include gaps between characters, words, and text lines. However, our algorithm is very accurate and detects text at the character-level, it will not include the above gaps and thus may not have a good recall rate. Another concern is that the aim of each text localization system can differ. A method designed for text detection from scanned book cover may not be suitable for frames in video. Even for text detection in video, an application can focus on captions rather than scene text. The fact was that there was no substantial comparison between different algorithms reported in related literature at and before the time of publications P1 and P2.

To gain a better understanding of the relevance of the results obtained by the proposed method, benchmark comparisons with other methods (state of the art at the time of publications) were carried out. Because the implementation details in related work were not available, the benchmark algorithms were not carefully optimized. Therefore, the results should only be taken as baseline comparisons to facilitate the evaluation of the proposed method. Table 3.4 summarizes the performances of different systems. A was the proposed method without recognition confidence verification, B was A plus recognition confidence, C used the proposed hybrid approach, D used a top-down and bottom-up combination with RGB color quantization based on (Fujisawa et al., 1999), E used binary edge image to locate character-like components and identify them according to colors and locations (Suen and Wang, 1997), and F was a texture-based algorithm that adopted an NN to analyze the textural properties of the texts, followed by a CC analysis to detect the boundary (Li et al., 2000). The database I consisted of 60 images (of 640×480 pixels) containing pure Chinese characters. Database II were 48 images (of various sizes) randomly selected from the ICDAR2003 testing database, which were all of English letters. Because method D and F could not give character-level accuracy of the text blocks, a meditative measure was used to make the result comparable, as defined:

$$FRR = \frac{\# missed \quad pixels}{\# pixels \quad of \quad ground \quad truth}, \tag{3-3}$$

and

$$FAR = \frac{\#\,false \quad alarm \quad pixels}{\#\,pixels \quad of \quad ground \quad truth}. \tag{3-4}$$

Pixels in the gaps between characters were exclusive for character-level detection but included in other methods. Because the tested methods were implemented in different software platforms, the average processing time (in seconds) per image only reflected a rough estimation of the complexity.

Table 3.4. Performance of various systems

| Methods | Database I | | | Database II | | |
|---|---|---|---|---|---|---|
| | FRR | FAR | Avg. time | FRR | FAR | Avg. time |
| A | 6.6 | 5.1 | 0.37 | 47.8 | 33.2 | 2.9 |
| B | 6.4 | 4.2 | 0.79 | 50.2 | 26.7 | 3.8 |
| C | 5.3 | 2.2 | 2.8 | 42.3 | 20.2 | 5.6 |
| D | 10.5 | 23.7 | 4.8 | 59.1 | 62.4 | 10.3 |
| E | 18.2 | 19.7 | 0.26 | 61.6 | 69.1 | 0.91 |
| F | 7.0 | 12.6 | 9.5 | 44.7 | 79.5 | 24.2 |

Because the criteria presented by (3-3) and (3-4) is different from character-level measurement, and might not be suitable for all algorithms, the results only have relative confidence. It was noted that our approaches achieved better result for the test in Database I which only contains Chinese characters because our training set had similar property. Using recognition confidence got a bit higher FRR for Database II because our OCR performance for English letters was not good. Using AdaBoost for texture classification achieved best results for both databases, esp. for lowest FAR. Our method outperformed D and E because we adopted more comprehensive approaches for color decomposition and text verification. F had similar FRR but higher FAR, partially because the non-text samples used for training in the experiment were not carefully selected. D performed fastest because it only used edge information for CC analysis.

As for the current state of the art, even though a large number of algorithms have been proposed in the literature, no single method can provide satisfactory performance in all the applications due to the large variations in character font, size, texture, color, etc, referred to (Jung et al., 2004) for an excellent survey.

There are several information sources for text information extraction in images, e.g., color, texture, motion, shape, geometry, etc. It is advantageous to merge various information sources to enhance the performance of a text information extraction system. It is, however, not clear as to how to integrate the outputs of several approaches. In other words, most of the integration methods still followed the hybrid approach proposed in this chapter, or had some minor variations, e.g., the sequence of integration.

Another issue is that there is a clear need for a public domain and representative test database for objective benchmarking. The lack of a public test set makes it difficult to compare the performances of competing algorithms, and creates difficulties when merging several approaches.

Very little work had been done on scene text comparing with caption text in video. Besides text localization, the final text reading performance, i.e., taking count of the OCR result, will be more important for performance evaluation. In this sense, the current measure, either in character level or pixel level, may not reflect the overall performance of the text information extraction system.

Recent work intended to get improvement through the following ways. At first, texture-based approach had attracted much attention because it uses common features that text regions would have. Strong texture classifiers were designed, e.g., using SVM (Kim et al., 2003) and AdaBoost (Chen and Yuille, 2004). In (Kim et al., 2003), a CAMSHIFT search algorithm was adopted to locate the text regions based on the output of the texture classifier. In (Chen and Yuille, 2004), the authors put many efforts on selecting the features used for constructing the weak learners. Another attempt was to find more robust stoke filters that were elaborately designed based on local region analysis (Liu et al., 2006). The third concern was to try to find optimal integration of the outputs of different approaches. Comparing with the state of the art, our method used the CC-based approach before texture-based verification, which would be faster than the popular integration using the opposite order, and can provided character-level accuracy of the localization result. We also carried out a preliminary attempt to use AdaBoost for texture classification, though more elaborate work had been performed parallel.

## 3.5 Chapter summary

In this chapter we have addressed the problem of text extraction from complex scene background in camera images. Our aim is to find a generalized solution that can be adaptive to various use cases. Due to the complexity of both text and non-text samples, it is very difficult to construct a simple model to deal with the categorization.

From application point of view, the existing methods for text extraction can be classified into two categories: text detection with specific preconditions and text detection without any limitation. From the technical point of view, there can be also two typical categories: region-based approach and texture-based approach. All the current systems which have been implemented for commercial purposes, e.g., car plate detection, utilized much application-dependent knowledge to achieve good performance. On the contrary, non-restricted methods usually get poor results in an uncontrolled environment.

We introduced a multi-group decomposition scheme to deal with the variations of illumination, shadows and complex background, based on which text candidates can be extracted using connected-component analysis. A set of heuristics inspired by the geometrical property and recognition confidence of candidate regions were applied to identify true character-like regions from those candidates.

Texture-based verification algorithm for final decision was also proposed, which was the incremental work after Publication 1 and 2. AdaBoost learning method was introduced to combine a set of weak classifiers for constructing a stronger hypothesis. The hybrid scheme which integrates heuristics and probabilistic verification approaches achieved desirable performance in our experiments.

Although the proposed scheme was tend to give a uniform solution for text extraction problem, it was still more or less benefited from specific assumptions of the applications. To devise an indeed universal method is always a tough but attracting task. And it should be kept in mind that the details of implementation might be as important as the basic principle, for real applications.

# Chapter 4

# Bar Code Reading

Bar codes are being widely used in many fields for applications of great commercial value. By encoding a series of characters or symbols, bar codes are able to both carry explicit information and a database key. Nowadays, The availability of imaging phones provides people a mobile platform for decoding bar code rather than the use of the conventional scanner which is lack of mobility.

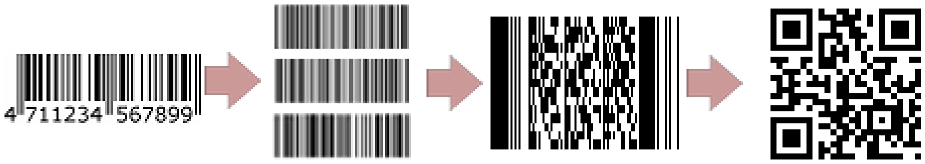
However, the short-distance capture of bar codes using an imaging phone inevitably makes bar code images blurred; meanwhile, these images are contaminated heavily with noises. Hence, it is a challenge for automatic bar code reading by imaging phones in such applications.

## 4.1 Bar code detection and segmentation

### 4.1.1 Brief view of bar codes

Simple 1D bar codes are just sequence of parallel light and dark stripes printed into background material. If the background is light enough only dark stripes i.e. bars have to be printed. Of course the same hold for dark backgrounds, when only light spaces are printed. The information included in bar code is coded to widths of the stripes or elements, where element is either one bar or space. Each element is formed from modules, which are the basic structures of bar codes. Usually the narrowest element contains one module. The wider elements are formed from several modules. Quite a lot of international 1D bar code standards had been introduced in the past half century, e.g., EAN-13 (Link-a), ISBNCode, UPC, Code128, etc., which are being used in department stores, factories, mail distribution centers, and banks.

2D bar codes were designed to carry significantly more data with higher information density and robustness than its 1D counterpart. There are two types of 2D bar code standards: stacked codes and matrix codes. A stacked bar code simply aligns multiple 1D bar codes in some specific combinations and put them together to form a larger "2D" code format. Comparing with stacked bar codes, the matrix types of 2-D bar codes have higher density because they indeed utilize two dimensional information rather than just stack rows of 1-D bar codes. The problem brought by matrix is that it can not be decoded by linear scans with traditional linear scanners.



1D bar code example

(EAN-13)

Multiple bar code layout

2D code with stacked bar codes (stacked code type)

2D code (matrix code type)

Figure 4.1. From 1D bar codes to 2D bar codes.

### 4.1.2 Bar code detection

Unlike scanning with traditional linear scanner that directly captures the waveform of the bar code, the image obtained by the phone camera may contains background objects other than the desired bar code. The purpose of bar code detection is to locate the bar code area so that its following processing can be carried out.

Any 1D bar code is composed of parallel and adjacent bars and spaces, which are horizontally aligned. Therefore the bar code region should be obviously dominated with vertical textures. Based on the knowledge, approaches of texture analysis are devised for detecting the bar code regions from their background. Jain and Chen utilized multichannel Cabor filtering technique to locate bar codes on plannar and curved surfaces (Jain and Chen, 1993), in which both unsupervised and supervised methods are proposed. In the unsupervised method, two highest radial frequencies and four orientations are used. The normalized pattern vectors after filtering are clustered into *k* grouped regions based on a squared-error clustering algorithm. And the bar code region in the segmented image is identified using the following heuristic: perform the connected component analysis, and remove all “small” or “large” connected components. Another heuristic, based on the aspect ratio, can also be used if more than one connected component survives the size heuristic. The supervised method uses a one-layer feedforward neural network to perform the 2-class classification, where rotation invariant features are selected as input vectors of the classifier. Muniz *et al.* used Hough-transform to locate and read bar codes that has the advantage of noise-insensitivity (Muniz et al., 1999). The problem of their approach is computational complexity, especially when the bar code contains many bars and spaces. Ando and Hontani proposed a method of feature extraction after categorization and projection for edges, ridges, corners and vertices, and extended the method to region-based image analysis with closely spaced features like bar codes (Ando and Hontani, 2001). The consequent application is to extract and read bar codes in 3-D scene. The first step is categorization of local image segments into unidirectional variation (UNIVAR), omnidirectionall variation (OMNIVAR), and no variation (NOVAR). The second step is devoted to detailed analysis along the informative axis of UNIVAR and OMNIVAR to localize edge/ridge and peak/corner/vertex, respectively. A bar code region can be represented as:

1). It is a global UNIVAR with a parallelogram shape (definition by region).

2). Opposing two sides of it are UNIVAR-NOVAR boundary and other two sides are UNIVAR-OMNIVAR boundary (definition by boundary).

Figure 4.2 illustrates the bar code region definition in the work of Ando and Hontani (Ando and Hontani, 2001).

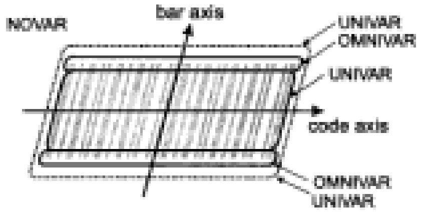


Figure 4.2. Distribution of UNIVAR/OMNIVAR/NOVAR around a barcode region.

Howlett *et al.* (Howlett et al., 1997) also proposed technique which locates bar codes using a neural network. Two neural network paradigms, i.e., the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) networks, are compared for use in their application. As reported, the system is sensitive to edges and certain types of textured or patterned surfaces, and the raised edge of an object sometimes gives raise to a mis-classification.

Similar to 1D bar codes, texture information is the dominant characteristic to be used for distinguishing the 2D bar code regions (both for stacked codes and matrix codes) from the background. Normand and Gaudin (Normand and Gaudin, 1994) performed bar code location directly in the spatial domain instead of using some frequential domains (Fourier, Gabor, or Discrete Cosine Transforms, etc.). The diagram of their system is shown in Figure 4.3. Four orientations have been considered: if the gradient along one of these orientations is more than a threshold value, the corresponding pixel is kept in the binary image of its maximum gradient. Thus, four binary images whose pixels are mutually exclusive are obtained. The next step is down-sampling, which speeds up the processing which follows, and furthermore filters the areas which do not have a high density of pixels of the same orientation. Next, complementary oriented morphological filters are applied to merge the blobs of pixels which are close together, and to eliminate the isolated ones. Finally connected components are extracted and some geometrical tests are performed to check the bar code consistence hypothesis (Gaudin et al., 1993).

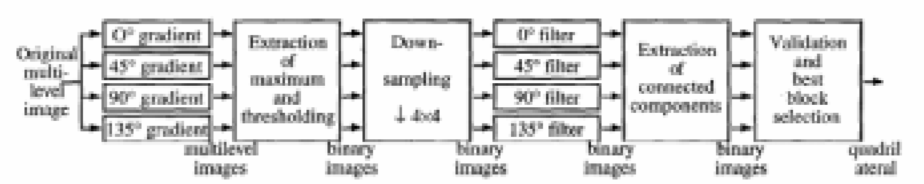


Figure 4.3. Bar code location method (Normand and Gaudin, 1994).

Ouaviani *et al.* (Ouaviani et al., 1999) performed Region of Interest (ROI) detection based on 3×3 Sobel masks, and locate the 2D bar code by initializing a set of candidate points inside the possible bar code regions. Region growing modifies the boundary of the region and convex hull computation ensures the segmentation of the whole code pattern even if it is not a connect one.

In our work of the thesis, a wavelet-based method is used to locate the 1D bar code region from a bar code image, which is proved to be robust even for complex background and blurred images. For 2D bar code location, we simply applied Hough transform to detect a quadrilateral area that is full of edges. The advantage of using Hough transform is that it is insensitive to noise, which is very common for the images captured by a camera phone.

### 4.1.3 Bar code character segmentation

1D Bar code from any standard usually contains start character, mandatory check character and stop character. The message that the bar code carries is located between start and check character. The purpose of bar code character segmentation is to extract the bar code information in symbol level, based on which the segmented characters will be input to the classifiers and decoding can be completed. Actually character segmentation and decoding can be performed at the same time if the quality of the input bar code image is good enough. In this case, the widths of bars and spaces can be extracted accurately, and a rule-based segmentation scheme is able to get all the character borders, meanwhile the code of each character is also calculated with a simple matching algorithm. Most existing work adopted this principle. However, for the blurred and noisy images captured by camera phones, simply comparing the bar/space width with the theoretical module width will not work in most situations. To solve this problem, we have to get the waveform segments of the characters first, and devise more robust classifiers to recognize each character as a whole signal, rather than decomposing the character further into bars and spaces, because accurate positions of the borders between bars and spaces are unavailable due to the blur. In our work we adopted an iterative algorithm to initialize and adjust the character borders based on the bar code structure. Multiple segmentation choices are given to the recognition engine, which later on determines the best candidate and output the decoding result.

For 2D bar codes, usually there is not necessary or feasible to segment characters before recognition is performed, mainly because the alignment pattern module placement may be interlaced in order to increase the capability of error-tolerance, e.g., Data Matrix. Instead, it is required to detect some specific patterns such as finder patterns, size indicators, orientation indicators, etc., so that the decoding operation can be implemented based on the obtained high-level information. Because these patterns should be designed for easy detection, normally it is not difficult to extract them after the bar code region is located.

## 4.2 1D bar code recognition

A 1D bar code can be interpreted as a waveform sampling along the code axis. An intuitive method to decode the bar code is to use edge information (Uusikyla, 2000), which is based on the detection of zero crossings of the second derivatives (ZC2).

Derivative operator emphasizes high frequency components of the signal, and that is why edge detection methods based on derivatives are very sensitive to noise. Using thresholding or lowpass filter the effects of noise can be reduced and the real edge can be distinguished from the false ones in some senses. Noise removal from bar code signal is more difficult than from most other signals, because non-blurred signal contains very high frequency components in which the noise is having. Lowpass filtering blurs the bar code edges and sometimes some of the edges may even disappear. An extreme condition is that the number of detected ZC2s is less than the real number of edges because of the blurring introduced by short-distance capture with camera phones, which make the edge-based algorithm impossible to recognize the bar code. Another difficulty comes from the non-uniform property of some input bar code images, in which part of the bar code is comparatively clear but other parts may be blurred, possibly because hand shakings during the imaging stage. A single filtering can not adapt to the complexity in this case.

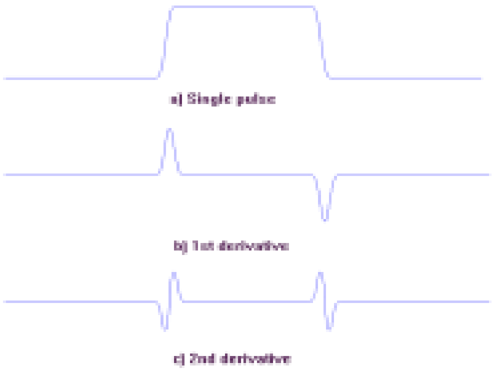


Figure 4.4. Edge detection using derivative operators. In (a) a single slightly blurred pulse is presented. From (b) it can be seen that the edges of the pulse are clearly visible in first derivative signal. To locate the exact locations of the edges, i.e. the extremum points of the 1$^{st}$ derivative, zero crossings of the 2$^{nd}$ derivative can be used.

Peaks contain slightly less information than edges but are preferred over edges because of their tolerance to convolution distortion (Joseph and Pavlidis, 1994). Ideally, all the peak locations and at least one edge location are required to reconstruct all the edge locations. Joseph and Pavlidis proposed an approach to use waveform peaks as a new feature that is more resistant to the blurring process (Joseph and Pavlidis, 1994). In addition, an estimate of the standard deviation of the blurring kernel is used to perform minor deblurring of the waveform. Statistical pattern recognition is used to classify the peaks as bar code characters. The noise tolerance of this recognition algorithm is increased by using an adaptive, histogram-based technique to remove the noise. The reported maximum blurring factor value of this approach is almost equal to the width of the unit stripe. One problem of this method comes from the Gaussian hypothesis of the blurring kernel, whereas in real conditions the blurring model may not be a Gaussian and even not symmetrical in different directions.

Most edge detection and blur reversing algorithms work well if the blurring is small relative to the unit stripe. Performance is then limited primarily by signal-to-noise ratio (SNR). Nevertheless, as relative blurring increase, reading errors occur in conventional readers that are independent of SNR. Increased signal does not, in these cases, reduce errors. In practice, it is difficult to estimate directly the blur factor. Turin and Boie (Turin and Boie, 1998) proposed to estimate the signal parameters indirectly by fitting the distorted signal to its model. By using the Kullback-Leibler divergence (KLD) between the signals, they developed a deterministic EM algorithm for solving nonlinear inverse problems and applied this algorithm to estimating the signal parameters. The algorithm is iterative, and it monotonically decreases the KLD. As a byproduct of the algorithm the blur factor and the pulse width of the signal are also estimated. The EM algorithm shows promise for extending the performance of bar code readers.

Shellhammer et al. (Shellhammer et al., 1999) proposed a cost-effective system to perform bar code reading, which is based on selective sampling scheme. The principle of their approach is to sample the edge information with both the time and the strength in order to control the cost of the edge detection circuit. In addition, the edge-enhancement filter helps to control the overshoot or undershoot effect. This solution is devised for the laser beam scanner, for which the distortion model may differ from that of the sampling with optical cameras.

We aim to provide a general 1D bar code reading solution for camera devices ranging from low-end to high-end. However, it is very difficult to robustly extract accurate features such as edges or peaks of the bars and spaces from the bar code images taken by a low-end camera phone.

In our work of this thesis, we proposed a statistical method which can analysis and learn the natures of the contaminated bar code symbols so that the trained classifiers are able to discriminate different code patterns by extracting their statistical features. In this way, it is not necessary to rely on the accurate extraction of the edge or peak locations of bar code waveforms, which is even unavailable in extreme cases.

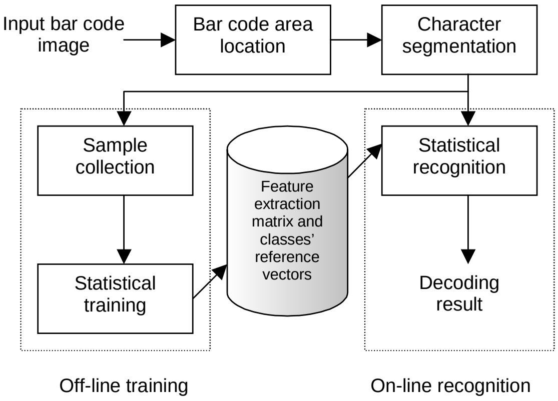


Figure 4.5. Diagram of the 1D bar code decoding system.

The proposed scheme is described in Figure 4.5. First, the bar code area in the image is located using a wavelet-based method. Then a rule-based segmentation algorithm is applied to extract bar code characters from the original waveform. The marked divisions, namely, the waveforms of the bar code characters, are input to the decoding block. After post-processing, the final code is output. In the statistical training line, samples (segmented waveforms of characters) of each code pattern are input to the training block and supervised learning is performed, based on the modified Generalized Learning Vector Quantization (GLVQ) method (Fukumoto et al., 2000). The output of the training block is a feature extraction matrix and class reference vectors, which are used in the on-line statistical recognition procedure. The initial feature extraction matrix and reference vectors are given by Linear Discriminant Analysis (LDA) and *K*-means clustering method respectively. To keep generality, a widely used 1D bar code standard, EAN-13, is chosen for our experiment. The system achieved desirable performance running both in Nokia 7650 and Nokia 3650 camera phones.

## 4.3 2D bar code design and decoding

The traditional one-dimensional bar codes have an apparent shortness in terms of information density (Pavlidis et al., 1992; Joseph et al., 1990). The vertical dimension does not carry any information but only provides a redundancy that is especially convenient for decoding by handset laser scanner when the user is not careful about the orientation and registration bounds. Nowadays more and more applications require a much longer bar code to encoding larger amount of information tips such as the price, product name, manufacturer, functionality, and expiration date of a product. Therefore the 2D bar codes were designed to carry significantly more data than its 1D counterpart.

An area scanner, such as a charge coupled device (CCD) scanner, is generally used in industries to scan a 2D bar code. The emergence of camera phones is changing the current status, as for 1D bar code reading, too.

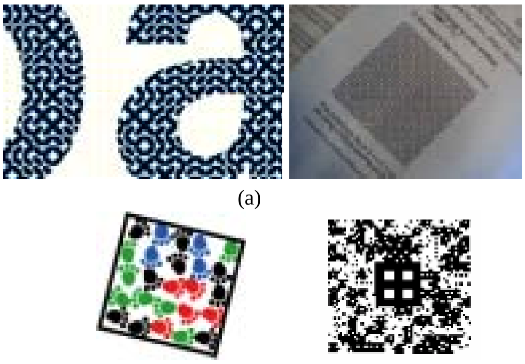
More than thirty 2D bar codes have been invented or redesigned since the first true 2D bar code, Code 49, was introduced nearly twenty years ago (Pavlidis et al., 1992) and a certain amount of research on 2D codes has been conducted. Combining with camera phones, however, questions concerning the performance of decoding issues come out: are the existing 2D bar codes suitable for scanning by the mobile phones? If yes, which one is the best choice? If no, is it necessary to design a new 2D bar code tailored for the mobile phones or even establish a standard of that?

H. Kato and her colleagues have put their efforts on the criteria for a 2D bar code used for a camera equipped mobile phone and identified factors that require certain attributes of the 2D bar code (Kato and Tan, 2005):

1. Use of mobile phone for image processing;
2. Being adapted to the nature of objects, such as robust error correction due to bar code damage;
3. Applications that are commonly offered for end users with a camera phone;
4. Multi-language encoding capability.

From the technical point of view, however, the requirements should be reviewed in a hierarchical perspective: low-level encoding deals with fundamental features such as bar code shape, orientation, and module characteristics that can directly affect image processing and low-level decoding, i.e., to identify each module with black and white (or even with grayscale or color if such information has been used); whereas those features related to applications, error correction, language, and additional aspects should be carried out by the high-level encoding. Therefore the design or selection of the 2D code for camera phones should focus on image processing requirements that are the main issues and special problems introduced by the phone camera whose imaging quality might not always be satisfactory for decoding of a high density 2D bar code.

Comparing with matrix code, stacked codes such as Code 49 and PDF417 do not fully utilized the two dimensional capacity so that they have relative lower information density than the matrix codes. To take full advantage of 2D information, we prefer matrix codes as the basis for 2D bar code design. Although a visual code can be of any shape – square, rectangle, circle, triangle, and whatever, it is easy to be recognized that pixel efficiency will restrict the choices of code shapes. Example of non-efficient use is ”Data Glyphs”, which needs terrible amounts of resolution to represent a single bit, although data glyphs can be hidden into pictures quite well, and be scanned with a high resolution scanner (Link-b). The problem is that data glyphs use diagonal stripes to represent bits (as shown in Figure 4.6a) – and we need to read these through bitmap data. Thus square and rectangle are preferable rather than circle and triangle or any other non-regular shapes. There are also different alternatives related to using colors in visual codes: colored codes (as shown in Figure 4.6b), monochrome code, and grayscale codes. After considering different media types that are used for keeping the code, colors and grayscale are not suggested to be used so that it can give support for wider media range that can bear the visual code and allows faxing and photocopying visual codes without losing readability. In addition, image capture by the phone camera in a variety of lighting conditions may not retain enough discriminate information of color and grayscale. Finally, scalability is an optional requirement that could support the future use of more powerful cameras by applying multi-layer data encoding and multi-resolution module alignment (as shown in Figure 4.6c).



(a)

(b) (c)

Figure 4.6. Example pictures of (a) Data Glyphs, (b) colored code, (c) layering data technique (using bigger cells and sub cells).

Many existing 2D bar codes, e.g., Data Matrix, QR Code, satisfied the criteria described above. The motivation of designing new codes mainly comes from the commercial value, i.e., the issues related to intelligent property rights. In addition, a simple solution that minimizes the cost of information header (code finder, size indicator, orientation indicator, error correction, etc.) is still helpful and attracts attention of research. In our work of the thesis, a simple 2D bar code design was proposed which has simple but effective indications and module placement for easy decoding. A patent application regarding this code design has been submitted.

Similar to the case of 1D bar codes, the decoding performance of 2D bar codes will be degraded mainly because of the poor quality of input images caused by short-distance capture using mobile phone cameras. Even if the module size of 2D bar codes is usually a bit larger than the 1D counterpart, the layout of the codes is spanned in two dimensions, which will be very sensitive to the geometrical distortion and blurring, and the decoding error can be transmitted to neighborhoods in both directions.

Ouaviani *et al.* (Ouaviani et al., 1999) meshed, resampled and binarized the 2D bar codes to compute the bit pattern. However, a direct binarization does not work if the image is too blurred and the mesh method will be damaged when there is nonlinear distortion for which simple linear transformation helps little. Figure 4.7 gives an example of binarization result, which is difficult to be used for decoding directly.

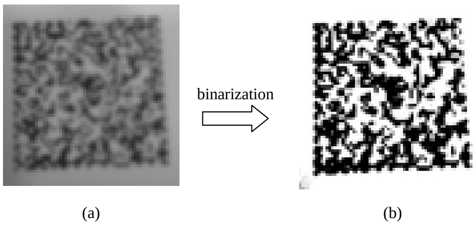


Figure 4.7. Binarization of 2D bar code.

Few attempts have been conducted to deal with 2D bar code decoding from poor quality images. An intuitive solution is to migrate the methods of 1D bar code reading and extend them for the 2D case. Three problems need to be addressed: first, the sampling scheme should be carefully considered because there are often nonlinear distortions such as barrel or pincushion distortion caused by the lens of the phone camera when the lens is moved very close to the target which is out of the focus range; second, deblurring is a necessary task since it is almost impossible for the user to capture clear image of the 2D bar code whose physical size is quite small; third, the parameters of decoding process should be adaptive to variations because the behaviors of users during the photo capture are different and not easy to be predicted so that the input images can vary a lot.

We proposed an integrated framework in this thesis to deal with the complexity of 2D bar code reading with camera phones. As illustrated in Figure 4.8, the code image is scanned using Bi-directional centripetal run-length (BCRL) method, where a mixed model is applied to perform decoding bit by bit, and an iteration ensures the adaptability to variations of input. In order to enhance the tolerance of high order distortion, grid vibration method is introduced to compensate the effect of non-linear distortion. The principle of the grid vibration is to shift the sampling position of each module by so called elastic matching approach. An efficient deblur approach is also introduced before decoding, which is based on the simulation of the blurring model for the phone camera and the estimation of blurring factors.

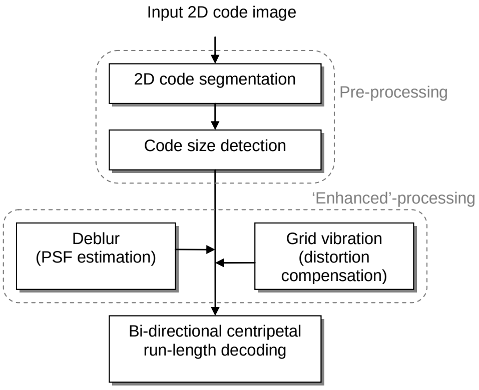


Figure 4.8 Framework of 2D bar code decoding method

## 4.4 Chapter summary

Bar code reading solution is a typical application-driven technique and the research interests are mainly impulsed by the limitations of optical equipments, i.e., the cameras integrated with mobile devices. Comparing with traditional usage of laser or CCD scanners, the images of bar codes taken by a phone camera are relatively poor, say, the resolution of the images is limited, the quality of the images is low since the bar code in the image is blurred and deformed caused by the too short distance photo-capture and perspective.

In this chapter we have reviewed existing work on 1D and 2D bar codes location, segmentation, and decoding. Bar code location algorithms utilize the texture characteristic that bar code regions are dominated by parallel bars and spaces (1D bar codes or 2D stacked codes) or well aligned black and white cells (2D matrix codes). Character segmentation is needed for 1D bar code reading when the input images are very blurred and extraction of edges is not robust. Because existing work did not use a low-end camera phone as an input device for bar code decoding, there are few comparable or referenced systems that can be directly applied in our work.

Our contribution was to propose an integrated framework both for 1D and 2D bar codes reading. Heuristics such as texture analyzers, edge detectors, are employed for bar code region location. Statistical training is introduced into the decoding stage, which can learn the distortion model and meanwhile estimate the suitable parameters for recognition and deblurring. We proposed the first system that recognized 1D barcode in character level using statistical approach comparing with the previous systems that can only perform in the single bar/space level.

Prior work had limitation in dealing with the maximum blurring factor value that is almost equal to the width of the unit stripe (Joseph and Pavlidis, 1994; Turin and Boie, 1998). And this is derived from a Gaussian hypothesis of the blurring kernel, whereas in real conditions the blurring model may not be a Gaussian and even not symmetrical in different directions. Another concern comes from the resolution of the phone camera. In our training and test database, the maximum width of a unit stripe is about 4.5 pixels, which may introduce non-trivial quantization error if traditional algorithms are applied. And in this case, the noise effect must be addressed.

We used EAN-13 and a self-designed code* for 1D and 2D bar code reading performance testing respectively. The actual blurring factor is larger than the width of the unit stripe. The experimental results proved the efficiency and effectiveness of our system.

* Design of the 2D code leads to a pending patent with US2007013767 (A1) and WO2007006385.

# Chapter 5

# Facial Image Processing and Expressional Manipulation

Faces are the most common objects that people observe everyday and play a key role in human communication. Therefore automatic facial image processing including face detection, facial feature extraction, face caricature, face recognition, and facial expression understanding, etc., has attracted much attention in the past decades. Because there are a lot of concrete applications driven by face images, research in this field will still stimulate people's interests in many sub-areas.

## 5.1 Face detection and facial feature extraction

Face detection and facial feature extraction play a very important role in many applications such as face recognition, human-machine interaction, and emotion understanding, etc. This section intends to give a review of those schemes that try to automatically find the facial feature regions or locate the accurate positions of the facial features and represent them with effective algorithms.

The methods of automatic extraction of facial features can be classified by their usage of three types of information: template matching, intensity and geometrical features (Ryu and Oh, 2001). In general, template matching requires many templates to accommodate varying pose whereas the intensity method requires good lighting conditions. Geometrical features utilize the information that a human face is always consist of several facial features with fixed geometrical relationship. Yet how to initialize the structure and evaluate the correction of the facial features under the structure is not very easy.

Here we classify the algorithms of facial feature detection into three typical categories: i) knowledge- based methods, including geometrical model, intensity correlation, which require much pre-knowledge about the skeleton structure of a human face; ii) the methods which utilize energy function judgment, give the determination by diverse forms of cost functions of some local facial parts or the entire combination of them, e.g., the templates of facial features, the corner or contrast models, and Hough transform; iii)  algorithms using statistical features, which are mostly like an computational approach to measure  the similarities between the detected facial features and the training features, where structural information is not explicit.

### 5.1.1 Knowledge-based methods

Heuristic methods are based on human knowledge but not the knowledge learned by machine itself. Human intuition can help to go into the problem more rapidly, however, bias is inevitably introduced at the same time. In general, those knowledge-based methods are easy to use, but it should be kept in mind that to extend the methods to other conditions with a theoretical framework is always required to achieve good generalization.

Jeng, *et al*. (Jeng et al., 1998) proposed an approach to detect facial features by using a geometrical face model constructed according to the relative geometrical relation among facial organs on a face. The basic idea is to remove small light details and to enhance the contrast with a preprocessing step, and then the processed image will be thresholded to produce a sketch-like binary image which preserves most of the important information including facial features. Afterwards a labeling step and a grouping algorithm will be used to group detected features block by block. These feature blocks are regarded as facial feature candidates. A geometrical face model shown in Figure 5.1 is used to check the candidates based on the distances defined in the model and a set of evaluation functions. The geometrical face model is constructed according to the real proportion between each facial organ.

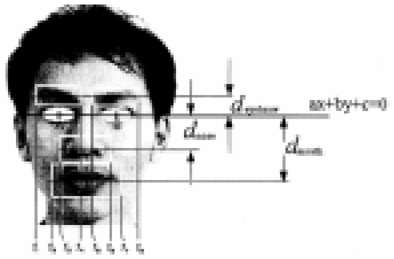
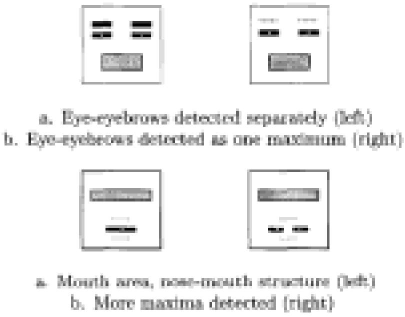


Figure 5.1. Geometrical face model (Jeng et al., 1998).

In (Wong et al., 2001), facial feature extraction based on the symmetrical measurement and projection minimum follows a face detection step which utilizes a genetic algorithm to search for possible face regions in an image and the eigenface technique to determine the fitness of the regions. Their method for detecting and extracting the facial features in a gray-level image is divided into two stages. Firstly, the possible human eye regions are detected by testing all the valley regions in an image. A pair of eye candidates is selected by means of the genetic algorithm to form a possible face candidate. The fitness value of each candidate is measured based on its projection on the eigenfaces. In order to improve the level of detection reliability, each possible face region is normalized for illumination; the shirring effect, when the head is tilted, is also considered as well. After a number of iterations, all the face candidates with a high fitness value are selected for further verification. At this stage, the face symmetry is measured and the existence of the different facial features is determined by evaluating the topographic relief of the normalized face regions. The facial features extracted include the eyebrow, the iris, the nostril, and the mouth corner.

In (Zhang and Lenders, 2000) the authors first use a typical histogram of face image to fast locate the face region where they suppose that there is only one face in the image. Then by analyzing the binary image obtained in the previous step, they locate the hair region, and the rough eye region that is enclosed in the hair region based on some conditions such as symmetry, parallel, and relative positions. In the next stage, they propose a knowledge-based edge detection, which includes iris locating, top and bottom points locating, upper eyelid edge detection, eye corner detection and lower eyelid detection, in the extracted eye region. One possible problem in their method is that the eye region might not be detected if the eyes or eyebrows are connected with the hair after binarization.

Huang, *et al*. (Huang and Mariana, 2000) proposed a structure model, which is a simplified version of geometrical face model, for face detection and precise eyes location. Figure 5.2 shows the two possible sub-structures for eyes region and local minimum feature for mouth area. All the relationship among the substructures should satisfy certain geometric conditions.



a. Eye-eyebrows detected separately (left)
b. Eye-eyebrows detected as one maximum (right)

a. Mouth area, nose-mouth structure (left)
b. More maxima detected (right)

Figure 5.2. Sub-structures model.

### 5.1.2 Energy functions

In the geometrical face model proposed in (Jeng et al., 1998), they had used so called evaluation functions to determine the fitness of the facial feature candidates. Methods that use the energy functions to compute the cost values of the possible facial features have a common sense: they all define some forms of metrics which measure the fitness of one or more facial features and the final judgment is based on the cost values. However, they differ from the basis by which the cost functions are established and the combination of those functions with other useful information. Previous knowledge is certainly very important to give the computational judgment.

Lau et al. proposed an energy function which is the sum of seven weighted terms for facial feature extraction (Lau et al., 2001). By allocating different values for the weighting factors, the function can extract different fiducial points. The algorithm begins by detecting the location of both irises as they are the most salient features on a face. The iris search region is defined as 2/5 height of the face and the seven terms of the energy function include central weighting function, mean crossing function, intensity function, edge response function, light spot function, eye white function, and eye space function. Search regions of fiducial points and the weighting factors are very important for feature extraction. However, from the paper, it is not clear how to choose the weighting factors except for empirical determination.

Perez, et al. (Perez et al., 2001) proposed a method for eye tracking with five stages: coarse and fine face detection, eye region initialization with maximum probability, map of the pupil location and pupil detection. The last stage uses line integral within and outside the pupil, which makes a form of cost function. It supposed that the pupil is dark and circular surrounded by a lighter area, which is not always satisfied in real cases.

Huang and Mariana proposed a scheme which detects precise location of eyes mainly based on Hough-transform, contour correlation, and region homogeneity (Huang and Mariana, 2000). A set of judgment functions are built to measure these three properties and select the best center and circle of the iris. This scheme requires that the eye is open and the edge can be clearly detected. The definition of the cost functions and the thresholds is based on prior knowledge and empirical adjustment.

Kawaguchi, et al., proposed an algorithm to detect the irises of both eyes from a human face in an intensity image with three stages (Kawaguchi, 2000; Rizon, 2000): first, intensity valleys, which they called as blobs, are extracted as the candidates for the irises; then for each pair of blobs, the algorithm computes a cost using Hough transform and reparability filter to measure the fit of the pair of blobs to the image; finally, the pair of blobs with smallest cost are selected as the irises. In their algorithm, a set of templates are defined for detection of blobs, calculation of costs and selection of the best pair of blobs, respectively as shown in Figure 5.3. As they mentioned in the paper, the templates they used are not sensitive to the variation of the face size and orientation.

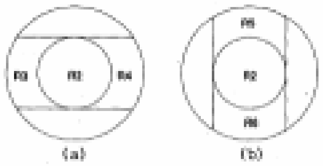


Figure 5.3. Templates used for iris detection.

In Lam and Li's paper, corners are selected as a salient feature to locate eyes in a head-shoulder image (Lam and Li, 1998). An earlier paper of Lam and Yan (Lam and Yan, 1996) also proposed the corner-based approach to detect eyes and gave a scheme to represent the shape of the extracted eyes based on snakes and deformable templates, which uses some new energy terms different to snakes. The eyes model they used is shown in Figure. 5.4. The model classifies the corners of an eye into one of six categories. After detecting the reliable corners, pairs of irises are grouped by using Gabor wavelet which extracts the low-level features at the candidate positions of irises. In order to verify the eye region, an eye-pair template is used which is obtained by averaging a number of eye regions with defined resolution. The difference between the eye region candidate and the eye template is measured as well as the symmetry of the possible eye region. A linear combination of the two measurements is used as the final cost and the region with lowest value is chosen as the final target.

There are some open questions have to be considered for the energy-function-based methods. For example, how should parameterization, such as threshold setting be implemented? How should a set of cost functions be combined to achieve best performance? And can we ensure that those functions work well if environment such as illumination changes? Prior knowledge plays a key role here, but general solution is still desired.

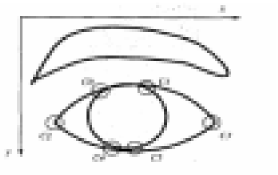


Figure 5.4. Eyes model and corners.

### 5.1.3 Statistical approaches

Statistical features are widely used in pattern analysis because they can in most cases be extracted more stably than structural information. Another advantage of the statistical algorithms is that they usually do not require much prior knowledge about the geometrical information of the facial features which is very important in the knowledge-based schemes. The statistical category includes many algorithms and feature extractors such as eigenfeatures (Xue et al., 2001), Gabor (Smeraldi and Bigun, 1998), Wavelet (Kim, 2004), ICA (Choi and Takaya, 2001), genetic algorithm (GA) (Lin and Wu, 1999) , support vector machines (SVM) (Osuna et al., 1997), boosting (Ma et al., 2004) and neural networks (Yeasin, 2000).

The principle of eigenface method for face detection and eigenfeature method for facial feature detection is based on Primary Component Analysis (PCA). Because the facial features such as eyes and mouth are distinctive in nature compared to other regions of the face, the general shape and texture of the eyes and mouth reside within a certain range of the space spanned by the eigenvectors and the eigenvalues describing their shapes and textures. This important property leads to a facial feature detector that not only uses a relatively small number of training samples, but also lends themselves to robustness against pose variations. Similarly, a face region can be well reconstructed by eigenfaces, whereas a non-face region normally does not have this property.

Some fast search approaches have been developed based on statistical features. Although the results are not guaranteed to be global optima, GA can solve the most troublesome task of template matching with acceptable speed. In (Lin and Wu, 1999), they used GA for the second stage of feature extraction. It follows from the experiment that the computational cost in the second stage feature extraction is largely reduced when the search range is large. In (Yen, 2002) the authors applied the edge density distribution of the image. In the preprocessing stage a face is approximated to an ellipse, and GA is applied to search for the best ellipse region matched.

Smeraldi and Bigun (Smeraldi and Bigun, 1998) proposed a saccadic exploration scheme of the Gabor decomposition. The human eye explores a visual scene by performing a sequence of large “jumps”, know as saccades, between the different points of interest. The proposed approach in their work is an attention-driven search based on a model of saccadic gaze movements. The algorithm is built around a sparse log-polar retinotropic grid. The Gabor decomposition is computed only on the points of the grid, so that the computational effort is greatly reduced. For the application of eye detection, a local appearance-based description of the search target (the eyes) is constructed by averaging the Gabor response from the center of the eyes of the persons in the training set.

Standard AdaBoost learns a “strong classifier” by combining a set of “weak classifiers” using a set of weights, where the selection of features and weights are learned through supervised training off-line. In (Ma et al., 2004), AdaBoost is used to learn conditional probability distributions of the pattern being eye or eye-pair. In practice, the conditional probabilities are extremely small for most parts of a face area where eyes do not appear.

Neural networks, an effective tool for learning examples and training the classifier, had been used by many researchers to extract facial features. The diversity is due to the variation of the neural networks structure, the forms of the input and output, the methods of training and evaluation, and the computational and storage abilities of the neural networks.

Computers can perform excellent learning capability if human had taught them with a suitable framework, i.e., the statistic architecture. Not so much prior knowledge is necessary as used in the heuristic methods, only a set of basic observations are needed for initialization. To get better performance with the statistical methods, the sample collection and method of training should be very important, which also require human interactions so that “pure” data-driven methods seldom work in practical cases.

### 5.1.4 Hybrid scheme

In fact most of the works presented above use hybrid architecture to implement facial feature extraction. The problem we concern about is whether several local optimal approaches will naturally produce a global optimal scheme. Unfortunately, a random combination may lead to an unknown result, if the conditions are not strictly controlled. However, facial features do have specific structure so that we certainly have some intrinsic restrictions that could be used as the principle to construct a powerful framework for facial feature extraction. Figure 5.5 illustrates the flowchart of face detection and facial feature extraction proposed by this thesis. Face detection is performed with two steps: first a coarse detection based on horizontal object extraction will detect face candidates with pre-defined structural patterns, which is fast and easy to be implemented; secondly the fine detection step is carried out which uses subspace method and AdaBoost algorithm to verify the candidate regions. Afterwards accurate eye location is employed based on the algorithm presented in (Ma et al., 2004), which gives the precise position of the eye-pair. Finally Multi-Resolution Active Shape Model (MRASM) is applied (a modified version based on (Efros and Leung, 2001)) to extract the facial features and represent the shape of the features as well.

Most of the horizontal textures congregate around the organs such as eyebrows, eyes, mouth and nostril. When the image is sub-sampled into low resolution, the character of horizontal textures is more stable and prominent. Therefore, a multi-scale and multi-level method is introduced to extract horizontal textures with different sizes and different conditions of illumination. After getting the object image of horizontal textures, possible pairs of eyes are searched by using three pre-defined templates and heuristic rules are constructed based on the structure of facial features. In terms of a given pair of eyes, the corresponding objects that represent nose, mouth, etc., are sought to get a face candidate.

Eigenface is used to verify the candidate faces that detected in the coarse detection step. Suppose that the face vectors are modeled by a random vector $\boldsymbol{x}$ with a covariance matrix

$$\boldsymbol{C} = E[\boldsymbol{x}\boldsymbol{x}^T]. \tag{5-1}$$

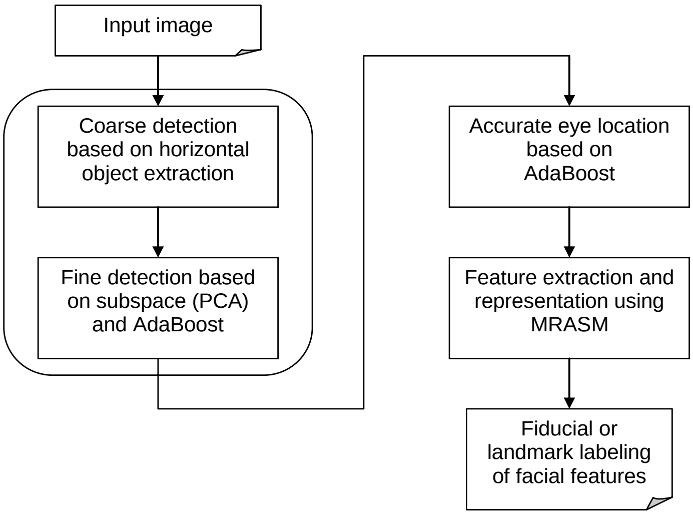

Input image
Coarse detection based on horizontal object extraction
Fine detection based on subspace (PCA) and AdaBoost
Accurate eye location based on AdaBoost
Feature extraction and representation using MRASM
Fiducial or landmark labeling of facial features

Figure 5.5. Flowchart of face detection and facial feature extraction.

The eigenvetors associated with the first few largest eigenvalues are referred to eigenfaces because they have face-like images. Specifically, suppose the eigenvectors of $\boldsymbol{C}$ are $\boldsymbol{u}_1$, $\boldsymbol{u}_2 \ldots \boldsymbol{u}_n$ and are associated with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Then

$$\boldsymbol{x} = \sum_{i=1}^{n} x_i \boldsymbol{u}_i \, . \qquad (5\text{-}2)$$

By selecting $m$ that $\lambda_i$ is small for $i>m$, reconstruction of $\boldsymbol{x}$ can be achieved by the $m$ eigenfaces

$$\boldsymbol{x}' = \sum_{i=1}^{m} x_i \boldsymbol{u}_i \, . \qquad (5\text{-}3)$$

For verification of candidate faces, the similarity measurement of vector $\boldsymbol{x}$ and $\boldsymbol{x}'$ can be applied because a non-face image lacks of facial structure and can not be represented well by eigenfaces. Therefore, if the similarity is small, the candidate will be rejected as non-face.

$sim(\boldsymbol{x}, \boldsymbol{x}') < T$, then to be rejected. (5-4)

Because there is no explicit way to select parameter $m$ and threshold $T$, and the selection of eigenfaces according to eigenvalues in descending order may not lead to optimal result, a training method based on AdaBoost which has been described in Chapter 3 is introduced. The positive set contains 2,057 face samples including different illumination conditions, poses, and properties (e.g., wearing glasses or not) and the negative set contains more than 16,000 non-face samples that mainly generated in the coarse detection step. A set of weak classifiers are generated based on formula (5-4), and the weights are trained according to AdaBoost algorithm (Figure 3.5). 219 images captured by digital camera (with resolution 640×480) and camera phone (with resolution 320×240) that contain 413 faces totally are used for testing, and the reported face detection rate is 93.7%, false acceptance rate is 3.6%.

After face detection and eye location, we try to extract facial features and find the way of interpreting the features with effective manners. A first popular shape model is active contour model, or called snake, which is an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges (Cass et al., 1988). Snakes lock onto nearby edges, localizing them accurately. In (Pardas, 2000), snakes are used to track the eyelids along a sequence, which allows large motion of the eyes and rotation and zooming of the head. However, the classical snakes deform to nearby edges without considering geometric structure, whereas for extracting facial features point-correspondences that have anatomical meaning are preferred. Some works (Shen, 2000; Chen, 2001; Yeasin, 2000; Park et al., 2001) tried to use modified versions of snakes to adapt to practical applications.

Although snakes can be directly used for facial feature extractions, the prominent geometric structure of human faces may not be well utilized. Instead, the active shape model could be a more qualified candidate to perform the interpretation of faces.

Active Shape Model (ASM) is a statistical model of shape which is used to represent objects in images (Cootes and Tailor, 2002; Huang et al., 2004). The shape of an object is represented by a set of points, which are commonly in two or three dimensions. The characteristic of ASM is that it allows to both analyze new shapes, and to synthesize shapes similar to those in a training set. By analyzing the variations in shape over the training set, ASM can mimic this variation using the shape constrains and find new objects with similar shape in an image.

The shape variation of faces is modeled by

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{\Phi b}, \tag{5-5}$$

where $\boldsymbol{\Phi}$ contains the $t$ eigenvectors corresponding to the largest eigenvalues, and $\boldsymbol{b}$ is a $t$ dimensional vector defines a set of parameters of a deformable model. By varying the elements of $\boldsymbol{b}$ we can vary the shape, $\boldsymbol{x}$. The variance of the *ith* parameter, $\boldsymbol{b}_i$, across the training set is given by the corresponding eigenvalue. By applying limits of the parameter $\boldsymbol{b}_i$, we ensure that the shape generated is similar to those in the original training set.

Thus an example of a model in an image is described by the shape parameters, $\boldsymbol{b}$, combined with a transformation from the model co-ordinate frame to the image co-ordinate frame. Typically this will be a similarity transformation defining the position, $(X_t, Y_t)$, orientation, $\theta$, and scale, s, of the model in the image.

The positions of the model points in the image, $\boldsymbol{x}$, are then given by

$$\boldsymbol{x} = T_{X_t,Y_t,s,\theta}(\overline{\boldsymbol{x}} + \boldsymbol{\Phi b}), \tag{5-6}$$

where the function $T_{X_t,Y_t,s,\theta}$ performs a rotation by $\theta$, a scaling by $s$ and a translation by $(X_t, Y_t)$.

Suppose now we wish to find the best pose and shape parameters to match a model instance $\boldsymbol{x}$ to a new set of image points, $\boldsymbol{Y}$. Minimizing the sum of square distances between corresponding model an image points is equivalent to minimizing the expression

$$f = |\boldsymbol{Y} - T_{X_t,Y_t,s,\theta}(\overline{\boldsymbol{x}} + \boldsymbol{\Phi b})|^2. \tag{5-7}$$

A simple iterative approach to achieving this is as follows:

1. Initialize the shape parameters, $\boldsymbol{b}$ , to zero.

2. Generate the model instance $\boldsymbol{x} = \overline{\boldsymbol{x}} + \boldsymbol{\Phi b}$ .

3. Find the pose parameters $(X_t, Y_t, s, \theta)$ which best map $\boldsymbol{x}$ to $\boldsymbol{Y}$ .

4. Invert the pose parameters and use to project $\boldsymbol{Y}$ into the model co-ordinate frame:

$$\boldsymbol{y} = T^{-1}_{X_t, Y_t, s, \theta}(\boldsymbol{Y}) . \tag{5-8}$$

5. Project $\boldsymbol{y}$ into the tangent plane to $\boldsymbol{x}$ by scaling by $1/(\boldsymbol{y} \cdot \overline{\boldsymbol{x}})$ .

6. Update the model parameters to match to $\boldsymbol{y}$ :

$$\boldsymbol{b} = \boldsymbol{\Phi}^T (\boldsymbol{y} - \overline{\boldsymbol{x}}) . \tag{5-9}$$

7. Apply constraints on $\boldsymbol{b}$ .

8. If not converged, return to step 2.

Convergence is declared when applying an iteration produces no significant change in the pose or shape parameters. This approach usually converges in a few iterations.

ASM applies a simple method of modeling the structure which has been found to be effective in many applications (though is not necessarily optimal). Essentially we sample along the profiles normal to the boundaries in the training set, and build statistical models of the grey-level structure, as shown in Figure 5.6.

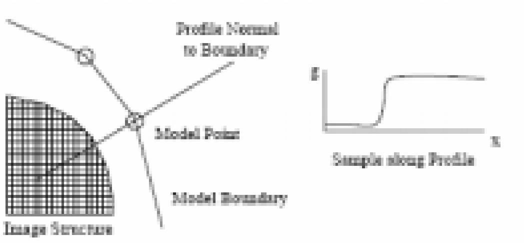


Figure 5.6. At each model point sample along a profile normal to the boundary.

Suppose for a given point we sample along a profile *k* pixels either side of the model point in the *ith* training image. We have 2*k*+1 samples which can be put in a vector $\boldsymbol{g}_i$ . To reduce the effects of global intensity changes we sample the derivative along the profile, rather than the absolute grey-level values. We then normalize the sample by dividing through by the sum of absolute element values,

$$\boldsymbol{g}_i \rightarrow \frac{1}{\sum_j |\boldsymbol{g}_{ij}|} \boldsymbol{g}_i . \tag{5-10}$$

We repeat this for each training image, to get a set of normalized samples $\{\boldsymbol{g}_i\}$ for the given model point. We assume that these are distributed as a multivariate gaussian, and estimate their mean $\overline{\boldsymbol{g}}$ and covariance $C_g$ . This gives a statistical model for the grey-level profile about the point. This is repeated for every model point, giving one grey-level model for each point.

The quality of fit of a new sample, $\boldsymbol{g}_s$ , to the model is given by

$$f(\boldsymbol{g}_s) = (\boldsymbol{g}_s - \overline{\boldsymbol{g}})^T C_g^{-1} (\boldsymbol{g}_s - \overline{\boldsymbol{g}}) . \tag{5-11}$$

This is the Mahalanobis distance of the sample from the model mean, and is linearly related to the log of the probability that $\boldsymbol{g}_s$ is drawn from the distribution. Minimising $f(\boldsymbol{g}_s)$ is equivalent to maximizing the probability that $\boldsymbol{g}_s$ comes from the distribution.

During search we sample a profile *m* pixels either side of the current point (*m*>*k*). We then test the quality of fit of the corresponding grey-level model at each of the 2(*m* - *k*)+1 possible positions along the sample as shown in Figure 5.7 and choose the one which gives the best match (lowest value of $f(\boldsymbol{g}_s)$).

This is repeated for every model point, giving a suggested new position for each point. We then apply one iteration of the algorithm above to update the current pose and shape parameters to best match the model to the new points.

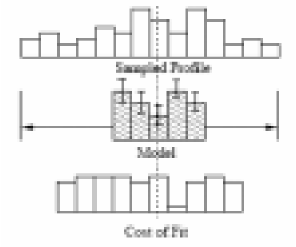


Figure 5.7. Search along sampled profile to find best fit of grey-level model.

To deal with the high-curvature feature points such as the canthus and the corners of the mouth, we modify the ASM algorithm to search for and match the high-curvature feature points by estimating the Gaussian model of the 2D grey-level structure and searching for these feature points in a 2D local region instead of along the 1D normal direction.

Considering the time consumption of ASM, we also adopt a multi-resolution version to speed up the convergence and improve the efficiency and robustness of the search procedure. The MRASM search algorithm is as follows:

1. set *L*=$L_{max}$.

2. While *L*≥0,

(a) Compute model point positions at level *L*;

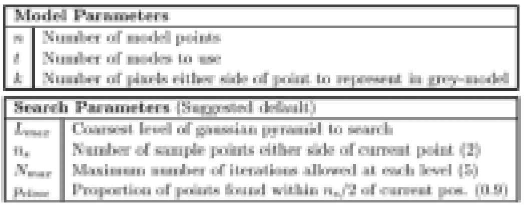
(b) Search at $n_s$ points on profile either side each current point;

(c) Update pose and shape parameters;

(d) Return to 2(a) unless more than $p_{close}$ (proportion of points found close) of the points are found close to the current position, or $N_{max}$ (Maximum number of iterations at each level) iterations have been applied at this resolution;

(e) If *L*>0 the *L*→(*L*-1).

3. Final result is given by the parameters after convergence at level 0.



| **Model Parameters** | |
|---|---|
| $n$ | Number of model points |
| $t$ | Number of modes to use |
| $k$ | Number of pixels either side of point to represent in grey-model |

| **Search Parameters** (Suggested default) | |
|---|---|
| $L_{max}$ | Coarsest level of gaussian pyramid to search |
| $n_s$ | Number of sample points either side of current point (2) |
| $N_{max}$ | Maximum number of iterations allowed at each level (5) |
| $p_{close}$ | Proportion of points found within $n_s/2$ of current pos. (0.9) |

Figure 5.8. MRASM search algorithm.

Figure 5.9 illustrates an example of facial feature extraction by ASM. The initialization is based on the result of eye location which makes the searching performance more efficient.

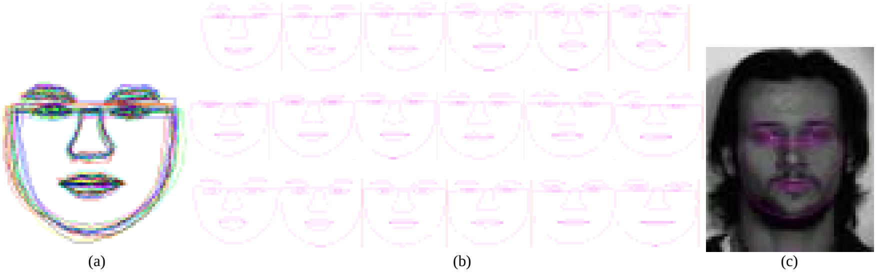


Figure 5.9. Example of facial feature extraction and representation by ASM: (a) aligned training set of facial features, (b) the first three modes of the model, (c) searching result.

## 5.2 Facial caricature

Facial image processing includes a series of applications that make funs and challenge the capability of machine learning. Facial caricature system is one of good examples that extracts the characteristics of human face and exaggerates those features to generate impressive caricatures as an artist could draw (Koshimizu et al., 1999; Nishino et al., 1999; Yamashita et al., 2000; Fujiwara, 2001).

Chen *et al*. (Chen et al., 2001) proposed a systematic solution of facial sketch generation using non-parametric sampling, which is an example-based scheme under a statistical framework. In the training phase, facial feature points are located in any input image, based on which the average shape of all input sketches is defined as meanshape. A geometric transformation is then defined to warp any input shape to the meanshape, and prior probability distribution is estimated. At runtime, facial feature points extraction and geometric transformation are also carried out to the input face to warp the shape to the meanshape. Non-parametric sampling (Efros and Leung, 2001) is performed to obtain the expected sketch image. Then the estimated probability distribution in the training stage is applied to obtain the refined sketch based on the sketch model, at last inverse geometric transformation is employed to get the final sketch.

In the work of this thesis, an image based drawing system is implemented for mobile platform, which is mainly inspired by (Chen et al., 2001)but further introduces parametric morphing algorithm to generate expressional effect and color processing to mimic watercolor painting style [P5, P6].

The contour sketch process utilizes multiple colors to represent the depth of an edge, and the sketch shape is directly generated based on the result of ASM search rather than probability sampling based on the sketch model in (Chen et al., 2001), which is simple but not loss of salient features. The morphing algorithm used in the system is based on smooth affine transformation. A set of basic rules corresponding to different emotional deforming parameters are defined and the affine transformation is carried out in several facial feature regions. The advantage of the parametric morphing algorithm is its simpleness and exaggeration effects, on the contrary, the non-parametric sampling costs much time.

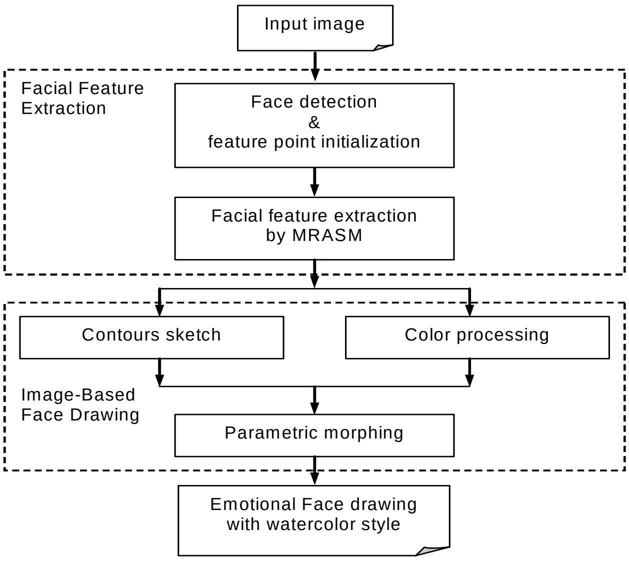

Input image
Facial Feature Extraction
Face detection & feature point initialization
Facial feature extraction by MRASM
Contours sketch
Color processing
Image-Based Face Drawing
Parametric morphing
Emotional Face drawing with watercolor style

Figure 5.10. Image-based drawing.

# 5.3 Expressional manipulation

There exist a number of applications for Human-Computer Interaction (HCI) to make use of automatic facial expression analysis. The main motivating principle for such applications is to give the ability to computers to adapt to the people's natural abilities rather than vice versa (Lisetti and Schiano, 2000). Facial expressions can indeed be considered as expressing communicative signals of intent, or expressing emotional inner states, or even as emotion activators.

Expressional manipulation presented in this section consists of two topics: facial expression recognition that analyses basic facial expressions and classify them into several prototypic emotional states, and facial expression synthesis which normally transforms an input face of an unknown person to one of the basic expressions.

## 5.3.1 Facial expression recognition

Development of an automatic facial expression analyzer has attracted great attention in the recent decades, an excellent survey can be found in (Pantic and Rothkrantz, 2000a).

### 5.3.1.1 The state of the art

In general, three main steps can be distinguished in tackling the problem of automated facial expression analysis. First, the face must be detected in a scene, which has been discussed in the previous sections. Next is to devise mechanisms for extracting the facial expression information from the observed facial image or image sequence. The final step is to define some set of categories, which we want to use for facial expression classification or interpretation, and to devise the mechanism of categorization.

One of the fundamental issues about the facial expression analysis is the representation of the visual information that an examined face might reveal. Some experiments suggest that the visual properties of the face, regarding the information about the shown facial expression, could be made clear by describing the movements of points belonging to the facial features (eyebrows, eyes, and mouth, etc.) and then by analyzing the relationships between those movements (Pantic and Rothkrantz, 2000a). Therefore different attempts were carried out to define point-based visual properties of facial expressions. Various analytic face representations yielded, in which the face is modeled as a set of facial points or as a set of templates fitted to the facial features. In another method called holistic approach, the face is represented as a whole unit such as a 3D wire-frame with a mapped texture and a spatio-temporal model of facial image motion. Hybrid approaches can be also applied, which typifies a combination of analytic and holistic approaches where a set of facial points is usually used to determine an initial position of a template that models the face.

After the face and its appearance have been perceived, the next step of an automated expression analyzer is to identify the facial expression conveyed by the face. A fundamental issue about the facial expression classification is to define a set of categories we want to deal with. Facial expressions can be classified in various ways – in terms of some prototypic expressions such as emotional expressions (anger, disgust, fear, happiness, sadness, surprise), in terms of nonprototypic expressions such as “raised brows”, or in terms of facial actions that cause an expression. The most known study on facial activity is the Facial Action Coding System (FACS) (Ekman and Friesen, 1978). FACS is designed for human observers to detect independent subtle changes in facial appearance caused by contractions of the facial muscles. In a form of rules, FACS provides a linguistic description of all possible, visually detectable facial changes in terms of 44 so-called Action Units (AUs). Using these rules, a trained human FACS coder decomposes a shown expression into the specific AUs that describe the expression. The benefit of automatic FACS has triggered researchers to take different approaches in tackling this problem.

Depending on the face model, a template-based or a feature-based method is applied for facial expression data extraction. Template-based methods fit a holistic face model to the input image or track it in the input image sequence. Feature-based methods localize the features of an analytic face model. Table 5.1 lists the examples of the methods utilized for facial expression information extraction.

Table 5.1. The methods for automatic facial expression data extraction

| Method type | Template-based |
|---|---|
| Description | Fit a holistic face model to the input image |
| Examples | Active Appearance Models (AAM) fitting (Edwards, 1998), labeled graph fitting by elastic graph matching (Hong, 1998), Point Distribution Model (PDM) fitting (Huang, 1997) , gradient-based optical flow (Yoneyama, 1997) |
| Method type | Feature-based |
| Description | Localize the features of an analytic face model to the input image |
| Examples | Brightness distribution extraction along 13 vertical facial lines (Kobayashi, 1997) multiple feature detectors per facial feature (Pantic and Rothkrantz, 2000b) |

Independent of the used classification categories, the mechanism of facial expression classification applied by an expression analyzer is either a template-based- or a neural-network-based- or a rule-based- classification method. Table 5.2 summarizes the utilized methods for facial expression emotional classification. In template-based classification methods, the encountered facial expression is compared to the templates defined for each expression category. The best match decides the category of the shown expression. There are a lot of combinations of different facial actions with different intensities, and everybody has his/her own maximal intensity of displaying a certain facial action, which introduces difficulties for template-based method. And it is also difficult to achieve a quantified recognition of a nonprototypic facial expression. The neural-network-based methods can be classified into template-based category, but a typical neural network can perform a quantified facial expression categorization into multiple classes while the template-based methods cannot achieve such a performance in general. The rule-based classification methods classify the examined facial expression into the basic emotion categories based on the previously encoded facial actions. The shown expression, described in terms of facial actions, is compared to the prototypic expressions defined for each of the emotion categories and classified in the optimal fitting category.

Table 5.2. The methods for facial expression classification

| | |
|---|---|
| Method type | Template-based |
| Description | Template matching decides the category of the shown expression |
| Examples | PCA based on Mahalonobis distance and LDA (Edwards, 1998), elastic graph matching (Hong, 1998), 2D emotion space (PCA) & minimum distance classifier (Huang, 1997) |
| Method type | Neural-network-based |
| Description | Neural network classification and may categorize into multiple classes |
| Examples | NN with backpropagation learning (Kobayashi, 1997) |
| Method type | Rule-based |
| Description | Comparison with prototypic expressions (emotional categories) in terms of facial actions |
| Examples | Expert system rules (Pantic and Rothkrantz, 2000b) |

In addition, Tian *et al*. developed an Automatic Face Analysis (AFA) system to analyze facial expressions based on both permanent facial features (brows, eyes, mouth) and transient facial features (deepening of facial furrows) in a nearly frontal-view face image sequence. The AFA system recognizes fine-grained changes in facial expression into AUs of the FACS, instead of a few prototypic expressions. However, the AFA system requires accurate locations of the facial features, which is difficult especially for some non-permanent features such as furrows and wrinkles. Chandrasiri *et al*. proposed Personal Facial Expression Space (PFES) to recognize person-specific, primary facial expression image sequences (Chandrasiri et al., 2004). On PFES, facial expression parameters which are compatible with MPEG-4 high level Facial Animation Parameters (FAP) can be extracted from a user's face image. The key limitation of PFES is that it can not process an unknown face that is not included in the trained person-specific space. In (Yeasin et al., 2006) Yeasin *et al*. used a subjective measurement of the intensity of basic expressions by associating a coefficient for the intensity by the relative image number in the expression image sequence. Though simple and effective for their application, this method does not align expression intensities of different.

It should be kept in mind that most existing expression analyzers perform a singular classification of the examined expression into one of the basic emotion categories. This approach has two main limitations. First, pure emotional expressions are seldom elicited. Most of the time, people show blends of emotional expressions. Second, it is not at all certain that all facial expressions displayed on the face can be classified under the six basic emotion categories.

#### 5.3.1.2 Manifold learning for facial expression analysis

In recent years manifold learning methods are used for facial expression analysis (Shan et al., 2005a; Hu et al., 2004; Chang et al., 2003), which is based on the fact that variations of face images can be represented as low dimensional manifolds embedded in the high dimensional image space.

***LLE***

The Locally Linear Embedding (LLE) recovers global nonlinear structure from locally linear fits, which is based on simple geometric intuitions (Roweis and Saul, 2000). Suppose the data consist of *N* real-valued vectors $\vec{\boldsymbol{X}}_i$, each of dimensionality *D*, sampled from some underlying manifold. Provided there is sufficient data (such that the manifold is well-sampled), we expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold. We characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. Reconstruction errors are measured by the cost function

$$\varepsilon(\boldsymbol{W}) = \sum_i \left| \vec{\boldsymbol{X}}_i - \sum\nolimits_j W_{ij} \vec{\boldsymbol{X}}_j \right|^2 , \qquad (5\text{-}12)$$

which adds up the squared distances between all the data points and their reconstructions. The weights $W_{ij}$ summarize the contribution of the *j*th data point to the *i*th reconstruction. To compute the weights $W_{ij}$ we minimize the cost function subject to two constraints: first, that each data point $\vec{\boldsymbol{X}}_i$ is reconstructed only from its neighbors, enforcing $W_{ij} = 0$ if $\vec{\boldsymbol{X}}_j$ does not belong to the set of neighbors of $\vec{\boldsymbol{X}}_i$; second, that the rows of the weight matrix sum to one: $\sum\nolimits_j W_{ij} = 1$. The optimal weights $W_{ij}$ subject to these constrains are found by solving a least-squares problem.

Suppose the data lie on or near a smooth nonlinear manifold of lower dimensionality $d << D$. To a good approximation then, there exists a linear mapping – consisting of a translation, rotation, and rescaling – that maps the high-dimensional coordinates of each neighborhood to global internal coordinates on the manifold. By design, the reconstruction weights $W_{ij}$ reflect intrinsic geometric properties of the data that are invariant to exactly such transformations. We therefore expect their characterization of local geometry in the original data space to be equally valid for local patches on the manifold. In particular, the same weighs $W_{ij}$ that reconstruct the *i*th data point in *D* dimensions should also reconstruct its embedded manifold coordinates in *d* dimensions.

LLE constructs a neighborhood-preserving mapping based on the above idea. In the final step of the algorithm, each high-dimensional observation $\vec{\boldsymbol{X}}_i$ is mapped to a low-dimensional vector $\vec{\boldsymbol{Y}}_i$ representing global internal coordinates on the manifold. This is done by choosing *d*-dimensional coordinates $\vec{\boldsymbol{Y}}_i$ to minimize the embedding cost function

$$\Phi(\boldsymbol{Y}) = \sum_i \left| \vec{\boldsymbol{Y}}_i - \sum\nolimits_j W_{ij} \vec{\boldsymbol{Y}}_j \right|^2 . \tag{5-13}$$

This cost function, like the previous one, is based on locally linear reconstruction errors, but here we fix the weights $W_{ij}$ while optimizing the coordinates $\vec{\boldsymbol{Y}}_i$.

### *LPP*

Locality Preserving Projections (LPP) is a linear approximation of Laplacian Eigenmap (He and Niyogi, 2003). It seeks a transformation $\boldsymbol{P}$ to project high-dimensional input data $\boldsymbol{X} = [x_1, x_2, ..., x_n]$ into a low-dimensional subspace $\boldsymbol{Y} = [y_1, y_2, ..., y_n]$ in which the local structure of the input data is be preserved. The linear transformation $\boldsymbol{P}$ can be obtained by minimizing the following objective function:

$$\min_{\boldsymbol{P}} \sum_{i,j=1}^{n} \left\| y_i - y_j \right\|^2 W_{ij} , \tag{5-14}$$

where $y_i = \boldsymbol{P}^{\mathrm{T}} x_i$, the weight matrix $\boldsymbol{W}$ is constructed through the adjacency graph with $k$ nearest neighbors or $\varepsilon$-neighborhoods. The minimization problem can be converted to solving a generalized eigenvalue problem as

$$\boldsymbol{XLX}^{\mathrm{T}} \boldsymbol{P} = \lambda \boldsymbol{XDX}^{\mathrm{T}} \boldsymbol{P} , \tag{5-15}$$

where $\boldsymbol{D}_{ii} = \sum\nolimits_j W_{ij}$ is a diagonal matrix, and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$.

### *SLPP*

When class information is available, LPP can be performed in a supervised manner (Cheng et al., 2005). The basic idea is to encode class information in the embedding when constructing the neighborhood graph, so that the local neighborhood of a sample $x_i$ from class $c$ should be composed of samples belonging to class $c$ only. This can be achieved by increasing the distances between samples belonging to different classes, as in (Ridder et al., 2003) and (Shan et al., 2005a), the following definition is used

$$Sup\Delta_{ij} = \Delta_{ij} + \alpha M \delta_{ij} \qquad \alpha \in [0,1] , \tag{5-16}$$

where $\Delta_{ij}$ denotes the distance between $x_i$ and $x_j$, $Sup\Delta_{ij}$ denotes the distance after incorporating class information, and $M = \max_{i,j} \Delta_{ij}$, $\delta_{ij} = 0$ if $x_i$ and $x_j$ belong to the same class, and 1 otherwise. The parameter $\alpha$ represents the degree of supervision. When $\alpha = 0$, one obtains unsupervised LPP; when $\alpha = 1$, the result is fully supervised LPP.

### *Manifold application in facial expression analysis*

Chang *et al.* (Chang et al., 2003) made first attempt to apply two types of embedding, LLE and Lipschitz embedding, to learn the structure of the expression manifold. In (Hu et al., 2004), they further proposed an approach for facial expression tracking and recognition based on Isomap embedding. One problem of these methods is that they learned the expression manifold in the feature space described by a large set of landmarks, e.g., using ASM, which requires complex extracting or tracking scheme and is not easy to be obtained accurately, additionally, the number of such landmark points is far beyond the number of fiducial points used in expression synthesis stage as we will discuss below.

Another potential risk is that the research was conducted on data sets containing only several subjects, the efficiency on a large number of subjects was not verified. Shan *et al.* (Shan et al., 2005a) first investigated an appearance manifold of facial expression based on a novel alignment method (SLPP) to keep the semantic similarity of facial expression from different subjects on one generalized manifold.

#### 5.3.1.3 BoostLBP for facial expression data extraction

Recently Local Binary Pattern (LBP) features are widely adopted for facial image representation and facial expression data extraction. The LBP operator is a powerful method of texture description (Ojala and Pietikainen, 2002). The operator labels the pixels of an image by thresholding the neighborhood of each pixel with the center value and considering the result as a binary number. Using circular neighborhoods and bilinearly interpolating the pixel values allow any radius and number of pixels in the neighborhood. Then the histogram of the labels can be used as a texture descriptor. An illustration of the basic LBP operator is shown in Figure 5.11. An extension to the original operator is to use so called uniform patterns. A LBP is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. We use $LBP_{P,R}^{u2}$ to denote a uniform LBP operator, which means using the operator in a neighborhood of *P* sampling points on a circle of radius *R*. Usually $LBP_{8,2}^{u2}$ is applied to extract LBP code for each pixel of face images. All feature values are quantified into 59 bins according to uniform strategy. A histogram of the labeled image $f_l(x,y)$ can be defined as

$$H_i = \sum_{x,y} I\{f_l(x,y) = i\}, \qquad i = 0,...n-1 \tag{5-17}$$

where *n* is the number of different labels produced by the LBP (for $LBP_{8,2}^{u2}$ *n* = 59) and

$$I\{A\} = \begin{cases} 1, & A \quad is \quad true \\ 0, & A \quad is \quad false \end{cases} \tag{5-18}$$

The histogram contains information about the distribution of the local micro-patterns such as edges, spots and flat areas over the whole image. For efficient face representation, feature extracted should retain also spatial information. Hence the face image is scanned with a scalable sub-window, thus a sequence of regions $R_0, R_1, ..., R_{m-1}$ is generated. A spatially enhanced histogram is defined as

$$H_{i,j} = I\{f_l(x,y) = i\} I\{(x,y) \in R_j\} \qquad i = 0,...,n-1, \quad j = 0,...,m-1 \tag{5-19}$$

In this histogram, the face is described on three different levels of locality: the labels for the histogram contain the pixel-level patterns, the labels are summed over a small region to produce information on a regional level and the regional histograms are concatenated to build a global description of the face.

When the image has been divided into regions, it can be expected that some of the regions contain more useful information than others in terms of distinguishing between different facial expressions. Intuitive weights can be assigned to different regions according to their importance to expression discriminance (Shan et al., 2005b). A more constructive approach was proposed by Huang *et al*. (Huang et al., 2005) that utilize Jensen-Shannon Boosting learning to select the most important features (regions).

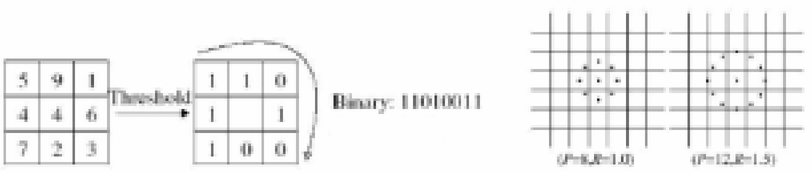


Figure 5.11. The LBP operator.

***JSBoost Learning***

Provided that we are given a set of labeled examples $\{x_i, y_i\}_{i=1}^{N}$ where $x_i \in R^d$ and $y_i \in \{-1,+1\}$, and are asked to give a decision $y$ for any input $x \in R^d$. Rather than using raw data, we may extract some features by mapping the raw data to a feature space using some mapping functions $\phi : R^d \to R$, where $\phi(\cdot)$ stands for LBP filters in this discussion. After a best feature $\phi_i$ is learned $(i = 1,...,k)$, it is mapped by a discriminant function $\varphi(\cdot) : R \to R$. For each feature $\phi_i$, we may obtain from ensembles of the two classes histograms as estimated distributions for the positive (intra-expression) and negative (extra-expression) examples, $h_i^+(\phi_i(x))$ and $h_i^-(\phi_i(x))$. An appropriate choice of the discriminant function could be

$$\varphi_i(\cdot) = \frac{1}{2}\log\frac{h_i^+(\cdot)}{h_i^-(\cdot)} \tag{5-20}$$

Our goal is to learn the features $\phi_i$ reliably so as to construct a classifier that produces the lowest error rate with a minimum number of features. The classification function thus becomes

$$F(x) = sign\left[\sum_{i=1}^{k}\frac{1}{2}\log\frac{h_i^+(\phi_i(x))}{h_i^-(\phi_i(x))}\right] \tag{5-21}$$

with $sign(\cdot) \in \{-1,+1\}$ being the class indicator function.

The feature selection can be performed by maximizing the JS-divergence, as defined below.

$$\begin{aligned} D_{SJS} &= \sum\nolimits_{l=0}^{L-1} h_i^+(l)\log[h_i^+(l)] - \sum\nolimits_{l=0}^{L-1} h_i^+(l)\log[\frac{h_i^+(l)+h_i^-(l)}{2}] \\ &\quad + \sum\nolimits_{l=0}^{L-1} h_i^-(l)\log[h_i^-(l)] - \sum\nolimits_{l=0}^{L-1} h_i^-(l)\log[\frac{h_i^+(l)+h_i^-(l)}{2}] \\ &= 2\cdot H(\frac{h_i^+ + h_i^-}{2}) - H(h_i^+) - H(h_i^-) \end{aligned} \tag{5-22}$$

where $L$ is the number of bins and $H$ is the Shannon entropy function. The most discriminating and efficient feature

$$\phi_k^* = \arg\max_{\phi_i} JS(\phi_i) \quad (5\text{-}23)$$

is called JS feature. Thus the JSBoost learning algorithm is illustrated as Figure 5.12. And the example of the selected features is shown in Figure 5.13.

```
0. (Input)
        Training examples $\{(x_1, y_1),...,(x_N, y_N)\}$, where $N = a + b$; of which $a$ examples have $y_i = +1$ and
        $b$ examples have $y_i = -1$.
1. (Initialization)
        $w_i = 1/2a$ for those examples with $y_i = +1$ and $w_i = 1/2b$ for those examples with $y_i = -1$.
2. (Forward Inclusion)
        For $k = 1,...,K$
    (1) Select JS feature $\phi_k$ by JS divergence with weight $w_i$ via Equ. 5-23. Let $f_k(x) = \frac{1}{2}\log\frac{h_k^+(\phi_k(x))}{h_k^-(\phi_k(x))}$;
    (2) Update weight $w_i \leftarrow w_i \cdot \exp(-\beta_k \cdot y_i f_k(x_i))$, $i = 1,...,N$, where $\beta_k = \log((1-\varepsilon_k)/\varepsilon_k)$ and $\varepsilon_k$
        is the training error of the current classifier. Normalize weights so that $\sum_i w_i = 1$.
3. (Output)
        $F(x) = \sum_{k=1}^{K} f_k(x)$
```

Figure 5.12. Jensen-Shannon boosting learning algorithm.

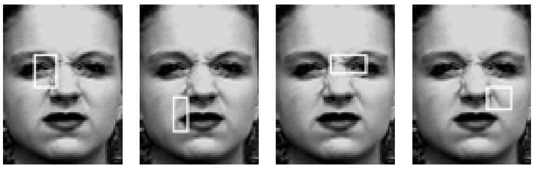


Figure 5.13. The first four sub-window from which the LBP features are obtained.

In the work of this thesis, we analyze the facial expressions in a generalized expression manifold based on a modified SLPP scheme, in which expressions of different subjects and different intensities are well aligned in a person-independent facial expression space. BoostLBP filters as well as raw image data and original LBP are applied in the feature extraction stage for SLPP training. Experiments show that JSBoost learning dose contribute to the facial expression recognition as expected.

### 5.3.2 Facial expression synthesis

The other component of the expressional manipulation system is realistic facial expression synthesis. There has been much research in this area, and expression mapping had become a popular method for generating facial animations. As pointed out in (Zhang et al., 2006), this method is a kind of warping-based approaches, which requires accurate labeling of feature positions of a subject's neutral face and another face of the same person with target expression. Because it considers shape changes only, the texture variations on the face are ignored, consequently it does not generate expression details such as wrinkles due to skin deformations. An alternative approach uses a large amount of sample views and applies morphing between them. The drawback of this method is that it is difficult to generate expressions for a new person who is not included in the training set.

Wang and Ahuja proposed an approach for facial expression decomposition with Higher-Order Singular Value Decomposition (HOSVD) that can model the mapping between persons and expressions, used for facial expression synthesis for a new person (Wang and Ahuja, 2003). One problem is that the global linearity assumption of expression variations introduces some artifacts and blurring while synthesizing expressions for a new person who is not in the training set. Du and Lin used PCA and linear mapping based on relative parameters as emotional function (Du and Lin, 2002). They encountered the similar problem as using HOSVD that large amount of training samples are demanded to well represent the variations of expressions for different subjects. Kouzani reported a Quadtree PCA (QPCA) to implement a global-local decomposition for approximating face images using a limited set of examples (Kouzani, 1999). Computation complexity is certainly increased by using QPCA, and the results do not look very good for human observation. Zhang *et al.* developed a geometry-driven facial expression synthesis system (Zhang et al., 2006). They subdivide the face into a number of subregions in order to deal with the limited space of all possible convex combinations of expression examples. The synthesis results look realistic and desirable. However, the blending along the subregion boundaries requires further efforts to avoid image discontinuities, and the registration of the large amount of feature points is a challenging task. Though it can be expanded to generate expressions for a new person, the system presented is person-specific.

Generally, a system that is intended to design facial expression synthesis should be capable to fulfill the following tasks. First, it is required to obtain realistic visual effects rather than only generating cartoon-like animations. Secondly, the system must be able to synthesize facial appearance for a new person, not limited to particular subjects within the training set. Finally, an efficient method is needed to synthesize arbitrary facial expressions with any desired intensities. The last task requires that facial expression synthesis and recognition should be performed under a unified framework with expression intensity alignment.

Let $I_P$ represent a face image, and $I_E$ be an expression image of this face. The procedure of expression synthesis is equivalent to setting up a mapping relation $M$ between a face and its expression, $I_E = M(I_P)$, where $M$ is supposed to be a complex nonlinear mapping.

In our work, a local geometry preserving based nonlinear method is proposed to approximate the mapping function $M$, which is inspired by LLE. It is assumed that small image patches in the face image and the expression image form manifold with similar local geometry in two different image spaces, and expression synthesis can be performed by giving training face-expression pair samples based on local nearest neighbors reconstruction.

Facial expressions of a new person can be synthesized under the assumption that similar persons have similar expression appearance and shape. However, all PCA based methods further assume that expression synthesis can be approximated by a linear combination of training face-expression pair samples. Due to the complexity of face structure, adopting this globally-linear assumption is not accurate when training samples are limited or there are big shape deformations of expressions.

Promising manifold learning methods such as LLE provide hints on this problem. The principle of LLE is to compute neighbor-preserving mapping between an original high-dimensional data space and a low-dimensional feature space, based on the simple geometric intuition that each data point and its neighbors lie on or close to a locally linear patch of the manifold. It is reasonable to adopt a local geometry preserving scheme to compute the mapping between the original face image space and the expression image space. To solve the problem of limited samples and deformable expression structure, a patch-based strategy is applied as in (Chang et al., 2004; Liu et al., 2005; Chen et al., 2001)

The proposed approach gets better performance than PCA-based method as reported in Publication 7. The reason is that the "double locality preserving" scheme - both locality with image patches in the spatial domain and locality with geometrical structure of manifold - is capable to approximate the global-nonlinear structure more efficiently.

## 5.4 Chapter summary

In this chapter we reviewed some key work of face detection, facial feature extraction, face caricature, facial expression analysis and synthesis. Face detection and feature location is a preliminary step for further processing, which is not as easy when there are illumination changes and face pose variations. In spite of the approaches or their combinations utilized to detect a face and facial features, heuristics such as the geometrical structure, texture properties of the feature regions, and symmetrical measurement, are always helpful and can be applied to speed up the detection process. An efficient framework for face detection and facial feature extraction, as applied in this thesis, could be: first, to locate some face candidates with a fast search using effective heuristic features; secondly use a statistical model to verify the candidates in which boost learning can be applied to build up a stronger classifier by combining a series of "weak" classifiers; then locate the accurate positions of the pair of eyes, basically with template matching or boosting method, for initialization of ASM search; and finally locate other features with a set of landmark points. Our contribution was to first introduce a multi-scale horizontal-component-based search algorithm in the coarse step of face detection, which can save up to 95% searching time comparing with pure sliding-window search based on template matching (including Adaboost binary classification). We also made some improvement for ASM searching scheme by introducing a 2D local search instead of 1D search to get better result for high-curvature feature points. Another piece is that we used a PCA based algorithm for "weak classifier" construction in boost learning. Because the "weak learner" is easy to be implemented and already not so weak, the number of features used for the final classifier could be reduced.

Face caricature and facial sketch is a comparatively emerging research field which tries to fill the gap between computer's processing and human perception of art. A few attempts have been proposed to synthesize facial caricatures either interactively or automatically. However, existing work did not cover the important characteristic of caricatures that exaggeration of a face or some specific features need to be introduced in order to produce impressional effects. We presented a parametric morphing algorithm to mimic the facial exaggeration effect. Some special processing was used to generate painting style like watercolor. The proposed method is effective for interaction based on mobile devices. Further efforts are needed to emphasize the capability of creating artistical drawings in an automatic manner.

Understanding and making use of facial expressions will generate differentiations for a lot of HCI applications. One of the most known tools is Facial Action Coding System (FACS) (Ekman and Friesen, 1978), which analysis facial expressions in terms of Action Units (AUs). FACS provides a linguistic description of all possible, visually detectable facial changes, however, to achieve a full-automatic FACS is still very difficult. Many existing work classified facial expressions into some prototypic expressions, e.g., emotional expressions. The main limitation of this approach is that pure emotional expressions are seldom elicited. Most of the time, people show blends of emotional expressions. Facial expression synthesis intends to generate realistic visual effects, and should be able to deal with a new person, not limited in the particular subjects. It is also desired that the synthesis algorithm can produce expressions with any intensity in a well controlled manner, and can work with the recognition part in a generalized framework. The surveyed systems encountered difficulties to fulfill these requirements at the same time. In publication P7, we proposed a generalized manifold to recognize and synthesize facial expressions, comparison with state of the art was presented. Although we used prototypic expressions, blends of emotional expressions can be analyzed and synthesized, and an affective interaction based on this can be conducted.

In this chapter we also briefly described our solutions regarding the above research questions. Some experimental results which were not presented in the attached publications are also included.

# Chapter 6

# Summary of Publications

This chapter provides a summary of the seven publications that form this thesis. All the publications deal with object extraction and recognition tasks for camera-based interaction on mobile platform, and the hybrid framework of heuristic and statistical approaches provides a principle in cooperation with contextual hypothesis. These publications can be broadly classified into three main categories according to their objectives. The first category comprising Publications 1 and 2 deals with the problem of extracting text or characters from natural scene images. The second category with Publications 3 and 4 is aimed at reading 1D and 2D bar codes by using camera phones. The last group comprises Publications 5-7 focus on human faces that algorithms of facial feature extraction and manipulation are proposed, facial expression recognition and synthesis are also explained and applied to a novel affective interaction.

The work has been carried out in Nokia Research Center, Beijing, during the years 2001-2007, and is a joint effort of researchers in the Visual Interaction System group who have been involved in the related projects. However, the contribution of the author to all these publications has been essential. The detailed contribution of the author to each of the seven publications is explained as following.

## 6.1 Publication 1: Text Location in Color Scene Images for Information Acquisition by Mobile Terminals

In publication one we considered the problem of detecting text from color scene images with complex background. A multi-group decomposition method that combines hue space, weak color, gray-level, and edge information is applied to disintegrate the original color image into different layers. A heuristic color clustering method in the hue space is proposed, which is simple but effective for real applications on mobile devices. A connected-component-based approach using Block Adjacency Graph (BAG) algorithm is employed to group and locate text regions in each image layer, and final result is obtained by grouping the regions from different layers.

For this work the author proposed the idea, planned and conducted all the detailed analysis and experiments. The author also wrote the publication with suggestions from Dr. Jari Kangas.

## 6.2 Publication 2: Character-Like Region Verification for Extracting Text in Scene Images

In publication two we further developed the approach of character-like region verification under the framework of multi-group decomposition. Alignment analysis is introduced to check the candidate regions in each binary image layer. Special concerns on the partition and combination of radicals of Chinese characters are proposed, and a priority adaptive segmentation method is applied to binarize the characters from their background for better recognition performance. Some heuristics such as texture statistics, recognition confidence, and align value are employed to further verify the truth of the final candidates. The algorithms are proved to be robust for a wide range of character fonts, illumination variations, and backgrounds.

The author was solely responsible for the original idea, conducted all the implementation and experiments, and wrote the publication with the help of Dr. Jari Kangas.

## 6.3 Publication 3: 1D Bar Code Reading on Camera Phones

In publication three we presented the research work on the bar code reading algorithms using a VGA camera phone. A widely used 1D bar code standard, EAN-13, is taken as an example to show the efficiency of the proposed method. A wavelet-based bar code location and knowledge-based bar code segmentation scheme is applied to extract bar code characters from poor-quality input images. In order to train an efficient recognition engine, a modified Generalized Learning Vector Quantization (GLVQ) method is designed for optimizing the feature extraction matrix and the class reference vectors.

The contribution of the author in this work was to suggest the idea of applying statistical training for bar code character recognition, and generate the training data, implement the software framework and run all tests. The bar code location and segmentation were done in cooperation with Dr. Kongqiao Wang, and the statistical training was performed in cooperation with Dr. Yanming Zou. The manuscript was co-authored with the above two also.

## 6.4 Publication 4: 2D Bar Codes Reading: Solutions for Camera Phones

In publication four we addressed the problem of specific 2D bar code design for mobile phones and introduced a low-level encoding method of matrix codes. At the same time, we proposed an efficient scheme for 2D bar codes decoding, of which the effort is put on solutions of the difficulties introduced by low image quality that is very common in bar code images taken by a phone camera. Heuristics are applied in various stages including code size indication, bar code location, deblurring modeling, and decoding. Grid vibration using statistical inference is also employed to get more robust results. A patch-based statistical framework is proposed for future research.

The design of the new matrix code and the bar code location were carried out in cooperation with Dr. Yanming Zou. The author proposed and implemented the decoding algorithms and conducted the experiments. The software implementation on camera phone was cooperated with Dr. Zou. The author also wrote the manuscript.

## 6.5 Publication 5: Facial Feature Extraction and Image-Based Face Drawing

In publication five we presented algorithms of facial feature extraction and based on the features extracted, an image-based drawing system was developed which automatically generates a drawing with embodiment of the individual face. Face detection is performed using horizontal texture information and subspace method based on eigenfaces. Then the pair of irises is located using Hough-transform and intensity valley correlation. Other facial features are detected based on the skeleton characteristics of the face. Finally a series of heuristics are employed to generate the stylized face drawings.

For this work the author proposed the idea, conducted all the planning and detailed implementations. The author wrote the publication under discussions with Dr. Kongqiao Wang.

## 6.6 Publication 6: Image-Based Face Drawing Using Active Shape Models and Parametric Morphing

In publication six the face drawing work was further developed in two aspects: a Multi-Resolution Active Shape Model (MRASM) is applied to extract facial features in a more effective manner; and a parametric morphing method is carried out to generate several expressions that embody a few basic emotions such as happiness, sadness, anger, and surprise.

The author was solely responsible for the original idea, conducted all the implementations and experiments, and wrote the publication.

## 6.7 Publication 7: Affective Interaction Based on Person-Independent Facial Expression Space

In publication seven we proposed a Person-Independent Facial Expression Space (PIFES) to analyze and synthesize facial expressions based on Supervised Locality Preserving Projections (SLPP), which aligns different subjects and different intensities of facial expressions on one generalized expression manifold. On PIFES, interactive curves of different patterns are generated according to the input facial expression image sequence, thus target responsive expression images are synthesized for different emotions. In order to generate arbitrary expressions for a new person with natural details, we also proposed a novel approach based on local geometry preserving between the input face image and the target expression image. Based on PIFES, a system is presented to realize an affective interaction between human and computer via automatic analysis of input facial expressions of the user and synthesis of the computer's facial expressions.

The contribution of the author in this work was to propose the original idea, conduct detailed design of algorithms, and implement all the software and experiments. The author also wrote the manuscript with suggestions from Dr. Kongqiao Wang.

# Chapter 7

# Conclusions

Interaction solutions for mobile devices with multi-modalities have attracted more and more attentions, in which visual channels contribute a lot in obtaining information from circumstances. In this thesis, we have focused on the study of visual object extraction and recognition from images captured by mobile imaging devices, which is one of the key components of camera-based interaction for mobile usages.

In many computer vision problems, theories that solutions are based on usually have very strong simplifications of reality, due to the highly complex dependence between an image and the objects that it represents. This means that performance of a computer vision system depends very much on many details of the implementation of the solution. We aim to find universal solutions for a set of problems, but it is crucial to be kept in mind that many implementation details and tricks are quite as important as the basic principles and models.

This thesis proposed a general solution that combines heuristic and statistical approaches in an empirical manner to achieve reliable performance for object extraction and recognition tasks. Applications of text detection from scene images, bar code reading with camera phones, and face image processings are manifested with detailed implementations, which could also be taken as examples for validating the hybrid approach.

Use of prior knowledge can have obvious effects on the application-oriented problems, which simplifies the solutions with specific purposes. For example, connected-component-based approaches have many advantages in terms of computational simplification, location accuracy, and easy of implementation, for text detection from complex background. Another instance is bar code location and segmentation that directly benefits from the dominant textural structure of parallel bars and spaces. Automatic face detection and facial feature extraction also make use of the knowledge-based methods to perform fast search. However, only use heuristics in such an intuitive way can not prevent the occurrence of extremely bad cases that exceptions lead to arbitrary crash. In another word, some extra actions are needed to guarantee that the system works beyond an acceptable performance with minimal risks.

Consequently we introduced statistical approaches and made a joint effort on the rational combination of the heuristics and statistical elements, both for the design of feature extractors and classifiers. Boosting learning algorithm is proved to be useful for combining a set of weak learners to achieve final strong classification capability. We have applied AdaBoost and JSBoost algorithms in text region verification, face detection and feature selection for facial expression analysis. Another statistical method for object shape interpretation, Active Shape Model (ASM), is employed in the facial feature extraction stage, where some improvement in searching scheme is performed to get better result for high-curvature feature points. Generalized Learning Vector Quantization (GLVQ) algorithm is used in bar code reading, which is aimed to deal with the blurring caused by out-of-focus capture. Regarding facial expression analysis and synthesis, we devised algorithms inspired by the locality preserving concept introduced by Locally Linear Embedding (LLE) and Locality Preserving Projections (LPP). A generalized expression manifold is formed by utilizing a Supervised LPP (SLPP) scheme, and expression recognition, intensity alignment, and expression synthesis are performed under the same framework. An affective interaction based on the framework is also proposed in Publication 7.

The nature of object extraction and recognition from images obtained by camera phones can not be easily modeled by uniform solutions. As presented at the beginning, the solutions are very much application-driven. The only principle we learned from the concrete implementations is that it is pretty efficient to apply heuristic but simple approaches at first to reduce the complexity of the problems and search for possible solutions, then use developed statistical learning approaches to deal with the remaining difficult but well-defined problems and get much better accuracy. The process can be evolved in some or all of the stages, and the combination of the approaches is problem-dependent.

Finally it should be kept in mind what we have presented several times: for concrete problems triggered by computer vision applications, implementation details and tricks are quite as important as the basic principles and modeling framework.

# References

Ando, S., and Hontani, H. (2001). Automatic visual searching and reading of barcodes in 3-D scene, *IVEC*, pp. 49-54.

Bala, J., DeJong, K., Huang, J., Vafaie, H., and Wechsler, H. (1996). Visual routine for eye detection using hybrid genetic architectures, *ICPR*, pp. 606-610.

Beletski, O., and Knorring, N. (2004). Requirements of visual code, *technical report, Nokia Research Center*.

Bunke, S.H., and Kronenberg, H. (1999). Identification of Text on Colored Book and Journal Covers, *Proc. of the Fifth ICDAR.*, pp. 57-62.

Canny, J. (1986). A computational approach to edge detection, *IEEE Trans. on PAMI*, Vol. 8, No. 6, pp. 679-698.

Casey, R.G., and Lecolinet, E. (1996). A Survey of Methods and Strategies in Character Segmentation, *IEEE Trans. on PAMI.*, Vol. 18, pp. 690 – 706.

Chandrasiri, N.P., Naemura, T., and Harashima, H. (2004). Interactive Analysis and Synthesis of Facial Expressions based on Personal Facial Expression Space, *FGR*.

Chang, H., Yeung, D.Y., and Xiong, Y. (2004). Super-resolution through neighbor embedding, *CVPR*.

Chang, Y., Hu, C., and Turk, M. (2003). Manifold of Facial Expression, *Int. Workshop on Analysis and Modeling of Faces and Gestures*.

Chen, H., Xu, Y., Shum, H.Y., Zhu, S., and Zheng, N. (2001). Example-based facial sketch generation with non-parametric sampling, *ICCV*.

Chen, T. F., and Vese, L.A. (2001). Active contours without edges, *IEEE Trans. on image processing*, Vol. 10 No. 2, pp. 266-277.

Chen, W.Y., and Chen, S.Y. (1998). Adaptive page segmentation for color technical journal's cover images, *Image and Vision Computing*, 16: pp. 855-877.

Chen, X., and Yuille, A.L. (2004). Detecting and Reading Text in Natural Scenes. *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 366-373.

Cheng, J. Liu, Q., Lu, H., and Chen, Y. (2005). Supervised kernel locality preserving projections for face recognition, *Neurocomputing* 67, 443-449.

Choi, K.Y. and Takaya, K. (2001). Facial feature extraction from a video sequence using independent component analysis (ICA), *IEEE conf. Communications, Computers and signal Processing,* pp. 259-262.

Cootes, T.F., and Taylor, C.J. (2002). Comparing variations on the active appearance model algorithm, *Proc. British machine vision conf.*, vol. 2, pp. 837-846.

Cootes, T.F., and Taylor, C.J. (2001). Statistical Models of Appearance for Computer Vision, *Technical Report,* Univ. of Manchester.

Cootes, T.F., Taylor, C.J., Cooper, D.H., and Graham, J. (1995). Active Shape Models- Their Training and Application, *Computer Vision and Image Understanding,* Vol. 61, No.1, pp. 38-59.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., and Taylor, J.G. (2001). Emotion Recognition in Human-Computer Interaction, *IEEE Signal Processing Magazine,* pp.32-80.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets, *Commun., Pure Appl., Math*. Vol. 41, pp. 909-996.

Du, Y., and Lin, X. (2002). Mapping Emotional Status to Facial Expressions, *ICPR*.

Duda, R.O., Hart, P.E., and Stork, D.G. (2001). *Pattern Classification,* Second Edition.

Edwards, G. J. *et al*. (1998). Face Recognition Using Active Appearance Models, *ECCV,* Vol. 2, pp. 581-595.

Efros, A.A., and Leung, T. K. (2001). Texture synthesis by non-parametric sampling, *Proc. of Inter. Conf. on ICCV*, pp. 20-27.

Ekman, P. and Friesen, W.V. (1978). *Facial Action Coding System (FACS): Manual*. Palo Alto: Consulting Psychologists Press.

Freund Y., and Schapire, R.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, vol. 55, No. 1, pp. 119-139.

Fujisawa, H., Sako, H., Okada, Y., and Lee, S.W. (1999). Information capturing camera and developmental issues, *Proc. of the Fifth ICDAR.*, pp. 205-208.

Fujiwara, T., Koshimizu, H., Fujimura, K., Kihara, H., Noguchi, Y., and Ishikawa, N. (2001). 3D modeling system of human face and full 3D facial caricaturing, *Proc. of 7th Inter. Conf. on Virtual Systems and Multimedia*, pp. 625-633.

Fukumoto, T., Wakabayashi, T., Kimura, F., and Miyake, Y. (2000). Accuracy improvement of handwritten character recognition by GLVQ, *IWFHR VII*, pp. 271-280.

Gaudin, C.V., Normand, N., and Barba, D. (1993). A bar code location algorithm using a two-dimensional approach, *ICDAR*, pp. 45-48.

Hamarneh, G., and Gustavsson, T. (2000). Combining snakes and active shape models for segmenting the human left ventricle in echocardiographic images, *IEEE conf. computers in cardiology*, Vol. 27, pp. 115-118.

Hara, F., Tanaka, K., Kobayashi, H., and Tange, A. (1997). Automatic Feature Extraction of Facial Organs and Contour, *Proc .of IEEE International Workshop on Robot and Human Communication*, pp. 386-391.

He, X., and Niyogi, P. (2003). Locality Preserving Projections, *NIPS*.

Hidai, K. *et al.* (2000). Robust Face Detection Against Brightness Fluctuation and Size Variation, *Proc. of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1379-1384.

Hong, *et al*. (1998). Online Facial Expression Recognition Based on Personalized Galleries, *FGR*, pp. 354-359.

Howlett, R.J., Berthier, S., and Awcock, G.J. (1997). Determining the location of industrial bar codes using neural networks, *IEE IPA*, pp. 511-515.

Hu, C., Chang, Y., Feris, R., and Yurk M. (2004). Manifold based analysis of facial expression, *CVPRW on Face Processing in Video*.

Huang, C.L., and Huang, Y.M. (1997). Facial Expression Recognition Using Model-Based Feature Extraction and Action Parameters Classification, *J. Visual Comm. and Image Representation*, Vol. 8, No. 3, pp. 278-290.

Huang, W., and Mariani, R. (2000). Face detection and precise eyes location, *ICPR*, pp. 722-727.

Huang, X., Li, S.Z., and Wang, Y. (2005). Jensen-Shannon boosting learning for object recognition, *CVPR*.

Huang, Y., Lin, S., Li, S.Z., Lu, H. and Shum H.Y. (2004). Face alignment under variable illumination, *FGR*.

Jain, A.K., and Chen, Y. (1993). Bar code localization using texture analysis, *ICDAR*, pp. 41-44.

Jain, A.K. and Yu, B. (1998). Automatic Text Location in Images and Video Frames, *Pattern Recognition*, Vol. 31, No. 12, pp. 2055-2076.

Jeng, S., Liao, H.M., Han, C., Chern, M., and Liu, Y. (1998). Facial feature detection using geometrical face model: an efficient approach, *Pattern Recognition*, Vol. 31, pp. 273-282.

Joseph, E., and Pavlidis, T. (1991). Waveform recognition with application to bar codes, *ICSMC*, Vol. 1, pp. 129-134.

Joseph, E., and Pavlidis, T. (1993). Deblurring of Bilevel Waveforms, *IEEE Trans. on Image Processing*, Vol. 2, No. 2, pp. 223-235.

Joseph, E., and Pavlidis, T. (1994). Bar code waveform recognition using peak locations, *IEEE Trans. on PAMI*, Vol. 16, No. 6, pp. 630-640.

Joseph, E., Pavlidis, T., and Wang, Y.P. (1990). Fundamentals of bar code information theory, *IEEE COMPUTER*, Vol. 23, No. 4, pp. 74-86.

Jung, K., Kim, K.I., and Jain, A.K. (2004). Text Information Extraction in Images and Video: a Survey. *Pattern Recognition*, 37(5): pp. 977-997.

Kanade, T., Cohn, J.F., and Tian, Y. (2000). Comprehensive Database for Facial Expression Analysis, *FGR*.

Kass, M., Witkin, A. and Terzopoulos, D. (1988). Snake: Active contour models, *Int. J. computer vision*, pp. 321-331, 1988. (also appeared in *Proc. of First International Conf. on Computer Vision*, pp. 259-269, 1987)

Kato, H., and Tan, K.T. (2005). 2D barcodes for mobile phones, *IEE Mobility Conference*.

Kawaguchi, T., Hidaka, D., and Rizon, M. (2000). Detection of eyes from human faces by hough transform and separability filter, *ICIP*, pp. 49-52.

Kim, K.A. (2004). Facial feature extraction using PCA and wavelet multi-resolution images, *FGR*.

Kim, K.I., Jung, K., and Kim, J.H. (2003). Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm, *IEEE Trans. on PAMI*, Vol. 25, No. 12, pp. 1631-1639.

Kim, P.K. (1999). Automatic Text Location in Complex Color Images Using Local Color Quantization, *TENCON 99. Proc. of the IEEE Region 10 Conference,* Vol. 1, pp. 629–632.

Kobayashi, H. and Hara, F. (1997). Facial Interaction between Animated 3D Face Robot and Human Beings, *ICSMC*, pp. 3732-3737.

Kobayashi H., Suzuki, S., and Takahashi, H. (1999). Automatic Extraction of Facial Organs and Recognition of Facial Expressions, *Proc .of IEEE International Workshop on Robot and Human Interaction,* pp.165-170.

Koshimizu, H., Tominaga, M., Fujiwara, T., and Murakami, K. (1999). On KANSEI Facial Image Processing For Computerized Facial Caricaturing System PICASSO, *Proc. of IEEE International Conference on Systems, Man, and Cybernetics,* Vol. 6, pp. 294-299.

Kouzani, A.Z. (1999). Facial Expression Synthesis, *ICIP*.

Lam, K.M., and Li, Y.L. (1998). An efficient approach for facial feature detection, *ICSP*, pp. 1100-1103.

Lam, K.M., and Yan, H. (1996). An improved method for locating and extracting the eye in human face image, *ICPR,* pp. 411-415.

Lau, C.M., Cham, W.K., Tsui, H.T., and Ngan, K.N. (2001). An energy function for facial feature extraction, Proceedings of international symposium on intelligent multimedia, video and speech processing, pp. 348- 351.

Li, H., Doermann, D., and Kia, O. (2000). Automatic Text Detection and Tracking in Digital Video, *IEEE Trans. Image Processing,* Vol. 9, No. 1, pp. 147-156.

Lin, C., and Wu, J. (1999). Automatic facial feature extraction by genetic algorithms, *IEEE trans. on image processing*, Vol. 9, No. 6, pp. 834-845.

Link-a, http://www.export911.com/e911/coding/ean13.htm#xEAN13Sym

Link-b, http://www.bmva.ac.uk/bmvc/2002/papers/60/full_60.pdf

Lisetti, C.L., and Schiano, D.J. (2000). Automatic Facial Expression Interpretation: Where Human-Computer Interaction, Artificial Intelligence and Cognitive Science Interesect, *Pragmatics and Cognition,* Vol. 8(1), pp. 185-235.

Liu, Q., Jung, C., Kim, S., Moon, Y., and Kim, J.Y. (2006). Stroke Filter for Text Localization in Video Images. *Proceedings of 2006 International Conference on Image Processing*, pp. 1473-1476.

Liu, Q., Tang, X., Jin, H., Lu, H., and Ma, S. (2005). A nonlinear approach for face sketch synthesis and recognition, *CVPR.*

Lucas, S. M. Panaretos, A., Sosa, L., Tang, A., Wong, S., and Young, R. (2003). ICDAR 2003 robust reading comepetitions, *Procedings of ICDAR*, pp. 682-687.

Ma, Y., Ding, X.,Wang, Z., and Wang, N. (2004). Robust precise eye location under probabilistic framework, *FGR.*

Mak, M. W. (1994). A lip-tracking system based on morphological processing and block matching techniques, *Signal Processing: Image Communication*, Vol. 6, pp. 335-348.

Muniz, R., Junco, L., and Otero, A. (1999). A robust software barcode reader using the Hough transform, *ICIIS*, pp.313-319.

Nishino, J., Kamyama, T., Shira, H., Odaka, T., and Ogura, H. (1999). Linguistic Knowledge Acquisition System on Facial Caricature Drawing System, *IEEE International Fuzzy Systems Conference Proceedings,* Vol. 3, pp. 1591-1596.

Normand, N., and Gaudin, C.V. (1994). A two-dimensional bar code reader, *ICPR*, Vol. 3, pp. 201-203.

Ohya, J., Shio, A., and Akamatsu, S. (1994). Recognizing Characters in Scene Images, *IEEE Trans. on PAMI.*, Vol. 16, No. 2, pp. 214-220.

Ojala, T., and Pietikainen, M. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Trans*. *PAMI*, pp. 971-987, Vol. 24, No. 7.

Osuna, E., Freund, R., and Girosit, F. (1997). Training Support Vector Machines: an Application to Face Detection, *Proc. of IEEE Computer Society Conference on CVPR*, pp.130-136.

Ouaviani, E., Pavan, A., Bottazzi, M., Brunelli, E., Caselli, F., and Guerrero, M. (1999). A common image processing framework for 2D barcode reading, *IEE conference on IPIA*, pp. 652-655.

Pantic, M., and Rothkrantz, J.M. (2000a). Automatic Analysis of Facial Expressions: The State of the Art, *IEEE Trans. on PAMI*, Vol.22, pp. 1424-1445.

Pantic, M., and Rothkrantz, .J.M. (2000b). Expert System for Automatic Analysis of Facial Expressions, Image *and Vision Computing J.*, vol. 18, no. 11, pp. 881-905.

Pardas, M. (2000). Extraction and tracking of the eyelids, *ICASSP*, pp. 2357-2360.

Park, H.W, Schoepflin, T., and Kim, Y. (2001). Active Contour Model with Gradient Directional Information: Directional Snake, *IEEE Trans. on Circuit and Systems for Video Technology,* Vol. 11, No. 2, pp.252-256.

Pavlidis, T., Swartz, J., and Wang, Y.P. (1992). Information encoding with two-dimensional bar codes, *IEEE COMPUTER*, Vol. 25, No. 6, pp. 18-28.

Perez, C.A., Palma, A., Holzmann, C.A., and Pena, C. (2001). Face and eye tracking algorithm based on digital image processing, *ICSMC.*

Piccioli, G., Micheli, E.D., Parodi, P., and Campani, M. (1996). Robust Method for Road Sign Detection and Recognition, *Image and Vision Computing*, Vol. 14, pp. 209-233.

Prasad, N.M., and Sowmya A. (2004). Multi-class unsupervised classification with label correction of HRCT lung images, *Proceedings of International Conference on* Intelligent *Sensing and Information Processing*, pp. 51-56.

Ridder, D., Kouropteva, O., Okun, O., Pietikainen, M., and Duin, R. P. W. (2003). Supervised locally linear embedding, *Proc. of Artificial Neural Networks and Neural Information Processing, ICANN/ICONIP*, 2003.

Rizon, M., and Kawaguchi, T. (2000). Automatic eye detection using intensity and edge information, *TENCON* , Vol. II, pp. 415-420.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, Vol. 290, 2323-2326.

Rowley, H.A., Baluja, S., and Kanade, T. (1998). Neural Network-Based Face Detection, *IEEE Tran. on PAMI.*, Vol. 20, No. 1, pp.23–38.

Ryu, Y.S., and Oh, S.Y. (2001). Automatic extraction of eye and mouth fields from a face image using eigenfeatures and multilayer perceptions, *Pattern Recognition*, Vol. 34, pp. 2459-2466.

Shan, C., Gong, S., and McOwan, P. W. (2005a). Appearance Manifold of Facial Expression, *ICCV workshop on HCI.*

Shan, C., Gong, S., and McOwan, P.W. (2005b). Robust facial expression recognition using local binary patterns, *ICIP*.

Shan, C., Gong, S., and McOwan, P.W. (2006). A Comprehensive Empirical Study on Linear Subspace Methods for Facial Expression Analysis, *CVPRW*.

Shellhammer, S.J., Goren, D.P., and Pavlidis, T. (1999). Novel signal-processing techniques in barcode scanning, *IEEE Robotics & Automation Magazine*, Vol. 6, No. 1, pp. 57-65.

Shen, D. and Davatzikos, C. (2000). An adaptive-focus deformable model using statistical and geometric information, *IEEE Trans. on PAMI*, Vol. 22, No. 8, pp. 901-913.

Shen, L., Wang, Q., and Xing, X. (2000). Automatic Human Face Location ad Tracing in a Complex Background, *Chinese Journal of Electronics*, vol. 9, no. 1, pp. 65-69.

Smeraldi, F., and Bigun, J. (1998). Facial feature detection by saccadic exploration of the gabor decomposition, *ICIP*, pp. 163-167.

Suen, H.M., and Wang, J.F. (1997). Segmentation of Uniform-Coloured Text from Colour Graphics Background, *IEE Proc.-Vis. Image Signal Process.*, Vol. 144, No. 6, pp. 317-322.

Tang, X., and Wang, X. (2003). Face sketch synthesis and recognition, *ICCV*.

Tian, Y., Kanade, T., and Cohn, J.F. (2000). Dual-State Parametric Eye Tracking, . *Proc. of 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 110-115.

Tian, Y., Kanade, T., and Cohn, J.F. (2001). Recognizing Action Units for Facial Expression Analysis, *IEEE Trans. on PAMI*, Vol. 23, pp. 97-115.

Trimeche, M. (2003). Out of focus analysis & correction for document images, *technical report of Nokia Research Center*.

Tsapatsoulis, N. *et al*. (2000) Face Detection in Color Images and Video Sequences, *MeLeCon.*, vol. II, pp. 498-502.

Turin, W., and Boie, R.A. (1996). Minimum discrimination information bar code decoding , *19th convention of Electrical and Electronics Engineers in Israel*, pp. 255-258.

Turin, W., and Boie, R.A. (1998). Bar code recovery via EM algorithm, *IEEE Trans. on PAMI*, Vol. 46, No. 2, pp. 354-363.

Uusikyla, J. (2000). Barcode recognition in digital image, *Master of science thesis*, TUT.

Wang, H. (2001). Automatic character location and segmentation in color scene images, *Proceedings of 11th International Conference on Image Analysis and Processing*, pp. 2-7.

Wang, H. (2007). Facial Expression Synthesis and Recognition with Intensity Alignment, *SIGMAP*.

Wang, H., and Zou, Y. (2006). Camera Readable 2D Bar Codes Design and Decoding for Mobile Phones, *Proceedings of International Conference on Image Processing*, pp. 469-472.

Wang, H., and Ahuja, N. (2003). Facial expression decomposition, *ICCV*.

Wang, K.Q., Zou, Y.M., and Wang, H. (2005). Barcode reading from images captured by camera phones, *IEE Mobility Conference*, 2005.

Wang, Y., and Yuan, B. (2000). Segmentation Method for Face Detection in Complex Background, *Electronics Letters*, Vol. 36, No. 3, pp. 213-214.

Wong, K., Lam, K., and Siu, W. (2001). An efficient algorithm for human face detection and facial feature extraction under different conditions, *Pattern Recognition*, Vol. 34, pp. 1993-2004.

Xue, Z., Li, S.Z., and Teoh, E.K. (2001). Facial feature extraction and image warping using PCA based statistic model, *ICIP*, pp. 689-692.

Yamashita, T., Ichimura, T., Tazaki, E., and Takahashi, M. (2000). Fuzzy Reasoning Model of Facial Selection and Its Applications, *4th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, pp. 860-863.

Ye, X., and Cheriet, M. (2001). Stroke-model-based character extraction from gray-level document images, *IEEE Trans. on PAMI.*, Vol. 10, No. 8, pp. 1152 – 1161.

Yeasin, M. (2000). Detecting and tracking human face and eye using an space-varying sensor and an active vision head, *CVPR*.

Yeasin, M., Bullot, B., and Sharma, R. (2006). Recognition of Facial Expressions and Measurement of Levels of Interest From Video, *IEEE Trans. on Multimedia*, Vol. 8, pp. 500-508.

Yen, G.G., and Nithianandan, N. (2002). Facial feature extraction using genetic algorithm, *Congress on Evolutionary Computation.*, pp. 1895-1900.

Yoneyama, M. *et al*. (1997). Facial Expressions Recognition Using Discrete Hopfield Neural Networks, *ICIP*, Vol. 3, pp. 117-120.

Yu B., and Jain, A.K. (1996). A Generic System for Form Dropout, *IEEE Trans. on PAMI.*, Vol. 18, No. 11, pp. 1127-1134.

Yu, S.U., and Shi, J. (2003). Object-specific figure-ground segregation, *CVPR*, Vol. 2, pp. 39-45.

Zhang (1997). Prolog to Face Recognition: Eigenface, Elastic Matching, and Neural Nets, *Proc. of the IEEE*, Vol. 85, No. 9, pp. 1422-1435.

Zhang, L., and Lenders, P. (2000). Knowledge-based eye detection for human face recognition, *Inter. Conf. on knowledge-based intelligent engineering & allied technologies*, pp. 117-120.

Zhang, Q., Liu, Z., Guo, B., Terzopoulos, E., and Shum, H.Y. (2006). Geometry-Driven Photorealistic Facial Expression Synthesis, *IEEE Trans. on Visualization and Computer Graphics*, Vol.12, pp. 48-60.

Zhong, Y., Karu, K., and Jain, A.K. (1995). Locationg Text in Complex Color Images, *Pattern Recognition*, Vol. 28, No. 10, pp. 1523-1535.

Zhong, Y., Zhang, H., and Jain, A.K. (2000). Automatic Caption Localization in compressed Video, *IEEE Trans. on PAMI.,* Vol. 22, No. 4, pp. 385-392.

Zhou, J., and Lopresti, D. (1997). Extracting Text from WWW Images, *Proceedings of ICDAR.*, pp. 248-252.

Zoller, T., Hermes, L., and Buhmann, J.M. (2002). Combined color and texture segmentation by parametric distributional clustering, *ICPR*, Vol. 2, pp. 627-630.

ISBN 978-951-22-9137-3 (printed)
ISBN 978-951-22-9134-2 (PDF)
ISSN 1455-0474
Picaset Oy, Helsinki 2007