Quad-Tree based Image Encoding Methods for Data-Adaptive Visual Feature Learning

Cuicui Zhang

Abstract

Visual feature learning is a holy grail in computer vision for a long time. It aims at learning good feature representations that can help machine learning algorithms in making accurate predictions. It is still a challenging task because of the high dimensionality and variability of image data, which can take a variety of forms such as static images, video sequences, views of multiple cameras, etc.. This dissertation seeks to develop data-adaptive visual feature learning algorithms that can effectively and efficiently learn good, semantically meaningful features for a wide range of image data.

Recently, many sophisticated feature learning algorithms have been proposed, e.g. deep learning. State-of-the-art algorithms adopt an image block (patch) based multilayer framework to learn a hierarchical feature representation, where high-level features are formulated by the composition of low-level features. The feature hierarchy increases the abstraction and generalization of the feature representation, which make these methods be successful in solving many computer vision problems. However, they still have several unsolved issues. The first problem relates to the block based low-level feature extraction, which is straightforward and not adaptive enough to fit image statistics. The second problem relates to the image pyramid based hierarchical models, which are also straightforward and not flexible enough to investigate the image structure.

The above mentioned problems that prevent the better performance of state-of-the-art algorithms are mainly because of the lack of image structure. While existing feature learning algorithms implicitly investigate the image structure by exploring the correlationship between local subregions, this dissertation provides an alternative way by explicitly learning the image spatial and hierarchical structure by using Quad-Tree based image encoding methods. Then, we incorporate the Quad-Tree based image structure into existing feature learning algorithms to make the learning procedure data-adaptive and more powerful. The resulting benefits are two-fold: (1) Low-level features extracted from local subregions of

variant sizes are adaptive to image statistics; (2) High-level features can be formulated according to the Quad-Tree structure, which is data-adaptive and more flexible than image pyramid based methods.

Besides the above mentioned problems, this dissertation also considers some other challenging issues when developing feature learning algorithms, such as feature learning with small training data and feature learning in dynamic environments. While existing feature learning algorithms can work well when the training data is large enough, they suffer from the small sample size (SSS) problem under small training data. Moreover, although existing feature learning can be applied to static images relatively easily, it is not so for dynamic environments (e.g. video sequences taken by a moving camera). Under such case, we need to add more constraints to make feature learning more naturally. According to the four challenging feature learning situations, this dissertation develops four Quad-Tree based feature learning algorithms, including:

- Feature learning from data-adaptive blocks decomposed by Quad-Tree;
- Hierarchical feature learning using Quad-Tree structure of images ;
- Feature learning from enlarged training data encoded by Quad-Tree for Small Sample Size problem;
- Feature learning in dynamic environments using Helmholtz-Hodge decomposition and Quad-Tree;

Thanking to the data-adaptive property, our methods can be widely applied to many computer vision tasks. We use two representative applications for evaluation: face recognition and motion segmentation. Extensive experimental results on several benchmarks and the comparative analysis with state-of-the-arts have demonstrated the effectiveness and merits of the proposed algorithms.

Acknowledgment

First of all, I would like to express my sincere gratitude and appreciation to my supervisor, Prof. Takashi Matsuyama. His riotous attitudes to research, broad knowledge in wide areas, strong passion in scientific research, and patient guidance though my studies have always enlightened me in solving problems and conquering bottlenecks. I also appreciate that he gave me all the freedom and resources with a supportable team environment to develop my skills in doing research. Moreover, I learned a lot from him on how to become a woman of value, an independent researcher, and so on.

I also would like to express my sincere appreciation to my advisor Dr. Xuefeng Liang. He helped me a lot, not only in doing research, but also other aspects of academic life. Without his generous help, this thesis cannot be completed. I cannot thank him enough.

I also would like to thank my thesis committee very much, Professor Michihiko Minoh, for taking much time in reviewing my thesis and giving me many invaluable comments.

I am also thankful to all the faculty members in Matsuyama Lab, Dr. Shohei Nobuhara, Dr.Hiroaki Kawashima, Dr. Tony Tung, Dr. Takekazu Kato, Dr. Takeshi Takai, Dr. Takeshi Takai, Dr. Xinpeng Zhang, Dr. Saher Javaid, Dr. Rodrigo Verschae. They give me many helpful comments in research meeting and daily life. Special thanks to our secretaries, Ms. Kazuyo Hashimoto, Ms. Shiho Kimura, Ms. Aya Inoue, Ms. Mayumi Izumi, Ms. Asako Yoshimura. I also thank to all the other members of Matsuyama Laboratory, especially but not limited to Ms. Luyang, Mr. Qun Shi, Mr. Yonetani Ryo, Ms. Erina Ishikawa, Ms. Feng Chen, Mr. Yanghong Zhong, Ms. Jenifer Kuo, Ms. Wei Ning. I especially thank Jennifer very much. She brought me sunshine to my life. She is alike an angle in my life.

I also thank my previous supervisors and friends very much: Prof. Caiming Zhang (my supervisor during Master's Study), Prof. Richard Hartley (my supervisor during visiting NICTA in Australia), Prof. Keiichi Uchimura and

Prof. Koutaki Gou (my supervisor during visiting Kumamoto University), Prof. Naixue Xiong, Jie Zhao, Qian Yu, Ruoshu Bu, Yi Li, Yongmi Yang, Buyu Liu, Miaomiao Liu, Xuming Han, Zhilei Liu, Zhiyan Zhou, Xiuqian Li, Guangpeng Zhou, Lin Zhang, Shu Ma, Zhi Li, Peng Cheng, Bo Hou, Min Lv, Xiaojie Li, Qinghai Li, Jinguo Zhou, Mingguo Shao, Mingyun Lv, Shuai Han, Chongke Bi, Wenpeng Chen, Tianye Liu, Guangming Tang, Yun Sun, Xuzhou Li, Tingting Chen, Xiaoyao Pi, Xiaoyuan Liu, Pei Du, Fangqi Zhu, Zhe Xu, Wei Chen, Siyang Yu, Jiefeng, Yaochun Qiang, Lei Sun, Liang Zhao, Ling Wang, Bingwei Tian, Lei Lv, Yongqiang Cui, Zhaozheng Hu, Qigui Niu, Zhiqiang Zuo, Feng Li, Huan Yang, Zhe Liu, Jing Liu, Huimin Wen, Peipei Li, Wu Zhang, Wenhao Zhu, Wei Li, Yue Chen, Yuanfeng Zhou, Yu Wei, Dongmei Niu, Zhenzhou Niu, Xifeng Gao, Weitao Li, Kunpeng Wang, Chao Peng, Xia Hu, Lin Ma, et al.. Thank you very much for their help and support.

Finally, I thank my family (Father: Shouguo Zhang; Mother: Zhentian Pi; Uncle: Taitian Pi; Brother: Zhonghua Zhang; Sister: Ying Wang, Xiaoyao Pi) for their love and support. Without their support, this thesis would not have been possible. Especially, I thank my Uncle Kaming Ng., who has supported my life during the undergraduate study. He paid the most part of my tuition when I was a undergraduate student. He told me that "Once you have extra money in your pocket, please think about others who have not enough money for their life in universities". I have been keeping these words in my heart and trying to do something for others. During my Ph.D. study, I helped three undergraduate students by paying their tuition, one student per year. I felt very happy from doing this. It remains me a proverb, "The roses in her hand; the flavor in mine".

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Learning is constructing or modifying representations of what is being experienced." — McCarthy, 1968.

As a sub-field of artificial intelligence (AI), machine learning aims to develop algorithms that can learn from complex data and make accurate prediction of previously unknown data. It has been recognized as the central to the success of AI and has been widely applied to AI-level applications such as computer vision, statistical pattern recognition, natural language processing, medical imaging, and data mining, and so on.

However, the success of machine learning usually depends on a good representation of the data [Lee10]. Working with raw images is undesirable due to many reasons, including: (a) much redundant and irrelevant information may be contained in the high-dimensional data; (b) no invariant parameters to cope with external factors, such as scaling, translation and orientation distortions, illumination changes, etc.; (c) lack of organization in natural images, which makes it difficult to learn the hidden structure in image data. In order for ease use of machine learning algorithms, many research efforts haven be paid on learning good representations, which can capture the hidden structure in image data, deal with the curse of dimensionality and addressing several underlying factors. The process of learning such good feature representations refers to "visual feature learning", which is a fundamental task to many machine learning applications such as classification, recognition, and so on. While many state-of-the-art algorithms rely on hand-crafted features, this dissertation seeks to develop data-adaptive visual feature learning algorithms, which can effectively and efficiently learn good, semantically meaningful feature representations for a variety of image data.

1.1 Visual Feature Learning

Visual feature learning acts as a bridge between raw images and machine learning algorithms. It transforms the raw image data into a good representation and then transferred the feature representation into machine learning algorithms to make decisions. The role of visual feature learning in a machine learning system is illustrated in Figure 1.1. Good feature representations ensure the accuracy of machine learning algorithms in making decisions thanking to their good capabilities in: (1) capturing all or most of the relevant information; (2) eliminating all or most redundant information; (3) dealing with external factors and variations; (4) managing their sizes to fulfill the memory and speed requirements of learning algorithms. However, it is still a challenging problem since the image data is usually of high dimensional and can take a variety of forms such as static images, video sequences, views of multiple cameras, etc.. Visual feature learning has attracted many interests of scientist in both theoretical and engineering fields.

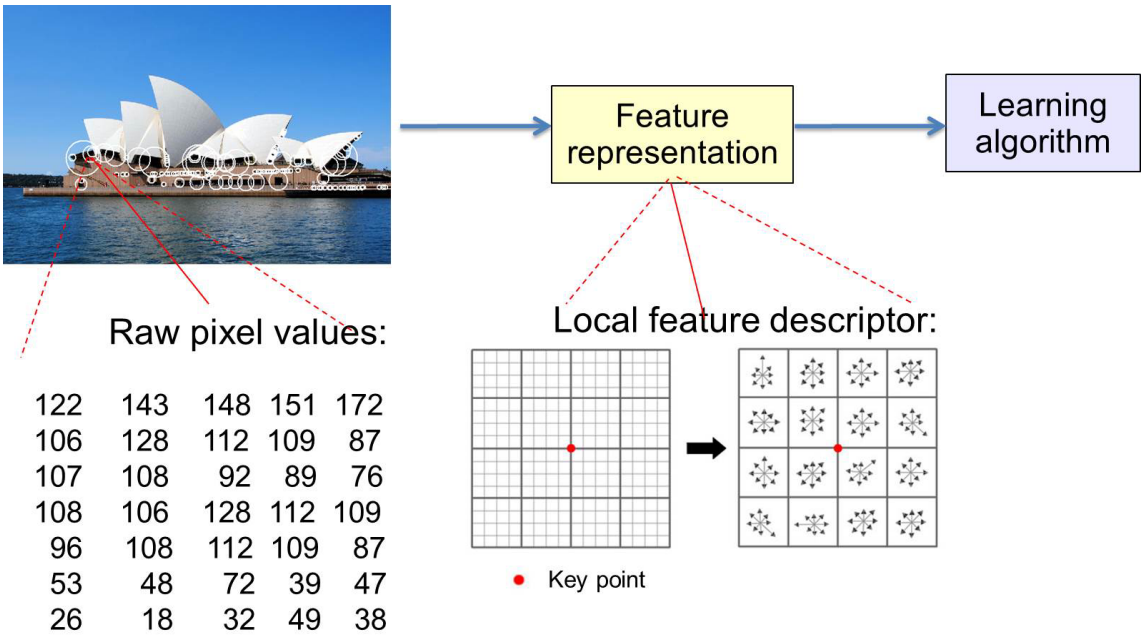


Figure 1.1: Illustration of the role of visual feature learning in a machine learning system, which acts as a bride between image data and learning algorithms.

Many visual feature learning algorithms have been proposed in the literature. Features can be extracted from either the whole image region or local subregions. Existing visual feature learning algorithms can be briefly divided into holistic and

local based methods, where the holistic based methods extract a single feature vector from the whole image region and the local based methods extract a series of feature vectors (bag-of-features) from a number of local subregions. Holistic methods can preserve the global shape and overall structure of the image data, but they have weak ability in dealing with local deformations. On the contrary, local methods have higher capability in coping with local deformations, however, they can not preserve the global information of the image. Recently, many research findings [FBL09, JDaHW14] suggest that the global feature formed based on local feature analysis is preferred over bag-of-features. Many sophisticated feature learning algorithms have been proposed for global feature generation. State-of-art algorithms adopt a patch based, multilayer framework to learn a hierarchical feature representation, where high-level features are formulated by the composition of low-level features according to a deep architecture. These algorithms refer to the *"hierarchical feature learning"* (shorten by "hierarchical learning") [JDaHW14].

A typical pipeline of a hierarchical learning algorithm comprises two procedures: (1) low-level feature extraction; and (2)high-level feature integration. A hierarchical learning framework is shown in Figure. 1.2. The success of hierarchical learning comes from the capability of learning feature hierarchies, which has strong representational power in encoding image variability and capturing the relationships in the data. It increases the abstraction and generalization of the feature representation and makes it more robust to deal with external factors. Hierarchical learning has achieved impressive results in a wide spectrum of computer vision applications, such as object detection [GDDM14], object categorization [FBL09], image segmentation [FCNL13, CFCNL14], image recognition [MS12], robotics ([HES+08]), information retrieval ([SMH07]), emotion understanding [KPB+13], etc..

1.2 Motivation

Although hierarchical learning algorithms have achieved many impressive results, just as the *No Free Lunch* theorem states that no technique can be optimal for all data, hierarchical learning also has some unsolved issues.

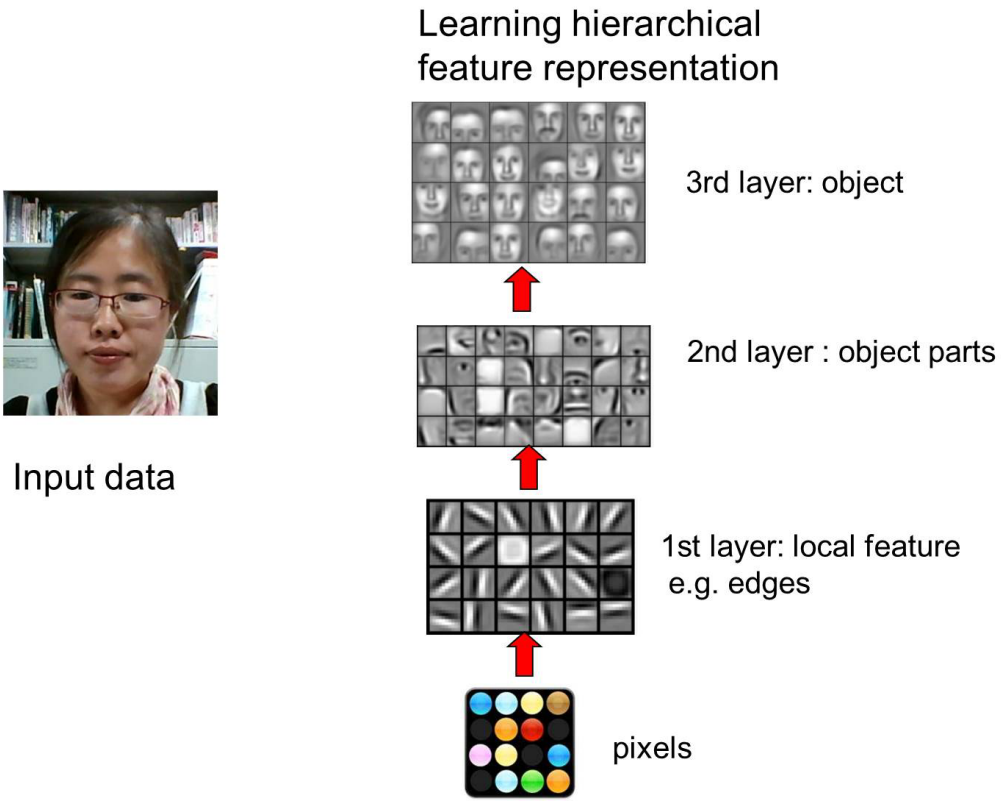


Figure 1.2: Illustration of a hierarchical learning framework, where higher-level features are formed by the composition of lower-level features according to a deep architecture. The right side of the figure appears at the tutorial of ECCV 2010 (http://ufldl.stanford.edu/eccv10-tutorial/), authored by Andrew Ng.

1.2.1 Challenging Issues

According to the two stages of a typical hierarchical learning pipeline: low-level feature extraction and high-level feature integration, one can conceive the central challenges in training hierarchical learning architectures are as follows:

- How to extract lower-level features in order to provide adequate input to higher-level features;
- How to adjust higher-level features to make good use of lower-level features.

As to the first problem, existing hierarchical learning algorithms usually employ image block based low-level feature extraction, where the image region is partitioned into blocks (patches) of the same size and low-level features are extract from each block with the same weight. They believe that the image fea-

ture distribution can be well exploited through block-based feature encoding and transformation. However, the block based method is straightforward and not adaptive enough to fit the image statistics. The feature distribution of natural images under complex learning situations can not be well investigated. To solve this problem, several state-of-the-art machine learning systems utilize hand-crafted features. However, the hand-crafted features rely on a huge amount of human efforts. And the procedure of generating hand-crafted features is time consuming, domain-specific and incomplete.

For the second problem, a number of deep architectures have been proposed for high-level feature integration. Early works utilize fully-connected neural networks, where the parameters of low-level features are learned based on backpropgation [RGEW86]. Since the backpropgation is very likely to get stuck in local minima especially when the parameters of low-level features are randomly initialized, hierarchical learning does not become successful until the development of two technologies recently: (1) local-connectivity based hierarchical architectures and (2) unsupervised pre-training for parameter learning. Although these two techniques have brought many impressive results, they still have some unsolved problems. The first problem relates to the local-connectivity based framework, which is defined using window-based image spatial pyramid. The image spatial pyramid is straightforward and not flexible enough to investigate the image structure. The second problem relates to the layer-wise unsupervised pretraining, which trains the parameters of low-level features layer-by-layer. This strategy can not train all layers at a time with respect to the original image input. It discards the image input after the first layer and makes be upper have have a loose connection with the original image input.

Another potential disadvantage with existing visual feature learning algorithms is that they require a large number of training samples for supervised learning applications such as recognition/classification. However, this can not meet in some practical applications such as person re-identification in multi-camera networks, surveillance photo identification, etc., where the training data is not enough due to the difficulty in data collection. The small training data leads to the small sample size (SSS) problem arising from the small number of training samples compared to the large dimensionality of the feature space. An extreme case of SSS is single sample per person (SSPP), where only one sample is available for each subject. Existing visual feature learning algorithms suffer from overfitting problem under SSS severely. Even worse, SSPP makes some feature

learning methods failure, e.g. linear discriminant analysis (LDA) since we can not generate the within-class scatter matrix under SSPP.

While existing visual feature learning algorithms can be applied to static images relatively easily, it is not so video sequences especially when the video is captured by a moving camera. . The video based feature learning is still a very challenging problem, which is usually with a large amount of video data and suffer from several problems due to the dynamic background.

The above mentioned issues still challenge the existing visual feature learning algorithms and prevent the better performance of state-of-the-arts. In this dissertation, we aim at developing data-adaptive and more powerful visual feature learning algorithms which can solve these challenging issues effectively.

1.2.2 The Core Aspect of This dissertation

A major factor that prevents the better performance of state-of-the-art visual feature learning algorithms might have been the lack of the image structure. Recently, many literature findings suggest the importance of learning the explicit image structures, which gives rise to relevant spatial layout properties of images and provides a perceptual hierarchical view on the correlationship between local subregions. In this dissertation, we investigate the image structure by using Quad-Tree based image encoding methods and incorporate it with existing visual feature learning algorithms to make the learning procedure data-adaptive and more powerful. Understanding the image structure can contribute to the learning of the semantics and the global structure of images. Instead of arbitrarily defining block based local region partition and image spatial pyramid, we use Quad-Tree to explicitly learn data-adaptive local subregions for low-level feature extraction and data-adaptive tree structure for high-level feature integration. While hierarchical learning implicitly exploits the image structure by modeling the correlationship between local subregions, Quad-Tree explicitly learns the hierarchical image structure by recursively decomposing an image region into local subregions of variant sizes. The benefits are three-fold: (1) the data-adaptive block based image region partition makes the low-level feature extraction be region specific and locally adapt to the image statistics; (2) the local-connectivity of subregions is defined according to the data-adaptive tree structure, which is a more sophisticated strategy than image spatial pyramid; (3) the Quad-Tree structure pre-trains the weights of local-subregions making the composition of low-level features to

high-level features more effectively and accurately. In contrast to the layer-by-layer unsupervised learning strategy, the tree structure provides a direct path of upper layers to the image input, allowing each layer to be trained respected to the image input. To solve the SSS problem, our algorithm first augments the small data by generating new samples and then perform ensemble learning in the enlarged training data more effectively. Finally, to extend the Quad-Tree based visual feature learning from static images to video sequences, we develop a motion segmentation algorithm, which performs Helmholtz-Hodge decomposition first for camera motion compensation and then perform Quad-Tree partition for object motion segmentation.

The remainder of this chapter is as follows: The proposed algorithms towards four feature learning situations are briefly described in Section 1.3. Section 1.4 summarizes the main contributions of this dissertation. The organization of this dissertation is presented in Section 1.5.

1.3 Proposed Algorithms

Towards the aforementioned four feature learning situations, we develop four Quad-Tree based image encoding methods for data-adaptive visual feature learning, including: (1) Feature learning from data-adaptive blocks decomposed by Quad-Tree; (2) Feature learning using Quad-Tree structure of images; (3) Feature learning from enlarged training data encoded by Quad-Tree for Small Sample Size problem; (4) Feature learning in dynamic environments using Helmholtz-Hodge Decomposition and Quad-Tree. The first three algorithms are mainly on feature learning from static image dataset while the last algorithm is on object motion segmentation and recovery in moving camera videos. In this dissertation, we use face recognition as research background for the problem definition and algorithm description of the first three algorithms while motion segmentation as the research background for the last algorithm. "Data" here has different meaning in different algorithms. For the first three algorithms 'data' refers to image dataset, while for the last algorithm data means "video sequence".

1.3.1 Quad-Tree Structure

Quad-Tree is an effective and commonly used hierarchical data structure that recursively subdivides an image region into four quadrants according to some cri-

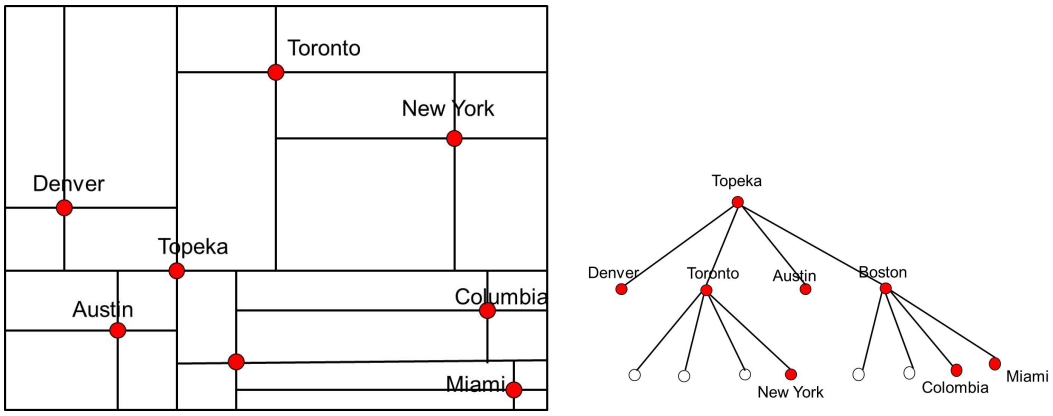


Figure 1.3: An example of Point Quad-Tree.

terion of image homogeneity. If a region does not meet the homogeneity criterion (e.g., its variance is higher than a threshold variance), it will be subdivided into four subregions of the same size. Iteratively, we can build a tree structure according to split process. The tree structure contains both leaf nodes and non-leaf nodes (denoted by homogeneous and heterogeneous subregions respectively). Statistics of different nodes at different levels provide the spatial structure of data presentation.

Quad-Tree has been widely applied in many image based applications, such as image representation and retrieval ([RMM11]), image annotation ([CMES10]), image recognition ([ZLM12, ZLM13]), video coding ([MSB$^+$10]). Besides computer vision, Quad-Tree can be used in many other research fields, such as computer graphics ([Ase07]), Geographical Information Systems (GIS) ([KPS$^+$04]), etc..

Quad-Trees can be divided into two types: Point Quad-Trees and Trie based Quad-Trees (also named as Point-Region Quad-Trees or Region Quad-Trees for short) [Sam06]. The Point Quad-Tree is an adaptation of a binary tree, which can be used to represent two-dimensional point data. The center of a subdivision is always on a point. The Trie based Quad-Tree is a more often used Quad-Tree than Point Quad-Trees. It is developed for region representations based on the successive subdivision of the image region into four equal-sized quadrants. Examples of Point Quad-Tree and Trie based Quad-Tree are shown in Figure 1.3 and Figure 1.4, respectively. Since this dissertation focuses on feature learning from the image region, We use the Trie based Quad-Tree for image encoding.

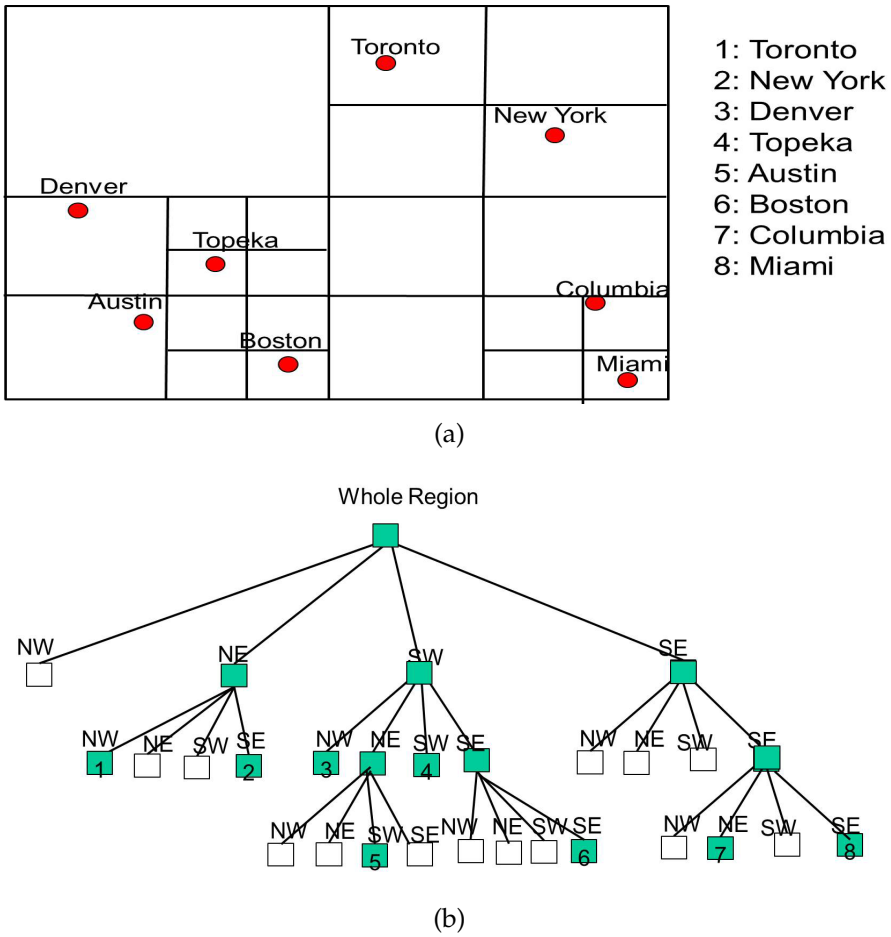


Figure 1.4: An example of Trie-based Quad-Tree.

1.3.2 Quad-Tree based Visual Feature Learning Algorithms

The developed four Quad-Tree based visual feature learning algorithms towards four challenging issues are briefly described as follows:

Algorithm 1: Feature learning from data-adaptive blocks decomposed by Quad-Tree

For the first problem of visual feature learning, where the block based feature extraction is not adaptive to the image spatial statistics and has limited capability to investigate the image structure, we develop a feature learning algorithm using data-adaptive blocks decomposed by Quad-Tree. Instead of arbitrarily partitioning the image region into fixed-sized blocks (patches) or relying on hand-designed local subregions, Quad-Tree partitions the image region into blocks of

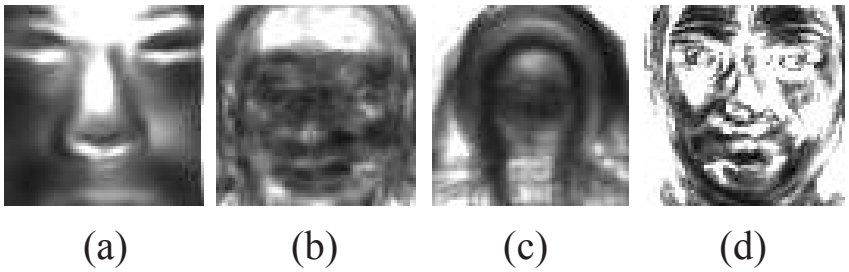


Figure 1.5: Templates for four databases: (a) Uchimura 3D, (b) ATT, (c) IFD, (d) JFFE database. (Copyrighted by ICPR2012 [ZLM12])

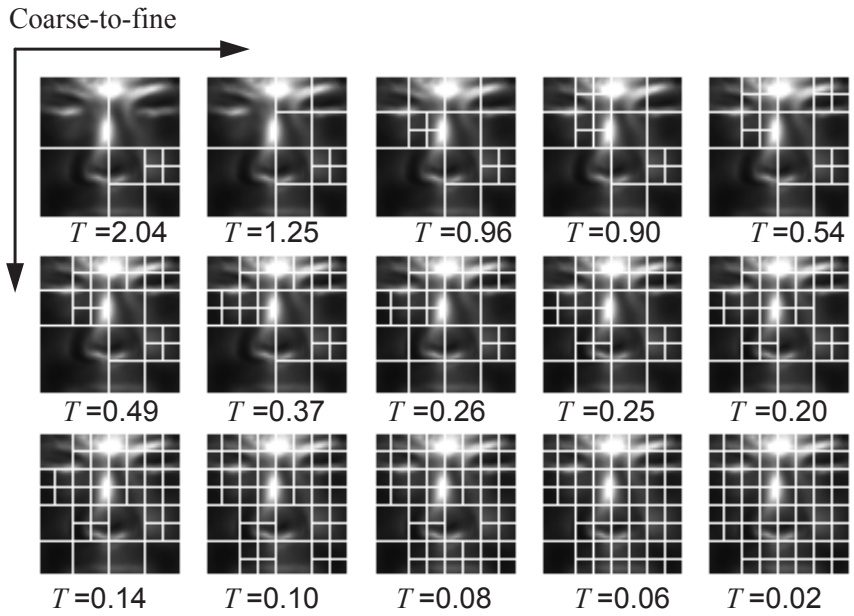


Figure 1.6: An example of Quad-Tree partitions on a face image in Uchimura database. (Copyrighted by ICPR2012 [ZLM12])

variant sizes according to the feature distribution across the image region. Specifically, in order to make the partition adapts to different datasets, we do not perform Quad-Tree decomposition on the original image region directly. Instead, we first generate a template to model the feature distribution across an image dataset and then perform Quad-Tree decomposition on the template according to a splitting criterion defined by variance of a region. If the variance of a subregion is higher than a threshold variance T_v, then it will be partitioned into four quadrants. The threshold T_v is a key parameter for Quad-Tree partition. Since local variances usually vary on different databases, even in one database, it is rather difficult to find the best partition using one threshold. We, therefore, define a set

of thresholds in a descending order, and generate a series of Quad-Trees with different resolutions. The template image is split into less and bigger blocks when threshold is large, but into more and smaller blocks when threshold is small. Figure 1.5 shows the templates on four face databases (Uchimura 3D, ATT, IFD, and JFFE), and figure 1.6 illustrates Quad-tree partitions on the template image of the first database [ZUKZ10].

The Quad-Tree decomposition is performed on the template image instead of raw images. For each image dataset, we can generate a template image, which encodes the feature distribution of all images in the dataset. As the generation of the template image is adaptive to the image dataset, the Quad-Tree partition is also adaptive to image dataset. For each image dataset, we can generate a template image, and then have a Quad-Tree partition for a given threshold.

Algorithm 2: Hierarchical feature learning using Quad-Tree structure of images

After local feature analysis, the next step is on how to combine local features together. New trends on feature learning utilize hierarchical models, which can generate high-level features form local features according to a hierarchical architecture. However, as aforementioned, existing hierarchical learning algorithms mainly utilize image pyramid based hierarchical models, which are not data-adaptive and not flexible enough for high-level feature generation. Moreover, existing works utilize layerwise unsupervised pre-training for weight assignment of local subregions, which makes upper layers have an increasing loose connection with the image input. To solve these two problems, in this algorithm, we develop an alternative hierarchical feature learning algorithm using the Quad-Tree structure of images. The Quad-Tree structure explicitly learns the image hierarchical structure. In contrast to the image pyramid, which is complete tree structure for all the images without any change, our Quad-Tree structure is a data-adaptive tree structure, where leaf nodes can locate at different levels instead of the the lowest level. For each image dataset, we can get a Quad-Tree structure for a given threshold. Then high-level features can be formulated according to the tree structure. The essential idea behind this algorithm is shown in Figure 1.7. This algorithm combines a top-down image structure prediction and a bottom-up deep feature integration together. Instead of using a layer-wise unsupervised pre-training for weight assignment, the weights of local subregions are assigned according to their discriminant powers encoded in the Quad-Tree structure. An example of weight assignment is illustrated in Figure 1.8.

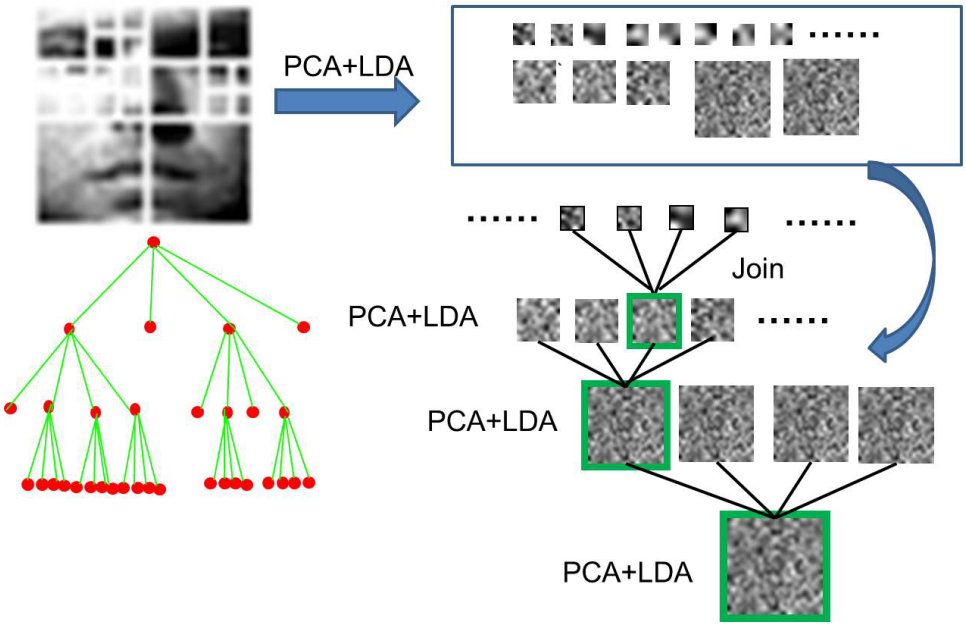


Figure 1.7: Hierarchical feature learning using Quad-Tree structure of images. (Copyrighted by CRC Press / Balkema (Taylor & Francis Group) [ZLM14b])

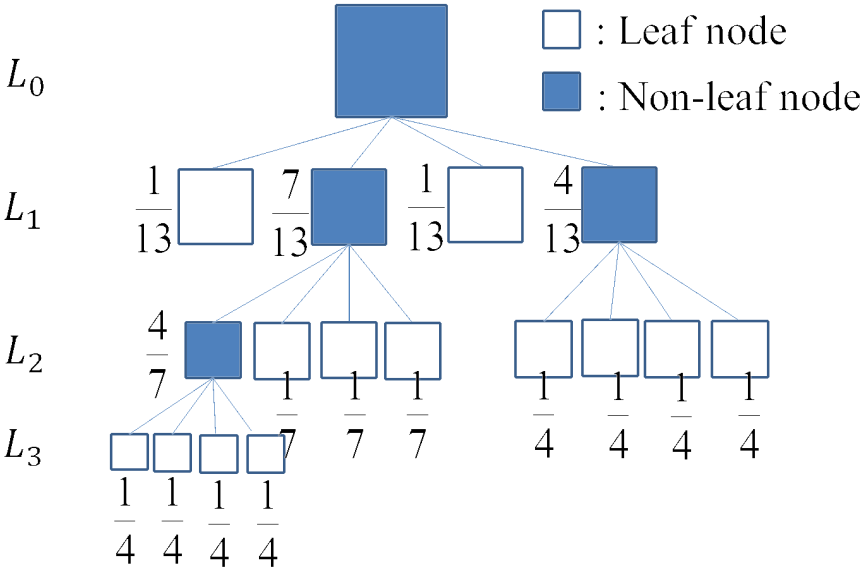


Figure 1.8: An example of weight assignment according to Quad-Tree structure.

The main difference between Algorithm 1 and Algorithm 2 is two-folds: 1. The weight assignment is different. In Algorithm 1, we treat each block equally and assign the same weight to them for feature learning. In Algorithm 2, we find that the classification accuracy of blocks are actually different from each other.

Thus, we assign different weights to them according to the tree structure. 2. The learning architecture is also different. Algorithm 1 utilize image blocks for feature learning, which locates at the image planer surfaces. In algorithm 2 we define a hierarchic model (denoted by the Quad-Tree structure) to learn more abstract features based on this multi-layer architecture.

Algorithm 3: Feature learning from enlarged training data encoded by Quad-Tree for Small Sample Size problem

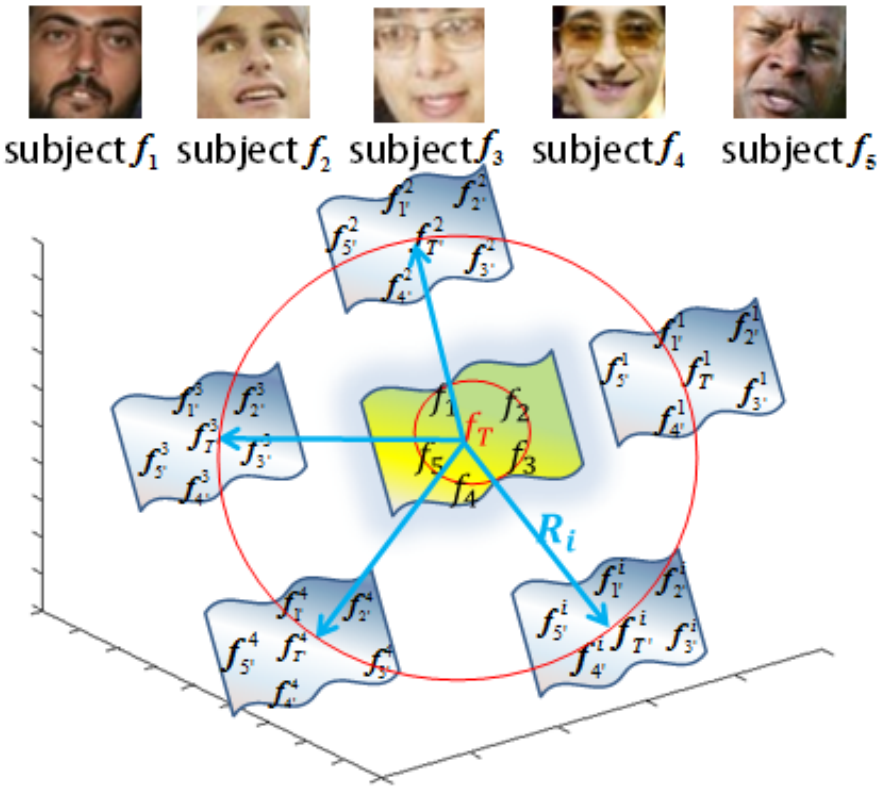


Figure 1.9: Feature space expansion through generating new samples [ZLM14a].

Supervised based feature learning usually requires large training data. While existing algorithms can work well when the training data is large enough, they are not well suited to feature learning with small training data. The small training data will lead to the SSS problem, under which existing algorithms suffer from overfitting problem severely. In order to evaluate the readability of our method, we investigate feature learning with small training data in this algorithm. To solve the SSS problem, we develop a feature learning algorithm from enlarged

training data encoded by Quad-Trees. The small training data is first augmented by Quad-Tree based image encoding method through generating new samples. Then ensemble learning is performed in the expanded feature space to deal with the SSS problem. The main idea of this method is illustrated in Figure 1.9. The small training data locates around the center of the feature space. To explore the new possibilities of the small sample space, we add a set of random matrices to the template of the original small sample set. For each random matrix R_i, we can get a new template, on which Quad-Tree decomposition is performed to detect the regions of high density of discriminant features. The images in the original small sample set are re-organized according to the Quad-Tree structure to generate a new sample set. And a base classifiers is trained from this new training sample set for ensemble. For given L random matrices, we can get L new training sample sets and thus have L base classifiers. Since all new samples are generated by re-organizing the original images according to Quad-Tree structures, they should locate around the original small sample set. The new samples expand the feature space to a large extent.

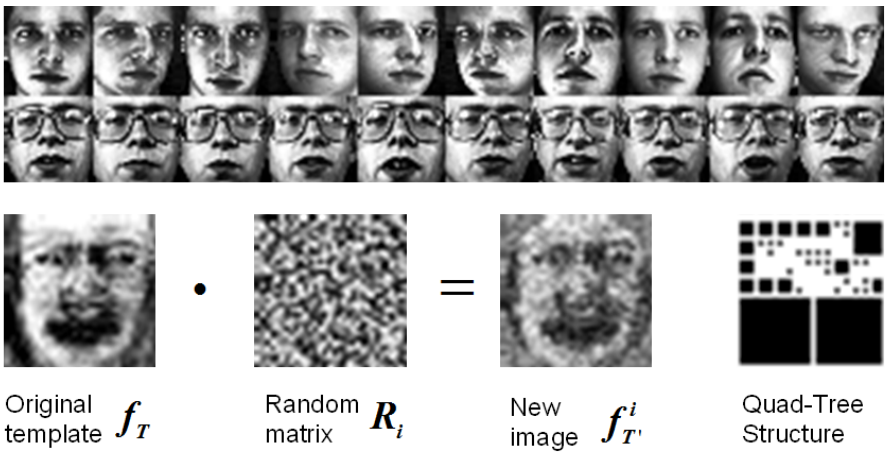


Figure 1.10: Illustration of Quad-Tree based small data augmentation on an example face database: Yale2. (Copyrighted by IET [ZLM13])

The procedure of new sample generation based on Quad-Tree is illustrated in Figure 1.10. Since not all generated samples are appropriate for classification. Integrating inappropriate base classifiers may lead to the Diversity/Accuracy dilemma in ensemble. To solve this problem, we introduce a tailored 0–1 Knapsack solution for base classifier selection. Our Quad-Tree based ensemble framework solves two problems of existing ensemble learning algorithms: (1) base clas-

sifiers trained form insufficient training data are not diverse for ensemble; (2) integrating all base classifiers may lead to the Diversity/Accuracy dilemma.

The relationship between Algorithms 1–3 can be explained as follows: Algorithm 1 acts as a warm-up algorithm. It is beginning to use Quad-Tree for image region partition and feature learning. In Algorithm 2, the Quad-Tree based feature learning becomes maturity. We define a more sophisticated hierarchical model using the Quad-Tree structure to learn more abstract features and to assign different weights to local subregions for combination. In Algorithm 3, we extend the Quad-Tree based method to another case of feature learning, where the training data is small. Algorithm 3 solves the SSS problem and makes feature learning be more robust. Algorithm 2 is the key algorithm in this dissertation.

Algorithm 4: Feature learning in dynamic environments using Helmholtz-Hodge Decomposition and Quad-Tree

Feature learning can be performed in static images or dynamic environments (video sequences). While feature learning in static images is relatively easier, it not so for dynamic environments. In order to further evaluate the reliability of our method in dealing with more challenging cases, this algorithm investigates feature learning in dynamic environments (video sequences taken by a moving camera). In the moving camera videos, camera motion and local object motions are mixed and dependent with other. Local object motions can offer motion features to many video based applications such as action recognition, tracking, content based surveillance, etc.. Since the camera motion and local object motions can influence with each other, if we use the original mixed motion field for these applications, these applications will be very challenging and less accurate. Thus, we need to segment object motions from the scene and recover them to their true values. In this case, just Quad-Tree is not enough since the Quad-Tree structure is sensitive to the dynamic background. To solve this problem, we introduce an amended Helmholtz-Hodge Decomposition (HHD) first, which can be used for camera motion compensation. After camera motion compensation, we can then apply Quad-Tree partition in the rested motion field for object motion compensation. The Quad-Tree can not only detect object motion from the scene, it can also deal with depth discontinuities in the scene, which makes our method be available for motion segmentation in 3D scenes, where depth discontinuities lead to motion discontinuities in the scene. After motion segmentation, the camera motion and local object motions can be estimated and recovered to their true val-

ues. The pipeline of the HHD and Quad-Tree based feature learning algorithm in dynamic environments is illustrated in Figure 1.11.

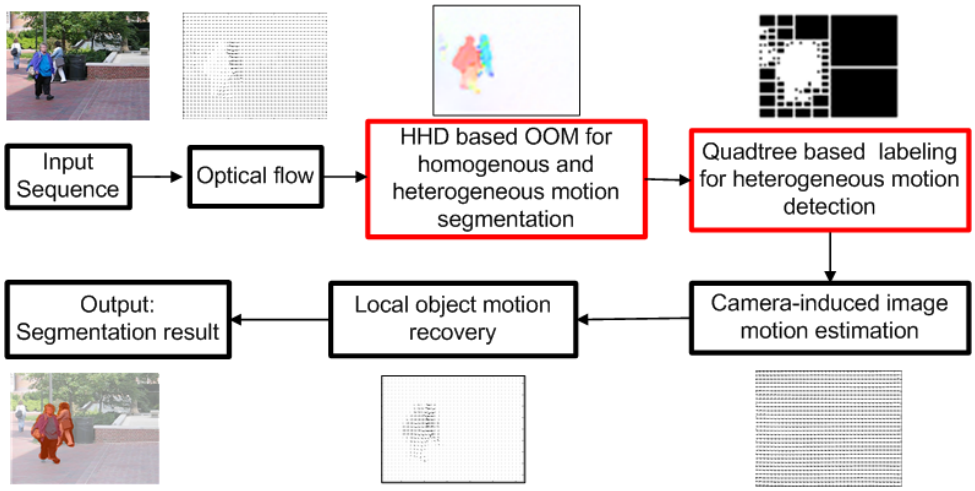


Figure 1.11: Illustration of the feature learning algorithm in dynamic environments using HHD and Quad-Tree, which HHD is performed for camera motion compensation and Quad-Tree is performed for object motion segmentation.

We need to explicitly mention that the first three algorithms are highly related with each other while the fourth algorithm is different from them in some sense. The first three algorithms share the same research background (face recognition). Their problem definition and algorithm description are also similar. However, the research background of the last algorithm is on motion segmentation. It investigates feature extraction from video sequences rather than image dataset. Thus, the problem definition and algorithm description are also different from the first three algorithms. However, these four algorithms share the same Quad-Tree based image encoding, which is the core-aspect of this dissertation.

1.4 Summary of Contributions

The main contributions of this dissertation are summarized as follows:

- We investigate image spatial and hierarchical structure by developing Quad-Tree based image encoding methods and incorporate them with visual feature learning algorithms to make the learning procedure more data-adaptive and powerful.

- We develop a feature learning algorithm from data-adaptive image blocks decomposed by Quad-Tree, which makes the local feature analysis be adaptive to image spatial statistics and more robust to deal with local deformations than image block (patch) based methods.

- We develop a hierarchical feature learning algorithm using Quad-Tree structure of images. Quad-Tree explicitly learn the image hierarchical structure making hierarchical feature learning be more effective. The Quad-Tree architecture is a data-adaptive tree structure and more flexible than image pyramid based hierarchical models. The Quad-Tree structure can also learns the parameters of low-level features for composition.

- We develop a feature learning algorithm using expanded training data encoded by Quad-Tree for small sample size problem. This algorithm augments the small training data by generating new samples, which makes base classifier ensemble be more effective in solving the SSS problem.

- We develop a feature learning algorithm in dynamic environments using Helmholtz-Hodge decomposition and Quad-Tree. This algorithm can be applied for motion segmentation in moving camera videos, which is the fundamental task to many video based applications. This method can extend Quad-Tree based feature learning from static images to dynamic environments more naturally.

- To solve the Diversity/Accuracy Dilemma in ensemble, we develop a base classifier selection algorithm using a tailored 0–1 Knapsack solution, which helps ensemble learning in finding the best combination of base classifiers for ensemble.

1.5 Organization of This dissertation

Chapter 1: Introduction In this chapter, I first describe the main problem and goal of this dissertation. Then, the motivation of this work is given by discussing several challenging issues to existing visual feature learning algorithms. Next, the proposed methods towards these challenging issues are presented including four Quad-Tree based visual feature learning algorithms. Finally, the main contributions and the roadmap of this dissertation are summarized.

Chapter 2: Literature Review In this chapter, I first describe several widely used computer vision features, which belong to three levels: low-level, mid-level, and high-level. Then I make a literature review on existing visual feature learning algorithms, which can be briefly divided into holistic and local based methods. We compare these two kinds of methods by listing their advantages and disadvantages. Then, we make a survey on the recent developed hierarchical learning algorithms. Finally, we introduce another visual feature learning technique: ensemble learning, which can be applied to deal with the SSS problem.

Chapter 3: Feature learning from data-adaptive blocks decomposed by Quad-Tree In this chapter, we develop a feature learning algorithm using data-adaptive blocks decomposed by Quad-Tree. This method can be used for local feature analysis using data-adaptive blocks, which can be directly used in machine learning tasks or act as input to more sophisticated feature learning algorithms (e.g. hierarchical learning) for further processing. We perform Quad-Tree partition on a template image of an image dataset instead of raw images to make the Quad-Tree partition adaptive to image dataset. Since it is difficult to define an appropriate threshold for Quad-Tree parition, we define a set of thresholds instead on a single one to get a set of Quad-Tree partitions accordingly. Good Quad-Tree partitions with high classification accuracy are selected based on a 0–1 Knapsack algorithm for decision fusion. We finally evaluated the proposed algorithm on face recognition using several standard face databases.

Chapter 4: Hierarchical feature learning using Quad-Tree structure of images The method in Chapter 3 is on feature learning from the planar image surface. In this chapter, we use a hierarchical model to learn high-level features from local feature composition. Moreover, in Chapter 3, we treat each local subregion equally and assign the same weight to them for combination. In this chapter, we find different local subregions actually perform differently and we assign different weights to them according to the tree structure. Specially, we use the Quad-Tree structure to define the image hierarchical structure in a top-down manner and utilize the tree structure to learn high level features in a bottom-up way. The tree structure is a data-adaptive one compared to other image pyramid based hierarchical models and weight assignment strategy makes the global feature learning more accurately. Finally, we evaluated this method on several face databases and compared with the method developed in Chapter 3 to show the effective of hierarchical feature learning.

Chapter 5: Feature learning from enlarged training data encoded by Quad-

Tree for Small Sample Size problem While the methods in Chapter 3 and 4 are on feature learning with large training data, this chapter investigates feature learning with small training data. To solve the SSS problem, we develop a feature learning algorithm from enlarged training data encoded by Quad-Trees. The small training data is first augmented by generating new samples using Quad-Trees, and then diverse base classifiers can be generated from the expanded feature space for ensemble. To solve the Diversity/Accuracy dilemma of ensemble, we develop a base classifier selection algorithm using a tailored 0–1 Knapsack solution. It just selects appropriate base classifiers for ensemble. The proposed method was evaluated on the SSS face recognition by comparing with several sate-of-the-art algorithms. We also compare this algorithm with the method developed in Chapter 3 to show the effectiveness of data augmentation in dealing with SSS problem.

Chapter 6: Feature learning in dynamic environments using Helmholtz-Hodge decomposition and Quad-Tree While the last three algorithms are on feature learning from static images, this chapter investigates feature extraction form dynamic environments. We develop an object motion segmentation method in moving camera videos using Helmholtz-Hodge decomposition and Quad-Tree. This method can extend the Quad-Tree based feature learning from static images to video sequences. As the Quad-Tree structure is sensitive to dynamic background, we first perform HHD in the mixed motion field for camera motion compensation. HHD releases the dependency between camera motion and local object motions by separating them into different components. It casts camera-induced image motion into a homogeneous motion field and rests the heterogeneous motion caused by moving cameras and depth discontinuities in a remainder. After camera motion compensation, we then perform a data-driven Quad-Tree partition in the rested motion field for object motion segmentation. The final camera motion and object motions can be estimated and recovered based on a surface fitting using a low-order polynomial function. We evaluated the developed method using several benchmark video sequences on motion segmentation.

Chapter 7: Conclusion We provide a summary of four Quad-Tree based image encoding methods for data-adaptive visual feature learning. Possible directions for future work are also included in this chapter.

Chapter 2

Literature Review

Visual feature learning, which supplies a semantic and meaningful representation for machine learning algorithms, plays a fundamental role in all level computer vision applications: low-level, mid-level and high-level. Features can be exacted from either the whole image region or local subregions, which refer to the holistic and local based feature extraction and representation, respectively. Holistic based methods extract a single feature vector from the whole image region and local based methods extract a series of feature vectors (bag-of-features) from a number of local subregions. These two types of methods have their own advantages and disadvantages. Recently, it is argued that the global feature representation formed based local feature analysis should be preferred than bag-of-features. Many sophisticated feature learning algorithms have been proposed for global feature learning. State-of-the-art algorithms utilize a multilayer framework to learn a hierarchical feature representation, which refer to the "hierarchical learning" algorithms.

This chapter focuses on the literature review on holistic and local based methods by comparing their advantages and disadvantages and makes a survey on popularly used hierarchical learning algorithms. Specifically, we first describe several widely-used computer vision features corresponding to different level of computer vision applications in Section 2.1. Then we make a literature review on holistic and local based approaches in Section 2.2. Next, the essentially idea behind hierarchical learning and several representative algorithms are explained in Section 2.3. Section 2.4 introduces an important feature learning technique, "ensemble learning", which can be used to deal with the Small Sample Size problem. Finally, Section 2.5 summaries this chapter.

2.1 Features in Computer Vision

Features span a wide spectrum in computer vision applications. In the low-level vision, which concerns about basic images processing tasks, some image primitives such as color, texture, edges, corners, blobs, ridges are extracted from the image region for further processing. In the mid-level vision, which need to specify which image parts belong to the object, several mid-level features are generated from low-level features to specify object components. In the high-level vision, where the whole object need to be involve in advanced processing such as object recognition and scene understanding, we need to use several high-level feature that can reflect the global structure of the image data. According to the abstraction level, computer vision features can be largely divided into:

- Low-level features : image primitives, such as color, texture, pixel values, edges, corners, blobs, bridges, etc..
- Mid-level features : object parts or components, which are usually generated from low-level features and the refer to subregion of the image other than the whole region.
- High-level features: whole part of object, which are calculated from the entire image.

Three-level features are illustrated according to a hierarchical feature learning architecture in Figure. 2.1. Features at the lowest level denote low-level features; features at middle layers refer to mid-level features; and features at the top layer refer to the high-level features.

For low-level features, many local feature descriptors were proposed in the literature, such as: SIFT, HOG, RIFT, SURF, GLOH, Shape Context, Textons, Spin images [Lee10]. These feature descriptors are more invariant to local deformations such as scale, rotation, translation than raw pixel values.

Popular mid-level features include bags-of-features, spatial pyramids, etc.. Mid-level features are usually formulated by the composition of low-level features based on a sequence of interchangeable modules such as coding and spatial pooling.

The highest level of the deep architecture refers to the global feature representation, which is transferred into liner or non-liner machine learning algorithms for high-level vision applications. Actually, many literature works do not differentiate between mid-level features and high-level features. They treat them as

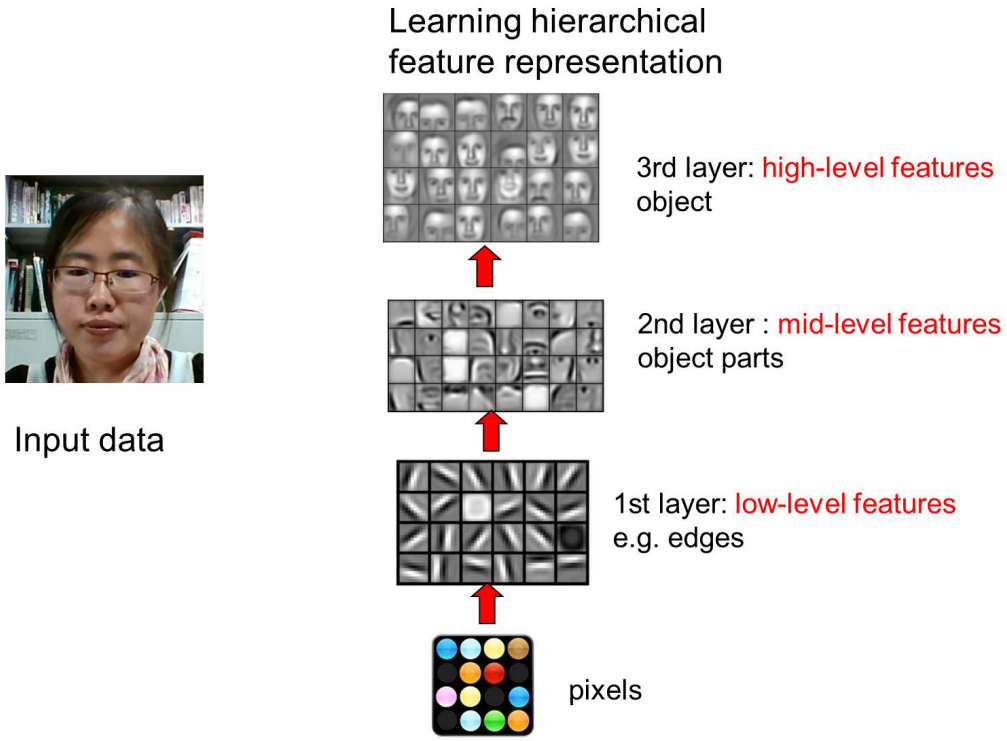


Figure 2.1: Illustration of three level features: low-level, mid-level and high-level.

a unit. They just use low-level features and high-level features (or lower-level features and higher-level features) to specify features at different levels.

2.2 Feature Extraction and Representation

Feature can be extracted either from the whole image region or local subregions, which refer to the holistic and local based feature learning algorithms, respectively. Figure 2.2 illustrates the essential idea of these tow types of methods. In this section, we make a literature review on these two kinds of methods and compare their advantages and disadvantageous.

2.2.1 Holistic Feature Extraction and Representation

Holistic based methods are the most commonly used algorithms in object recognition, such as face recognition. Appearance based methods have been acknowledged as one of the most important holistic based methods, which extracts a feature vector for each image based on the lexicographic ordering of raw pixel

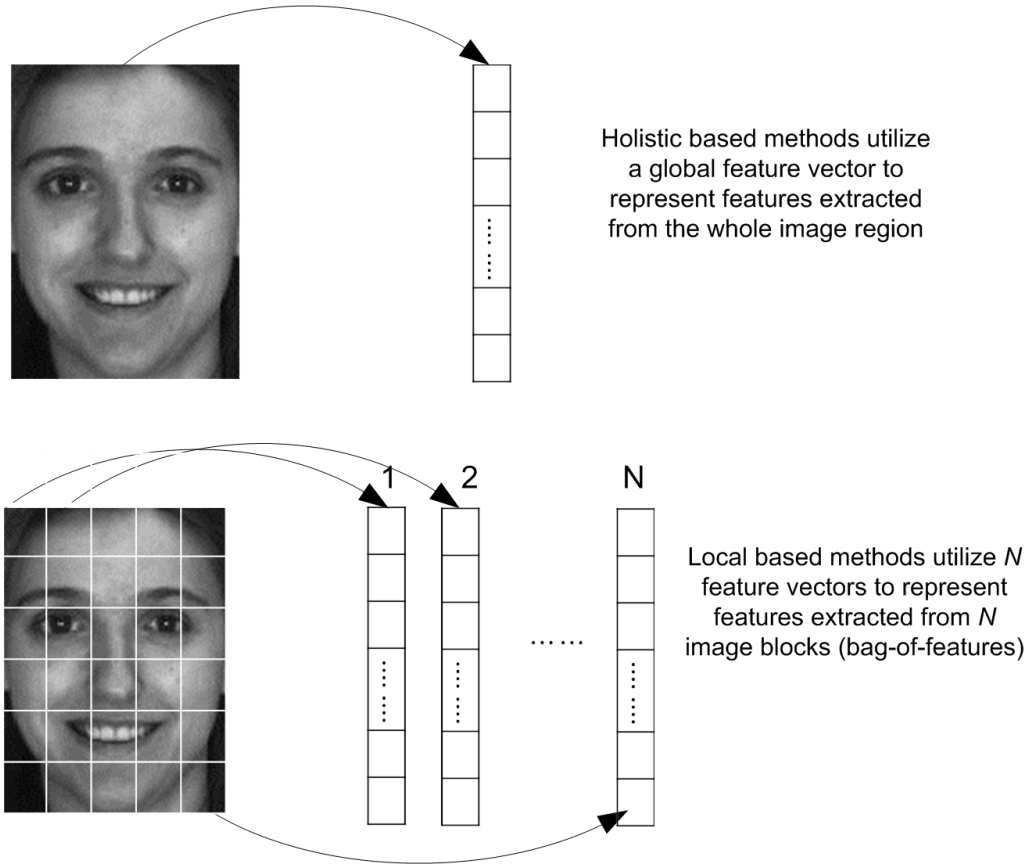


Figure 2.2: Illustration of holistic and local based feature extraction and representation methods. The holistic methods extract a global feature vector from the whole image region while local based methods extract a series of feature vectors from a number of image blocks or patches (bag-of-features).

values [SHR10]. An image can be considered as a point in a high-dimensional feature space, the dimensionality of which is determined by the size of images (in terms of pixels). Even for a small image of size 100×100, there is a very large feature space of dimensional 10,000. The curse-of-dimensionality is a big problem of appearance based methods, which prevents the high ability of machine learning algorithms in making decisions. To solve this problem, several dimension reduction algorithms were proposed. One of the most well-known method is the Principal Component Analysis (PCA) [OPM+11], which projects the images into a low-dimensional feature subspace and the major information of the image data can be persevered in this feature subspace meanwhile. This dimension reduction strategy can significantly reduce the number of features involved in machine learning algorithms, which has been applied to many research areas such as computer vision and statistical pattern recognition.

After dimension reduction, we can directly transfer the lower-dimensional feature vector to machine learning algorithms, or employ other discriminant feature learning methods in the low-dimensional feature subspace to generate more discriminant features. Well-known discriminant learning algorithms include the linear discriminant analysis (LDA) based methods [BHK97], independent component analysis (ICA) based methods [HO00], and their different variances such as kernel based methods [CHH+07, Yan02].

Compared with previously designed geometry based approachable, the appearance based methods have obtained many impressive results in object recognition applications. Holistic methods can preserve the global shape and whole structure of the image data. However, they have weak ability to deal with local deformations caused by image clutter, occlusion, facial expression, etc.. Thus, several local based approaches have been proposed to deal with these problems.

2.2.2 Local Feature Extraction and Representation

In contrast to holistic based methods which convert the whole image into a single fracture vector, local based methods extract a series of feature vectors from a number of local subregions. Local based approaches describe the image content in a local subregion/part of image. Through adding some transformation rule or specific measurements in each local subregion, the features extracted from local subregions are more invariant to external factors such as scale, translation, and rotation deformations.

In fact, many researchers do not differentiate between holistic and local based methods. They believe their very natures are the same. For each local subregion, we can treat it as a whole image region and extract a holistic feature representation from it. Similarly, local feature presentations can also be concatenated to form a holistic representation. Instead of using holistic and local to specify these two kinds of methods, many researchers use the term of global and bag-of-features to describe them considering the number of feature vectors they have.

Many local based methods have been proposed. The fist usage of the local based approaches are the Block PCA [GA04], Block LDA [CLZ04], UP [DHG+10], which partition the image region into several blocks and then extract a feature vector from each block. The second usage of local based approaches is what we can usually find in the literature works, such as Discrete Cosine Transform (DCT) [MGKR12], Discrete Wavelet Transform (DWT) [BG14], Gabor filtering, Local Bi-

nary Pattern (LBP) [OPM+11], Locality Preserving Projections (LPP) [CHH+07], etc.. We need to note that these methods are called local in the literature since they extract features from image blocks. However, they are holistic in nature since they still give rise to a single feature vector for each image by summarizing local feature vectors together.

When talking about local base methods, we need to discuss an important thing: image region partition for local subregion definition. Many mechanisms have been proposed for image region partition, which can be briefly divided into: (1) regular grid based methods; (2) unsupervised image segmentation based methods; (3) hand-craft region based methods. Image block based method is one of the most simple and widely used regular grid based approaches. It partitions an image region into a set of blocks of the same size according to the grid arrangement. Sliding window based method is a more sophisticated method, which define several overlapping blocks (windows) across the image region. Unsupervised image segmentation algorithms have also been widely used for image region partition, such as the superpixel based method [ASS+], [], etc.. Some state-of-the-art results are achieved by hand-craft region based methods. They manually design several local subregions for object recognition. An example method is the 30region method, which designs 30 regions according to facial component arrangement, which is used for face recognition [Spr11].

Holistic and Local Methods: How to Choose?

Many recognition/classification applications need to specify which methods to be used, holistic or local. These two kinds of methods both have their advantages and disadvantages. Holistic methods can preserve the global shape and configural information of images (e.g. the correlationship between facial components), which have been acknowledged as an important evidence for object recognition. However, these methods rely much on the image registration, which is quite time consuming and challenging since images can be taken in a variety of situations. Moreover, holistic methods have weak abilities in dealing with local deformations caused by occlusion, illumination changes, image clutter, etc.. Under such cases, the use of local based methods becomes a better choice. In addition, bag-of-feature methods also perform better than holistic methods when the training data is very small. They can train multiple base classifiers from a set of feature vectors and integrate them together to get the final prediction result. Base classifier integration can improve the performance of recognition using a single classi-

fier. This will be explained later. However, local based methods also have their weak points. For example, they can not preserve the global shape and structure of image data as well as holistic methods. Based on these explanations, we find that the choice of holistic or local based methods depends on the applications and data property, which should be determined case by case.

2.3 Hierarchical Feature Learning

As aforementioned, both holistic and local based approaches have their advantages and disadvantages. Recently, it is argued that the global feature representation formed based on local feature analysis can have a better representational ability than bag-of-features. Many sophisticated algorithms have been proposed for global feature learning. Early works utilize simple combination rules such as feature fusion, decision fusion to combine local features to a global feature representation. These methods may be effective in solving relatively simple tasks. But they have limited capability in dealing with complicated applications involving natural images and visual scenes. Recently, theoretical results ([Has86, BL07, FBL09, JDaHW14]) suggest the importance of learning hierarchical feature representations using multilayer frameworks, where high-level features are formulated by the composition of low-level features. Such hierarchical learning algorithms have achieved many state-of-the-art results in computer vision and pattern recognition applications, such as object detection [GDDM14], localization [SEZ+14], recognition [KSH12], image segmentation [FCNL13, CFCNL14], natural language processing (NLP) ([CW08]), robotics ([HES+08]), information retrieval ([SMH07]), etc.. They have also beat or matched state-of-the-arts in several scientific competitions such as Emotion Recognition in the Wild [KPB+13], Kaggle Dogs vs. Cats [SEZ+14], ILSVRC [KSH12], etc..

Most hierarchical learning algorithms start from block based low-level feature extraction. They encode the low-level features extracted from image blocks using several algorithms such as sparse coding and then high-level features are formulated by the composition of low-level features. Many hierarchical learning algorithms have been proposed, such as the neural network based algorithms ([BL07, RPCL07, VLBM08, CW08]) and graphical model based algorithms ([HO06, HOT06, ZLC09]). Early works utilize fully-connected neural networks, where the parameters of low-level features are learned based on backpropgation [RGEW86]. Since the backpropgation is very likely to get stuck in local minima

especially when the parameters of low-level features are randomly initialized, hierarchical learning does not become successful until the development of two technologies recently: Convolutional Neural Networks (CNNs) [LBBH98] and Deep Belief Networks (DBNs) [HO06, HOT06]. While CNNs employ local-connectivity based neural networks and DBNs utilize a layer-wise unsupervised pre-training for parameter initialization. These two techniques spark the growing interest of hierarchical learning in the recent years. Next, I will first briefly describe these two methods.

Convolutional Neural Networks(CNNs)

CNNs were first presented in [LBBH98] with an attempt to construct an intelligent computer vision system that can replicate the behavior of the human visual cortex. They utilize local connected feedforward networks to investigate the spatial correlationship between neuros of adjacent layers. A typical pipeline of CNNs comprises a convolution process and subsampling process. During the convolution process, a bank of filters are convoluted with an image. During the subsampling process, this bank of filters are aggregated at a local area of image (named as a local receptive field), where the size of the output can be reduced. These two processes are alternated until the size of the final feature representation is as small as what we desired [MS12]. The subsampling process ensures the effectiveness of CNNs in reducing the number of parameters to be learned and improves the general abstraction of the final feature representation meanwhile.

Deep Belief Networks(DBNs)

DBNs were first proposed by Hinton et al. in 2006 [HO06, HOT06]. They are multi-layer graphical models working based on greedy layer-wise unsupervised learning of Restricted Boltzmann Machines (RBMs). RBMs are bipartite undirected graphical models with two layers (Markov Random Field). A set of latent (or hidden) binary random variables are learned from RBMs for parameter initialization of the whole architecture. The final feature representation is generated based on supervised learning based fun-tuning. DBNs combine unsupervised pre-training and supervised fine-tuning together to generate the final feature representation. Besides RBM, there are some other techniques for unsupervised pre-training, such as stacked auto-encoders [WOY14] and stacked convolutional auto-encoders [MMCS11].

In contrast to CNNs which employ local connectivity of neural networks, DBNs still utilize a stochastically learning architecture with a dense connectivity between a set of input variables. The main merit of DBNs is the introduction of the unsupervised pre-training strategy for parameter initialization, which performs much better than random initialization.

Apart from the original version of CNNs and DBNs, there are also some other variants proposed in the literature. A representative method is Convolutional Deep Belief Networks(CDBNs) [HLLM12], which is a hybrid version of CNN and DBN. It has been applied to deal with large image classification problems involving high dimensional images [KSH12].

Deep PCA

The families of CNNs and DBNs utilize a connectionist network for hierarchical feature learning. However, these methods do not explicitly investigate the deep structure hypothesis. These algorithms are usually very complex and the generated feature representation is difficult to analyze. Recently, Many arguments have been posed on the high complexity of connectionist network based hierarchical learning. An alternative interesting study is to employ non-connectionist architectures beyond neural networks. Deep PCA [MS12] is one the most representative methods, which explicitly investigates the the deep structure hypothesis using a simplified and non-connectionist architecture. Deep PCA first splits the image region into equal-sized small patches. From the lowest level, the Multilayer PCA is applied on each local subregion for dimension reduction and feature extraction. Lower-level features are then combined according to the split-and-joint process to generate higher-level features. By applying PCA on each layer according to the deep architecture, the feature map becomes more abstract and powerful for image classification/recognition.

2.4 Ensemble Learning

While existing visual feature learning algorithms can enable machine learning algorithms in making accurate predictions, they are not well suited to learning situations with small training data. The small training data leads to the Small Sample Size problem [ZLM13], under which existing visual feature learning algorithms suffer from the overfitting problem severely. Recently, a machine-learning tech-

nique known as ensemble learning has received considerable attentions towards SSS. Ensemble algorithms are based on the idea that a pool of different classifiers (referring to base classifiers) can offer complementary information and bear different recognition errors. Thus, they can benefit from base classifier collaboration, where the overall discriminative power is higher than a single classifier.

A typical ensemble learning framework comprises two basic procedures: (1) base classifier definition, and (2) base classifier ensemble. Many algorithms have been proposed for the first procedure. Feature subspace aggregation based methods are one of the most representative methods, which have been widely applied in image classification and recognition tasks especially when the image data or dataset is of high-dimensional. Base classifiers a generated from different feature subspaces to bear different classification errors. Existing multi-subregion methods can be mainly divided into: (1) global feature selection based on random subspace [WT06]; (2) patch (block) based local feature extraction [TE10]; (3) global and local feature integration [ZZHS12]. The further discussion on these three kinds of methods will be included in Chapter 4.

In the context of ensemble, diversity has been acknowledge as a very important aspect, which measures the disagreement degree in the output of a set of base classifiers. Obviously, there is not a clear accuracy gain in an ensemble built from a set of identical base classifiers. The ideal situation is actually a set of base classifiers bear with different errors, such that they are combined to minimize the effect of these failures. Some ensemble rules are utilized to enhance the diversity of ensemble, such as bagging, boosting, and random forests. They harness the diversity of ensemble by training base classifiers in different feature subsets corresponding to different training samples. That is, each base classifier is trained from a subset of training samples to create a different classification hypothesis. Multiple hypothesizes are then combined to form an ensemble.

To aggregate multiple hypothesizes of base classifiers, several strategies are employed, such as majority voting. Apart from majority voting, some other sophisticated strategies are also employed, such as weighted majority voting [TE10], probabilistic approximation [Bar96], bayesian combination [KG12]. However, majority voting is still the most commonly used strategy thanks to its simplicity and effectiveness compared to more complex methods.

Existing ensemble learning algorithms mainly perform ensemble in the original training data. Although they can improve the performance in some sense, these methods still suffer from an underlying problem that the species of base

classifiers learned from insufficient training data are not diverse enough to form an adequate feature space. They are very likely to be tightly correlated and make similar errors. Having less diverse base classifiers has become a bottleneck of existing ensemble algorithms. To solve this problem, a potential new direction of ensemble learning is to augment the small data first and then perform ensemble in the enlarged training data more effectively [ZLM13]. The details of this strategy will be discussed in Chapter 4.

2.5 Summary

In this section, we first reviewed several widely used computer vision features corresponding to low-level, mi-level and high-level vision applications. Then, we compared two kinds of feature extraction and representation methods: holistic and local based methods. Next, we discussed an emerging and rapidly developed visual feature learning technique: hierarchical learning. Finally, we introduced another visual feature learning technique for the SSS problem: ensemble learning.

By comparing the holistic and local based methods, we find that both of them have advantages and disadvantages. The choice of which method to be used should depend on both applications and data property. By reviewing several hierarchical learning algorithms, we observe that they have high capability in dealing with real-world applications involving natural images. As to ensemble learning, through explaining the main idea of ensemble, we find that this strategy is effective in alleviating the SSS problem. New direction of ensemble has also been figured out in this chapter: ensemble learning based on data augmentation.

Chapter 3

Feature Learning from Data-Adaptive Blocks Decomposed by Quad-Tree

Features can be extracted from the whole image region or local subregions, which refer to holistic and local based methods. The holistic methods (named as global methods) can preserve the global shape of images while local methods (named as bag-of-features) are effective in dealing with local variations such as illumination changes, occlusion, local deformations, missing data etc.. Recently, it is argued that global feature learning based on local feature analysis is preferable than either the global methods or bag-of-features. Thus, local feature analysis is the first step of many feature learning algorithms. Existing well-known local feature extraction methods include (1) block based methods; and (2) hand-designed local subregion based methods. The block based methods are straightforward and not adaptive to image spatial statistics while the hand-designed local subregion methods rely on human efforts, time consuming, incomplete and usually depending on data registration.

To address these problems, this chapter develops a data-adaptive block based feature learning algorithm using Quad-Trees. Quad-Tree partitions the image region into data-adaptive blocks according to the feature distribution across the image region. Good subregions with high discriminant powers are selected using a 0–1 Knapsack algorithm for integration. The Quad-Tree based image region partition and 0–1 Knapsack based subregion selection make our method be region specific and data-adaptive. We evaluated the proposed method on several standard face databases. The experimental results demonstrate the effectiveness of our method.

3.1 Problem Description and Related Works

In holistic appearance-based methods, an image with size $n \times m$ pixels is represented by a long vector in the $n \times m$ dimensional space. Principal Component Analysis (PCA) [TP91] and Linear Discriminant Analysis (LDA) [BHK97] are two widely accepted representatives under this framework. Compared to the key-feature based approaches, the results of the novel appearance-based approach were striking. A whole range of new natural objects can be recognized well based on the vector representation. However, being based on a global description, local variations, such as motion deformation, illumination changing, data missing again form a major problem. They make modifications on the representation coefficients on the feature space limiting the usefulness of the system to cope with local variations. Multi-subregion fusion methods ([QSBS10], [BPF09], [AGA09]) were proposed as a solution for this problem. They divide the image into a set of disjoint subregions, perform recognition on each subregion and fuse the results.

Existing well-known local subregion methods can be mainly divided into two types: (1) block (patch) based algorithms, which divides an image region into regular blocks or utilizes a sliding window to locate subregions for feature extraction. Block based methods have been applied for many object recognition and classification problems. However, block based methods are very straightforward and not adaptive to image statistics. They have limited ability in investigating the actual feature distribution of natural images, which are usually non-uniform. (2) hand-designed local subregion based methods. These methods provide several hand-designed spatial regions for feature extraction. For example, in [Spr11] a set of 30 regions were designed for face recognition, see Fig. 3.1, which showed better performance than block based methods. However, these hand-designed regions are highly depending on the prior-knowledge of human beings. The process is time consuming, incomplete and usually relying on data registration. Experiential results demonstrated that these methods can outperform others on well registered databases but performs not so satisfactory on not well-registered databases.

Apart from the image region partition, another problem with existing local subregion based methods is on how to select subregions to build up the best combination of them. Some local subregions may not appropriate for classification. Integrating classification results of inappropriate subregions may reduce the final classification accuracy. To address these two problems, this chapter develops

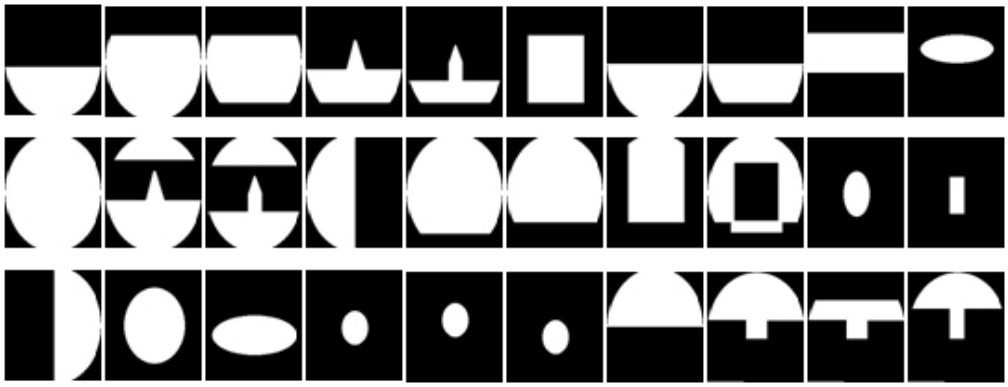


Figure 3.1: The hand-designed 30 regions [Spr11].

a data-adaptive image region partition method using Quad-Trees and introduce a local subregion selection method based on 0–1 Knapsack solution. These two strategies makes the local feature extraction be data adaptive and more robust and accurate.

3.2 Algorithm Architecture

The algorithm can be described as follows: first, Quad-Tree decomposes an image region into multiple local sub regions recursively according to an image homogeneity criterion function. If a region does not meet this criterion function, it will be decomposed into four subregions. In order to make the Quad-Tree partition adapts to variant databases, we perform Quad-Tree partition on a template image instead of original images. The template image is generated according to the feature distribution on the image region. In our method, the criterion of Quad-tree decomposition is using LDA-motivated total variance, which ensures the robustness to local noise and efficiency of computation. As not all local subreigons are appropriate for classification, we introduce an optimized solution based on 0-1 Knapsack algorithm to select subregions and fuse them to achieve expected performance.

3.2.1 Modeling the Feature Distribution using a Template

Instead of dividing the face region into a uniform blocks, Quad-tree partitions the face region into data-adaptive blocks of variant sizes by means of local discriminative variance. Here, the variance refers to the discriminative feature density.

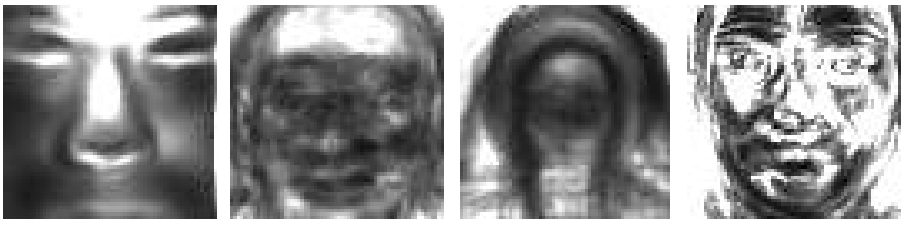
Larger partition implies that the block has lower density of discriminant features, and vice versa. Smaller partitions mean that the block has higher density of discriminant features. To make the Quad-Tree partition more robust to local noises, we consider to use the variance of all face images in the entire image dataset. Motivated by the idea of LDA, we define a template face by Eq. 3.2, which encodes the discriminative feature distribution by maximizing the between-class scatter matrix S_b and minimizing the within-class scatter matrix S_w (See Eq. 3.1). The template face represents the distribution of discriminative features across the face region for all the images in a database. That is, the total variance of entire database equal to the variance of the template (see Eq. 3.3). Example template faces for four face databases are shown in Fig. 3.2.

$$\begin{cases} S_b = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T, \\ S_w = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T, \end{cases} \tag{3.1}$$

$$template = diag(\frac{S_b}{S_w}), \tag{3.2}$$

$$totalVar = variance(diag(\frac{S_b}{S_w})), \tag{3.3}$$

where μ denotes the mean face of all face classes, μ_i is the mean face of class X_i, N_i denotes the sample number of class X_i, and x_k represents the k-th sample of class X_i.



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

Figure 3.2: Templates for four databases: (a) Uchimura 3D, (b) ATT, (c) IFD, (d) JFFE database. (Copyrighted by ICPR2012 [ZLM12])

3.2.2 Quad-Tree based Image Region Partition

Quad-Tree decomposition is performed based on a criterion function defined in Eq.3.4. If the variance of a region R (block) or one of its four sub-blocks (subR) is higher than a threshold variance ($T * totalVar$) , then R is split into smaller blocks.

$$\begin{cases} doSplit(R) = totalVar(R) > T * totalVar, \\ doSplit(R) = doSplit(R) | doSplit(subR). \end{cases} \tag{3.4}$$

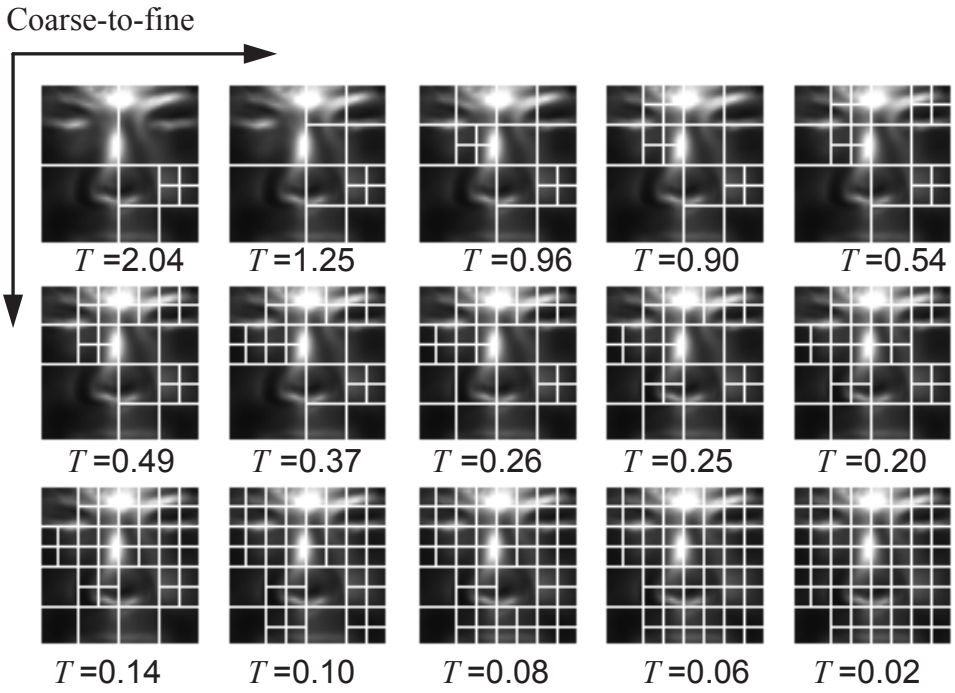


Figure 3.3: The Quad-tree partitions of template on Uchimura 3D database. The threshold T is in a descending order from left to right, from top to bottom. (Copyrighted by ICPR2012 [ZLM12])

The difficulty in Quad-Tree decomposition is how to define an appropriate threshold T. Local variances usually vary on different databases, so one universal threshold is not a good idea in our method. Even in one database, it is rather difficult to find the most appropriate threshold to get the best partition. In this dissertation, we define a set of thresholds instead of a single one. The thresholds are defined between a maximum value and a minimum value in a descending order. The maximum value is the start point where the template image begins to partition. The minimum value is the end point, where the template is partitioned into smallest blocks completely. The maximum value and the minimum value

are also depending on databases. That means, they are adaptive to image dataset. The template is split into less and bigger blocks for a large threshold, but into more and smaller blocks for a small threshold. Given a set of thresholds, we can get a set of Quad-Tree partitions with different resolutions (the partitions are from coarse to dense). Figure 3.3 illustrates the Quad-tree partitions on the template image of Uchimura 3D database [ZUKZ10].

3.2.3 Data-Adaptive Block based Feature Extraction

The Quad-Tree partition is performed on the template images instead of raw images. The template image is generated according to the feature distribution of all the images in an image dataset. For each image dataset, we can generate a template image. Since the generation of the template image is data-adaptive, the Quad-Tree partition is thus data-adaptive. Next, we will re-organize the original face images according to the Quad-Tree partition in the following way.

For each Quad-Tree partition, we re-organize the face images in the original image dataset according to the partition result. As explained above, larger partitions (blocks) mean that the density of discriminative features in them is low. Thus, there is no need to keep its original size. We will do downsampling on large blocks (the size is larger than the smallest size) by resizing them to a smaller size $((d/2) \times (d/2))$, where d is the width of large blocks (in pixel). For smallest blocks mean they the density of their discriminant features are high. Thus, we will not do down-sampling on them. Finally, block resizing result in a set of new images whose sizes are smaller than original face images. For each Quad-Tree partition, we will get a new set of images, from each appearance based feature extraction method is applied to generate feature subspace. In this dissertation, we use PCA+LDA for the extraction of features, where PCA is utilized for dimension reduction and LDA is employed to generate discriminant features in the PCA feature subspace.

Actually there are also some other kinds of feature extraction methods. The reason why we use PCA+LDA in this dissertation is as follows: the core aspect of this dissertation is Quad-Tree based image encoding methods, which can generate data-adaptive image representations for visual feature learning. The developed image encoding methods are independent with classifiers. PCA+LAD is recognized as a base line algorithm for face recognition and has been extensively studied in many object recognition/classification tasks. Using such base line clas-

sifiers without any other constraints can better demonstrate the effectiveness of our image encoding methods. For the future work, we can try more sophisticated classifiers to make the learning procedure more robust and powerful.

3.3 Decision Fusion of Multiple Subregions

The subregions obtained from Quad-Tree partitions represent portions of the face region in different locations. Their performances (recognition accuracy) are different from each other since they contain variant discriminative features. A single subregion is usually unlikely to achieve the best performance since the discriminative features contained in each single suregion are usually limited. To solve this problem, integration of the classification results of multiple subregions may work better than a single subregion. However, it is difficult to find the best combination of all subregions, some subregions may not be appropriate for classification. For example, in our method, some thresholds may not be good for Quad-Tree partition. Thus, the classification results of such local subregions are not accurate. In our work, motivated from the idea of 0-1 Knapsack problem, we covert the problem of subregion selection and fusion into an optimization problem, which is explained in the following section.

3.3.1 0–1 Knapsack Algorithm

The conventional 0–1 knapsack problem is depicted like this: given a set of items and a knapsack, each item has a mass and a value while the knapsack has its own capacity. We would like to select some items and put them into the knapsack so that the total value of selected items can be maximized while the total weight should not beyond the capacity of K. This can be interpreted as:

$$\begin{aligned} & max(value(K)) \\ & subject\ to \quad mass(K) \leq t_c \end{aligned} \tag{3.5}$$

where $value(K)$ and $mass(K)$ represent the total value and total mass of selected items in K, and t_c denotes the capacity of K. This problem is very similar to our problem. Given a set of subregions $S = S_1, S_2, \cdots, S_n$, each subregion S_i has two parameters: the size W_i and the discriminant power V_i. We aim at choosing a subset O of S such that the total size of selected subregions does not exceed the

capacity of O and the total value is maximized. The size (in pixel) and discriminative power are similar as the weight and the value in the conventional 0–1 Knapsack problem. As mentioned before, the discriminative power of a subregion can be represented by the density of feature distribution. Thus, we use the trace of $S_b^i./S_w^i$ in a subregion to represent its discriminant power (see Eq.3.6).

$$V_i = trace(\frac{S_b^i}{S_w^i}) \tag{3.6}$$

The capacity of the subset O is defined by the total size of the 30 regions illustrated in Fig. 3.1 since we want to investigate whether our method can work better than or equal to the performance of the state-of-the-art [Spr11] under the same computational cost. Like the conventional 0–1 Knapsack problem, we can also use dynamic programming to solve this optimization problem.

3.4 Experimental Evaluation

3.4.1 Datasets

Our method was evaluated on seven databases: one 3D database (Uchimura 3D database [ZUKZ10, ZLM12, WUKZ12]) and six widely used 2D databases: IFD [JM], JFFE database [DJL$^+$10], ORL (previously named as ATT database [NH98]), Extended Yale (Yale2) [GBK01], AR [MB98], and FERET [PMRR00]. The face data in these databases are under varied conditions including a variety of head poses (Uchimura, ORL, IFD, AR), illumination changing(Uchimura, ORL, IFD, Yale2, AR, FERET), partial data missing (Uchimura), facial expressions (ORL, IFD, JFFE, Yale2, AR, FERET), and facial details (e.g. with glasses or not: ORL, AR, FERET).

- **Uchimura 3D database**: This dataset was create by a 3D laser scanner: VIVID910. A 3D point cloud data of 640×480 was created along with color images. This dataset consists of 38 subjects. For each person, there are 10 samples, of which 9 samples are of various head poses and 1 sample of different illumination conditions. From the 3D data, we first generate a range image as the visual feature image according to the method in in [ZUKZ10]. And then features are extracted from the range image as other intensity images. For recognition, 5 samples of each subject are randomly selected as training data while the rest as test data.

Figure 3.4: Sample images of seven databases: (a)Uchimura 3D database, (b) IFD, (c) JFFE, (d) ORL, (e) Extended Yale (Yale2), (f) AR, (g) FERET.

- **Indian Face Database (IFD)**: this database contains 40 subjects. Each subject has 11 samples varied in head poses and facial expressions. Five samples per subject are randomly selected as training data and the rest as test data.

- **The Japanese Female Facial Expression Database (JFFE)**: this database was initially created for facial expression recognition. Ten subjects are contained in the databases. Each subject has 7 samples according to 7 basic facial expressions. For recognition, 4 samples are randomly selected as training data while the rest as test data.

- **ORL database (previous ATT database)**: this dataset was created by AT&T Laboratories Cambridge, which was founded in 1986 as the Olivetti Research Laboratory (ORL). Previously, it was named as ATT database. Currently, it is better known as ORL database. There are 40 subjects in this databases. Each subjects has 10 samples with variances in facial expressions, open or closed eyes, with glasses or no glasses, scale changes (up to about 10 percent), head poses. Five samples per subject are randomly selected as training data while the left ones as test data.

- **Extended Yale database (Yale2 database)**: more than 20,000 single light source images of 38 subjects with 576 viewing conditions (9 poses in 64 illumination conditions) are contained in the database. To evaluate the ro-

bustness of our method on the illumination changes, 5 samples of the 64 illumination conditions are randomly selected as training data, the left 59 images as test data.

- **AR database**: 4000 color face images of 126 people (70 men and 56 women) are contained in this database. They are frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). Each subject has 20 samples, which are divided into two sections taken in different time (separated by two weeks). As in [ZZHS12], the fist session images of 50 male subjects and 50 female subjects was selected as a subset for performance evaluation. For each subject, we randomly choose 5 samples for training and the left 9 images for test.

- **FERET database**: consists of 13539 images corresponding to 1565 subjects. Images differ in facial expressions, head position, lighting conditions, ethnicity, gender and age. To evaluate the robustness of our method to facial expressions, we worked with subset of front faces labeled as *Fa, Fb*, where *Fa* with regular facial expressions, and *Fb* with alternative facial expressions. There are 1201 subjects in the dataset. *Fa* samples of each subject are selected as training data, while *Fb* as test data.

The performances of the first three databases (Uchimura 3D, ifd, and JAFEE) are reported in the published work [ZLM12]. However, these databases are very small and out-of-date. To make the evaluation more convinced, we add four more databases (ORL, Yale2, AR, FERET) in the experiments, which are considered as benchmark databases to evaluate face recognition algorithms. Face images in the ORL, Yale2, and AR databases are aligned to 32×32 using the method in [CHH$^+$07]. FERET database is normalized using the *CSU Face Identification Evaluation System 5.1* [CSU03]. The face images are cropped to the same size 32×32. Sample images of all databases are shown in Fig. 3.4.

A template was created for each database first. Quad-Tree decomposition is then performed on each template to generate subregions. Our method generate $17 \sim 25$ Quad-Trees depending on databases. For each Quad-Tree partition, face images are encoded for feature extraction. The dimension of PCA subspace is defined by the number of samples minus the number of subjects in each database.

3.4.2 Experimental results

The performance of our method is compared with: (1) the block based method, which is the same the complete Quad-Tree based image region partition; (2) the aforementioned state-of-the-art method using hand-designed 30 regions [Spr11]. In [Spr11], since they donot know how to select the best combination of 30 regions, we used the following four combinations for comparison, which outperformed other combination based on experimental results: the total 30 regions, the 28 regions of highest accuracy, the 20 regions of highest accuracy, and the maximum single region (the single region, which has the best performance). These four combinations based on majority voting for integration were used for comparison. The results are illustrated in Table 3.1.

Table 3.1: Recognition rate of our Quad-tree based method compared to the block and 30-region based method [Spr11] on seven face databases.

Method \ Database		3D	IFD	JFFE	ORL	Yale2	AR	FERET
Block		**99.9**	88.1	95.0	92.7	76.3	83.2	59.6
30-region+ PCA-LDA	30 regions	98.9	87.3	95.0	93.5	91.6	88.6	75.0
	28 regions	98.9	86.0	95.0	94.0	91.2	88.6	73.3
	20 regions	98.9	87.7	95.0	93.5	91.9	88.6	**75.3**
	max single	99.3	87.7	**99.9**	94.5	89.9	86.6	67.3
Data-Adaptive Block		**99.9**	**91.4**	**99.9**	**96.0**	**91.9**	**89.6**	70.3

From the experimental results, we have the following observations:

Observation [1]: On Uchimura 3D and IFD database, we found none of combinations of the 30-region method could outperform the conventional PCA-LDA method. The reason for Uchimura 3D database is that no region in 30-region method covers the whole facial area. Therefore, the lack of face boundary, which is an important discriminative clue for recognition, downgrades the performance. Since our method partition the face image region from coarse to fine, at least one subregion can cover the whole face image, no important discriminative clue is lost.

Observation [2]: On IFD database, head poses are pretty diverse such that it is difficult to do facial registration. Since those 30 regions are designed based on registration, they are very likely to ignore the discriminative information on the data without well registration. On the contrary, our method adapts the shape of

partitioned subregion according to the data property.

Observation [3]: For the JFFE database, one single subregion achieves a better performance than the fusion of multiple regions and the PCA-LDA. By observing the shape of that subregion, we find that it removes most local variations, and happens to be very suitable for the face data in that database. In our method, the Quad-Tree partition also generates a most appropriate subregion for the face data. So, this single subregion is employed other than merely focusing on fusion of multiple subregions.

Observation [4]: On the ORL database, the 30-region method outperforms the conventional PCA-LDA, but it depends on which regions are chosen. We can see that the accuracy using the 20 regions are higher than that using 28 regions and 30 regions. Observing the recognition rate of each single region, we find that the rates vary obviously. These selected regions are the top 20, and involve much less local variations. As our method selects the combination of subregions using the optimized solution for 0-1 Knapsack problem, it can select the appropriate combination automatically.

Observation [5]: On the Yale 2 database, which contains much amount of local variations caused by illuminations changes, the 30-region method and our method improve the performance of conventional PCA-LDA significantly. This suggest the importance of using Multi-Subregion methods to deal with local variations. However, the difference between our method and the 30-region method is not as obvious. This might because the subset of Yale 2 database involved in experiments only contains frontal face images, which have been well-registered, under which the 30-region method can have a very good performance.

Observation [6]: On the AR database, the 30-region method improves the performance of conventional PCA-LDA method obviously. This is because the AR database features frontal view image with facial expressions, illumination changes, and occlusions. The 30 region method works well in in dealing with local deformations. And since the database contains no large variations, such as head poses, the 30 region method works well. However, we can still find that our method still works better than the 30 region method, which demonstrate the effectiveness of our method.

Observation [7]: For the above mentioned 6 databases, we find out our method outperforms the conventional PCA-LDA method and the state-of-the-art 30-region method obviously. Our method benefits from the data-driven image partition and it can be widely applied to diverse databases, especially for those

with a large number of variations. Moreover, the dynamic programming solution for 0-1 knapsack problem guarantees our method to select the best combination of multiple subregions automatically. We need to acknowledge that our method performs worse than 30-region on FERET database, which only contains frontal face images and has been every well registered using the *CSU Face Identification Evaluation System 5.1* [CSU03]. We argue that performance of 30-region method relies much on the data registration, which is consuming and challenging in many piratical feature learning situations with natural images. What is worse, since authors do not know how to select the best combination of multiple subregions, it is a serious issue when applying this method. In conclusion, the above experimental results proof our method has better ability in handling local variations and adapting to various face databases.

3.5 Summary

This chapter develops a feature learning algorithm from data-adaptive blocks decomposed by Quad-Trees. Quad-Tree partitions the image region into local subregions according to the feature distribution across the whole database. The local subregions are of different sizes and contain variant discriminant information. Our Quad-Tree based method is data-adaptive more appropriate than block based and hand-designed subregion methods for local feature extraction. To select good subregions for classification integration, we introduce an 0–1 Knapsack algorithm in this chapter. This chapter answers two questions remaining in other multi-subregion methods: 1. what subregions are good partitions? and 2. how to select and fuse these subregions? The Quad-Tree partition and subregion selection makes our method be data-adaptive and more robust than existing methods. Experimental results on four databases demonstrate that our method has better performance than block based and the hand-designed 30 region based method.

Chapter 4

Hierarchical Feature learning using Quad-Tree structure of images

In Chapter 3, we have investigated local feature analysis using data-adaptive blocks decomposed by Quad-Tree. In this chapter, we will explore how to combine local features to form a global feature representation. New trends in global feature learning utilize a multi-layer framework, where high-level features can be formulated by the composition of low-level features. The hierarchical architectures makes the final global feature representation be more abstract and have higher generalization ability. Difficulties in hierarchical feature learning lie in (1) how to define an appropriate hierarchical architecture and (2) how to learn parameters of local subregions for composition. Existing works mainly utilize image pyramid based hierarchical model and employ a unsupervised pre-training for parameter learning. Image pyramids are not data-adaptive and not flexible enough to explore the image hierarchical structure and the unsupervised pre-training is not well-suited for supervised learning applications.

To solve these problems, this chapter develops an alternative hierarchical learning algorithm using Quad-Tree structure of images. Quad-Tree explicitly exploits the image hierarchical structure by a top-down Quad-Tree partition. Then deep features can be formulated according to the tree structure in a bottom-up way. In contrast to image pyramid, Quad-Tree is a data-adaptive and more flexible tree structure for hierarchical learning. Moreover, parameters of local subregions can be defined according to the tree structure. We evaluated this algorithm on four standard face datasets. The comparison results with several methods using global, local features and a canonical deep learning algorithm "Deep PCA" demonstrate the better performance of our method than existing methods.

4.1 Problem Definition and Related Works

After local feature analysis, the next step is how to combine local feature together to formulate a global feature representation. Early works utilize simple liner combination rules such as weighted sum rule. However, the linear combination is not sophisticated enough to investigate the global image structure. New trends of feature learning utilize a multi-layer framework to learn global feature from local features, where high-level features can be formulated by the composition of low-level features. Such hierarchical learning algorithms have achieved many state-of-the-art results in computer vision tasks, such as object detection [GDDM14], localization [SEZ+14], recognition [KSH12], image segmentation [FCNL13, CFCNL14], natural language processing (NLP) ([CW08]), robotics ([HES+08]), information retrieval ([SMH07]), etc.. These methods benefit from the hiearhical architecture, which makes the final feature representation more abstract and have higher generalization ability for classification.

Previous works on hierarchical feature learning utilize fully-connected neural networks, where the parameters of low-level features are learned based on backpropgation [RGEW86]. Since the backpropgation is very likely to get stuck in local minima especially when the parameters of low-level features are randomly initialized, hierarchical learning does not become successful until the development of two technologies recently: Convolutional Neural Networks (CNNs) [LBBH98] and Deep Belief Networks (DBNs) [HO06, HOT06]. The main idea of CNNs and DBNs are described as follows. CNNs utilize local connected feedforward networks to investigate the spatial correlationship between neuros of adjacent layers. A typical pipeline of CNNs comprises a convolution process and subsampling process. During the convolution process, a bank of filters are convoluted with an image. During the subsampling process, this bank of filters are aggregated at a local area of image (named as a local receptive field), where the size of the output can be reduced. These two processes are alternated until the size of the final feature representation is as small as what we desired [MS12]. DBNs are multi-layer graphical models working based on greedy layer-wise unsupervised learning of Restricted Boltzmann Machines (RBMs). RBMs are bipartite undirected graphical models with two layers (Markov Random Field). A set of latent (or hidden) binary random variables are learned from RBMs for parameter initialization of the whole architecture. The final feature representation is generated based on supervised learning based fun-tuning. DBNs combine unsupervised pre-training

and supervised fine-tuning together to generate the final feature representation. Beyond CNNs and DBNs, there are also some other hierarchical feature learning algorithms, such as Deep PCA [MS12], which utilize a tree structure to learn more abstract features by applying PCA on each layer of the tree. Deep PCA has demonstrated high performance in object recognition and classification tasks such as face recognition.

However, existing local connectivity based networks still have some problems. The first problem relates to the hierarchical architecture. Existing methods mainly utilize image pyramid based networks. Image pyramid is a complete tree structure. We can have the same structure for all the images when the size of the image pyramid is fixed. Thus, these methods are not data-adaptive and not flexible enough to investigate the image hierarchical structure. The second problem is on the parameter learning. Existing methods (such as DBNs) utilize a layer-wise unsupervised pre-training for parameter learning. However, the unsupervised pre-training is not well-suited for supervised learning applications.

To solve these problems, this chapter develops an alternative hierarchical feature learning framework using Quad-Tree structure of images. Quad-Tree explicitly learns the image hierarchical structure, which can guide in hierarchical feature learning. The benefits are two-fold: (1) The Quad-Tree structure is data-adaptive tree structure and more flexible than image pyramid for hierarchical feature learning; (2) the parameters (weights) of local subregions can be defined according to the tree structure other than using unsupervised learning strategy.

4.2 Algorithm Architecture

The proposed hierarchical learning algorithm comprises a top-down Quad-Tree decomposition and a bottom-up deep feature integration [ZLM14b]. An overview of the hierarchical feature learning algorithm is illustrated in Fig.4.1. Unlike evenly partitioning image, Quad-Tree decomposes an image into uneven subregions according to the density of discriminant information across the image region. A smaller subregion contains a higher density of discriminant information, but a larger one contains low. Low-level features can be extracted from each subregion and higher-level features can be formed by low-level features according to the tree structure. Our Quad-Tree decomposition functions like a convolutional neural network, but builds up a incomplete hierarchy structure. And the aggregation of these uneven features in the multi-layer structures functions like a re-

cursive neural network, which encodes the correlations between local subregions. Next, I will explain each step based on face recognition as research background.

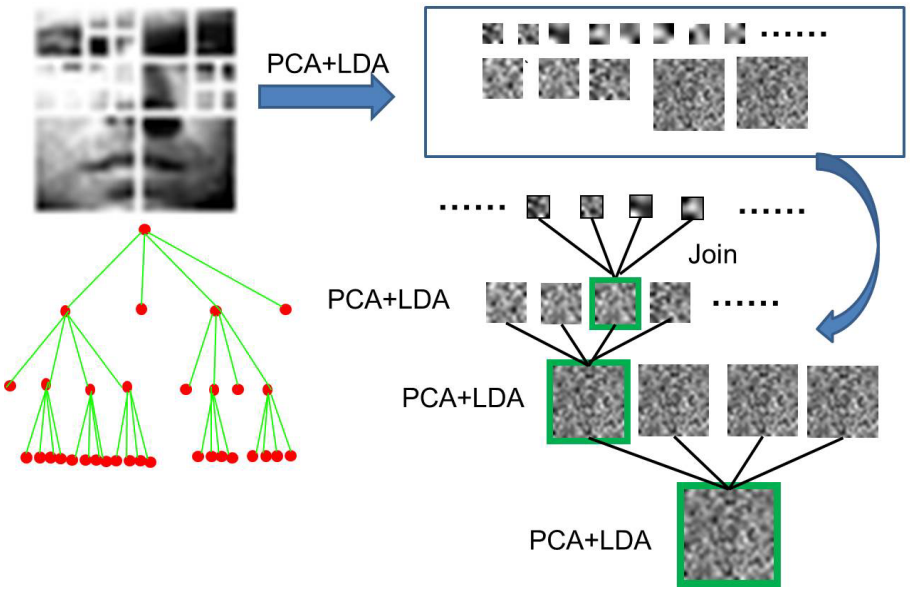


Figure 4.1: Hierarchical Feature Learning using Quad-Tree structure of images. (Copyrighted by CRC Press / Balkema (Taylor & Francis Group) [ZLM14b])

4.2.1 Top-down Image Structure Prediction

Instead of dividing the face region into a uniform blocks, Quad-Tree partitions the face region into data-adaptive blocks of variant sizes by means of local discriminative variance. Here, the variance refers to the discriminative feature density. Larger partition implies that the block has lower density of discriminant features, and vice versa. Smaller partitions mean that the block has higher density of discriminant features. To make the Quad-Tree partition more robust to local noises, we consider to use the variance of all face images in the entire image dataset. As stated in Chapter 3, we define a template image motivated by the idea of LDA, which encodes the discriminative feature distribution by maximizing the between-class scatter matrix S_b and minimizing the within-class scatter matrix S_w (See Eq. 4.1).

$$\begin{cases} S_b = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T, \\ S_w = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T, \end{cases} \tag{4.1}$$

where μ denotes the mean face of all face classes, μ_i is the mean face of class X_i, N_i denotes the sample number of class X_i, and x_k represents the k-th sample of class X_i. The template image T is defined by the diagonal vector of $\frac{S_b}{S_w}$ in Eq. (4.2), which represents the distribution of discriminative features across the face region for all the images in a database. For the face images of size $m \times n$, the covariance matrix $\frac{S_b}{S_w}$ is of $(m \times n)^2$. The template image is of the same size as the original face images.

$$T = diag(\frac{S_b}{S_w}), \tag{4.2}$$

We perform a top-down image region partition by performing Quad-Tree decomposition on T. Quad-Tree partitions T into smaller blocks recursively according to a criterion function defined in Eq.(4.3). If the variance of a region (starts from entire region of T) is higher than a threshold t_v, then $T$is split into four sub-blocks with the same size. After Quad-Tree partition, we have an uneven partitioned image / incomplete tree structure of an image as shown in Fig.4.1).

$$doSplit(r) = \begin{cases} true, while(var(r) \geq t_v), \\ false, otherwise, \end{cases} \tag{4.3}$$

The template T is split into less and bigger blocks for a large t_v, but into more and smaller blocks for a small t_v. Usually, it is rather difficult to find the best partition using a universal threshold. We, therefore, define a set of thresholds in a descending order as in Chapter 3 and generate a series of Quad-Trees partitions.

4.2.2 Bottom-up Hierarchical Feature Learning

For each Quad-Tree partition, we first extract low-level features from each subregion using PCA+LDA and then higher-level features can be formulated by the composition of lower-level features according to split-and-joint path encoded in the tree structure. For each joined node (non-leaf node) on each layer, we perform PCA+LDA further to generate more abstract features. Finally, we can construct a hierarchical feature representation according to the tree structure [ZLM14b].

Specifically, hierarchical feature learning produces a hierarchy of feature representing map, in which the higher layers correspond to more abstract overall description of images. It also encodes the correlation among the lower layers. While existing hierarchical feature learning algorithms implicitly learn the image structure by exploring the correlationship between local subregions in a bottom-up

way, our algorithm explicitly predicts the image structure by using the aforementioned Quad-Tree based image region partition in a top-down manner. Fig.4.1 shows that a face image is partitioned into many blocks with varied sizes. Blocks without green rectangle denote the leaf nodes, which can locate at different levels in the tree. These leaf nodes can be used as the input for local feature extraction based on PCA+LDA, and we select the top k_i vectors as a feature basis, where k_i is less than the corresponding block size, i denotes the level index in the tree hierarchy. The smaller i is, then bigger k_i becomes. Each block is projected into a feature subspace encoded by PCA+LDA, and the four reduced-dimensionality neighbor blocks are then joined together back to their father node in their original split order. For the new joined-nodes, we apply PCA+LDA again to generate more abstract features. And these features will act as new input to create up the higher layer. This process is repeated until reaching the root node. Since we apply PCA+LDA on each layer, we name this deep feature learning method as deep PCA. That is, our hierarchical feature learning is performed based on split-and-joined process according to the Quad-Tree structure and using a deep PCA as dimension reduction and feature extraction.

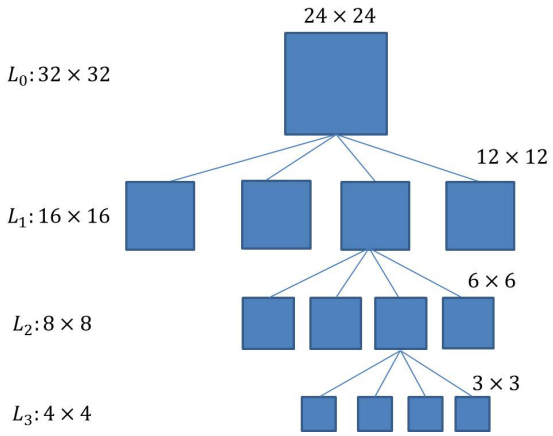


Figure 4.2: An example of bottom-up feature integration using Quad-Tree. This figure shows the number of features extracted from each local subregion for composition. We assume the original image size is 32×32. If the image size is 64×64, these numbers should be doubled.

An example of feature extraction is illustrated in Fig.4.2. The smallest blocks in Quad-Tree partition are of size 4×4 at level 3 ones. We apply PCA+LDA to these blocks, extract the top 3×3 vectors from each to describe the local feature. The four neighbors are then joined in the inverse order of the partition back to the father node at level 2, in which the original blocks have size of 8×8. At level 2, if

a block was not further decomposed, PCA+LDA extracts top 6×6 vectors which has the same size with the newly joined block from level 3. We recursively apply PCA+LDA extraction, and join the neighbor blocks to upper layer. Eventually, the procedure stops at root node. The last PCA+LDA selects about 30 top features as a vector. Obviously, these features preserve not only the global information thanks to the feature hierarchy, but also the local feature from blocks partitioned by our Quad-Tree partition.

4.2.3 Weight Assignment According to Quad-Tree Structure

The difficulty in high-level feature interpenetration is how to define the dimension of each local subregion for composition. In Algorithm 1 (Chapter 3), we assume that each local subregion has the same discriminant power for classification. We think they contribute the same to the final feature representation and thus we assign them the same weight and extract the same dimension of features for composition. However, from experimental results, we find that the actual discriminant power of each local subregion is different from each other. In this algorithm, we explore to assign different weights to local suregions (including leaf nodes and non-leaf nodes). The weight of each local subregion corresponds to the dimension of its PCA subspace. A high weight implies that this local subregion contains higher density of discriminant information. Thus, we need to extract high dimensional features, and vice versa.

The weight of each local subregion is defined according to the Quad-Tree structure. Since the larger weight implies that a local suregion contains higher density of discriminant features, which means the number of leaf nodes under this local subregion is larger. Thus, we use the number of leaf nodes under each local subregion to define its weight. Specifically, the weight of each node $subR_k$ is defined by the number of leaf nodes in the subtree rooted at $subR_k$ against that number rooted by its father node $subR_j$. An example of the Quad-Tree structure with weight assignment is shown in Fig. 4.3. When each group of four brother nodes are joined to generate a higher level node (their father node) $subR_j$, the dimension of the feature subspace of $subR_k$ is defined by the $w_k * d_j$, where w_k is the weight of $subR_k$ and d_j is the dimension of the feature subspace of $subR_j$.

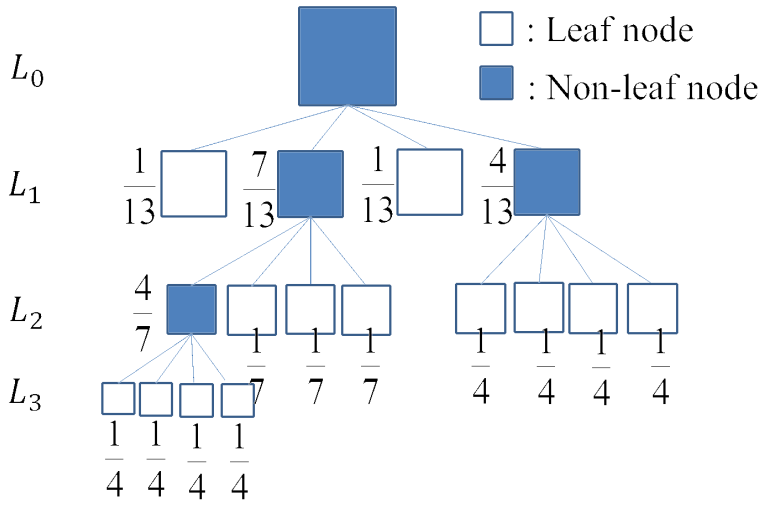


Figure 4.3: An example of Quad-Tree partition with weight assignment.

4.3 Experimental Evaluation

In this section,we first introduce the four standard databases used in the experiments. These four databases are the same as the ones used in Chapter 3. Then, we evaluated the performance of our algorithm by comparing with standard PCA+LDA method and several other state-of-the-arts. We also compare with the data-adaptive block method developed in Chapter 3, where the global feature representation is generated without using hierarchical feature learning.

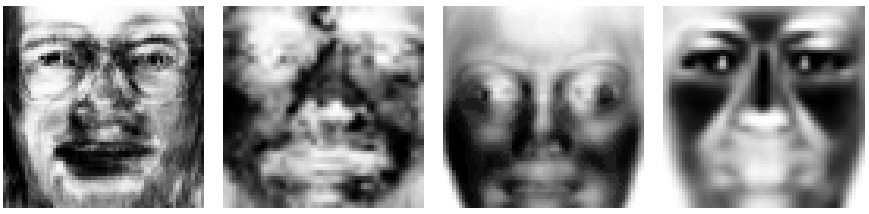
4.3.1 Databases

In order to compare with the method in Chapter, we used the same four databases in the experiments: ORL [SH94], Extended Yale (Yale2) [GBK01], AR [MB98], and FERET database [PMRR00]. The description of each database is as follows.

- **ORL database (previous ATT database)**: this dataset was created by AT&T Laboratories Cambridge, which was founded in 1986 as the Olivetti Research Laboratory (ORL). Previously, it was named as ATT database. Currently, it is better known as ORL database. There are 40 subjects in this databases. Each subjects has 10 samples with variances in facial expressions, open or closed eyes, with glasses or no glasses, scale changes (up to about 10 percent), head poses. Five samples per subject are randomly selected as training data while the left ones as test data.

- **Extended Yale database (Yale2 database)**: more than 20,000 single light source images of 38 subjects with 576 viewing conditions (9 poses in 64 illumination conditions) are contained in the database. To evaluate the robustness of our method on the illumination changes, 5 samples of the 64 illumination conditions are randomly selected as training data, the left 59 images as test data.
- **AR database**: 4000 color face images of 126 people (70 men and 56 women) are contained in this database. They are frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). Each subject has 20 samples, which are divided into two sections taken in different time (separated by two weeks). As in [ZZHS12], the fist session images of 50 male subjects and 50 female subjects was selected as a subset for performance evaluation. For each subject, we randomly choose 5 samples for training and the left 9 images for test.
- **FERET database**: consists of 13539 images corresponding to 1565 subjects. Images differ in facial expressions, head position, lighting conditions, ethnicity, gender and age. To evaluate the robustness of our method to facial expressions, we worked with subset of front faces labeled as *Fa*, *Fb*, where *Fa* with regular facial expressions, and *Fb* with alternative facial expressions. There are 1201 subjects in the dataset. *Fa* samples of each subject are selected as training data, while *Fb* as test data.

Face images in the ORL, Yale2, and AR databases are aligned to 32×32 using the method in [CHH$^+$07]. FERET database is normalized using the *CSU Face Identification Evaluation System 5.1* [CSU03]. The face images are cropped to the same size 32×32. The template images generated for each database are shown in Fig.4.4. An example of Quad-Tree partitions on ORL database are shown in Fig.4.5. Our method generates $8 \sim 15$ Quad-Trees depending on databases.



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

Figure 4.4: Template images on four databases: (a) ORL, (b) Yale2, (c) AR, (d) FERET. (Copyrighted by CRC Press / Balkema (Taylor & Francis Group) [ZLM14b])

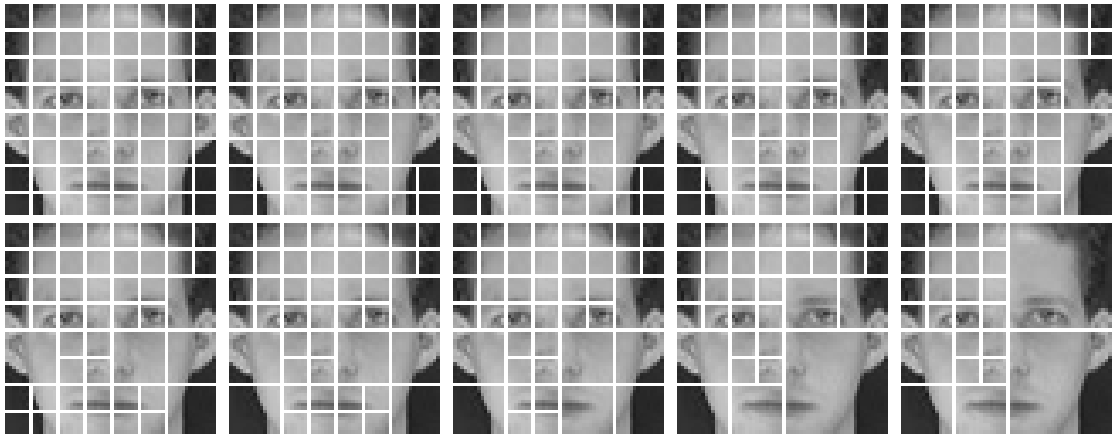


Figure 4.5: Quad-Tree partitions on the template image of ORL database.

4.3.2 Experimental Results

To demonstrate the good performance of our method, we compared our method with several existing algorithms using global feature, local features, and canonical deep feature, respectively. Beyond hierarchical learning, there are also some other global feature learning algorithms, which formulate the global feature representation based on local feature analysis. One of the representation methods is the multi-scale patch-based collaborative representation (MPCRC)[ZZHS12]. This method integrates the complementary information contained at different patch scales to form the final global feature representation. The comparison methods of this chapter are: (1) the conventional PCA+LDA method; (2) MPCRC Zhu12; (3) the 30-region method [Spr11], mentioned in Chapter 3, which defines 30 subregions according to the experimental experience; (4) the Data-Adaptive Block based method developed in Chapter 3; (5) the Deep PCA [MS12]. Table 4.1 illustrates the comparison results.

Table 4.1: Recognition accuracy of our Quad-Tree based hierarchical learning method compared to other state-of-the-arts.

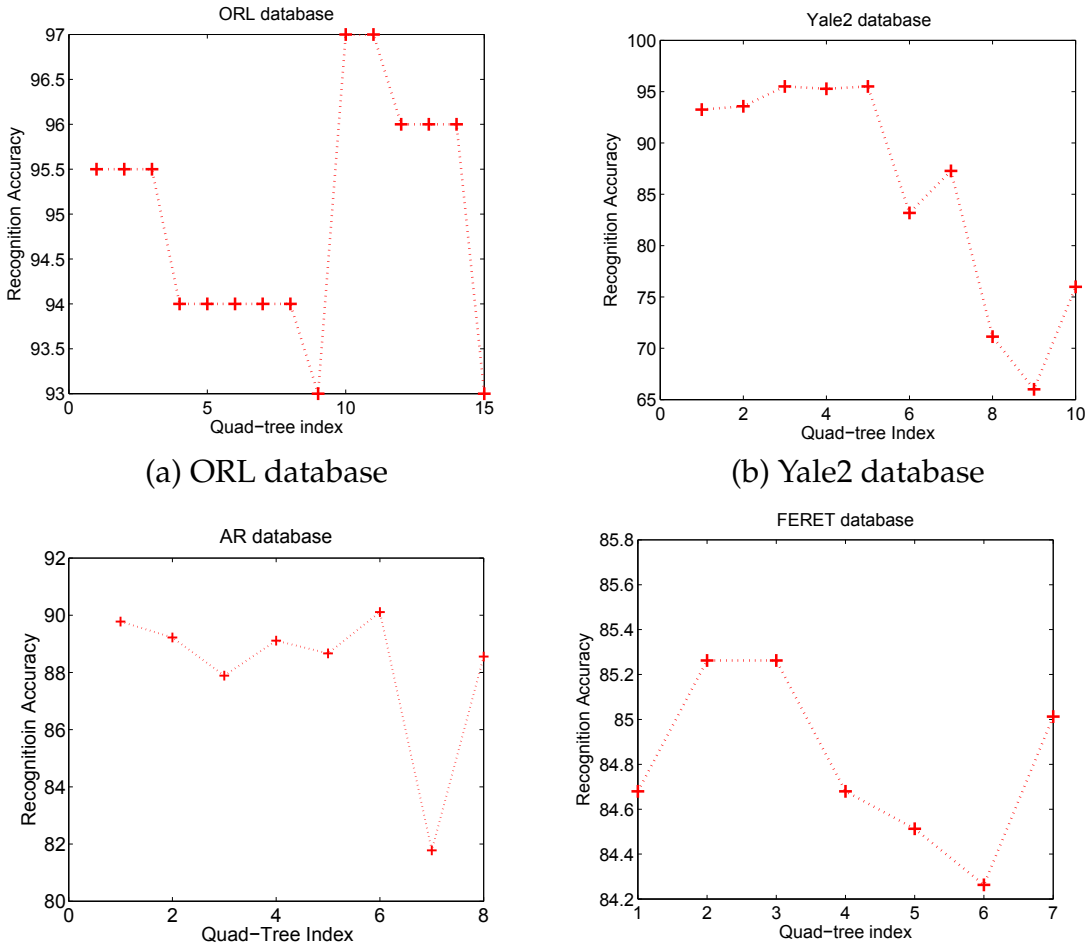
Method \ Database	ORL	Yale2	AR	FERET
PCA+LDA	92.70	76.30	83.22	59.62
MPCRC	91.50	92.11	88.60	73.64
30Regions	93.50	91.64	88.6	75.02
Data-Adaptive Block	96.00	91.86	89.56	70.28
deep_PCA	95.50	90.33	89.78	84.44
QT-Hierarchical	**97.00**	**92.94**	**90.44**	**85.43**

From Table 4.1, we have the following five observations:
Observation [1]: The conventional PCA+LDA method extracts a holistic feature representation from the image. It can preserve the global shape the images, but ignores the details such as local variations. It is effective in dealing with large variations such as head poses on ORL database. However, it has weak ability to cope with local deformations caused by illumination changes, and occlusions, facial expressions in Yale2, AR, and FERET databases. In contrast, local subregion based methods such as the 30region method, data-adaptive block based method, and the MPCRC can obtain a relatively higher performance on these three databases. This suggests the advantage of local based method in dealing with local deformations over holistic-based method.

Observation [2]: MPCRC method has a better performance than PCA+LDA on Yale2, AR, and FERET databases thanking to the collaborative of multi-scale patch based method. However, it performs not well on the ORL database, which has large variations caused by head poses. The main reason is that MPCRC is actually based on the local patches analysis, which has limited capability to investigate the global shape and structure of the image data. That is why we need to develop hierarchical learning algorithms to formulate good feature representation from local feature analysis.

Observation [3]: 30-Region method composes of 30 subregions which have large overlaps between local subregions. These subregions are manually designed by human beings after data registration. Among the 30 subregions, one "subregion" actually refers to the whole facial region. We can consider that this method is a brute-force integration of global feature and local feature representations. This method performs relatively stably on various databases, neither bad nor good. It performs well for well-registered databases such as FERET and Yale2, which mainly contain frontal face images. However, it has a low performance on not well registered databases, such as ORL database, which contains profile faces with different head poses.

Observation [4]: Deep PCA has a better performance than conventional PCA+LDA method on all databases, especially on FERET database, which contains the largest number of classes. This demonstrates the effectiveness of hierarchical learning over the shallow (flat) structure based method, which contains only a single layer. However, we also find that the Deep PCA method has a relatively lower performance than other methods, such as the data-adaptive block based method on the ORL database, MPCRC, 30-region method on Yale2



(a) ORL database (b) Yale2 database

(c) AR database (d) FERET database

Figure 4.6: Influence of Quad-Tree partitions to the deep_PCA based face recognition on four databases. (Copyrighted by CRC Press / Balkema (Taylor & Francis Group) [ZLM14b])

database. The reason might be that Deep PCA assumes the discriminative features are uniformly distributed. It employs image pyramid based hierarchical models and evenly aggregates features from lower level. The image pyramid is not adaptive and not flexible enough to investigate the image structure. This result verifies the importance of designing data-adaptive and flexible tree structure for hierarchical learning.

Observation [5]: Our method outperforms others in most experiments. Our hierarchical feature learning is developed based on a data-adaptive Quad-Tree structure. To explore how varied deep structures influence the performance of feature hierarchy, we introduced a set of thresholds for Quad-Tree partition. Fig. 4.6 plots the recognition accuracy of variant Quad-Tree partitions (the indices cor-

respond to different thresholds in ascending order). Please note that the leftmost point indicates the accuracy of the original Deep PCA, which is performed based on a complete tree structure. From this figure, we can find that hierarchical feature learning with different Quad-Tree partitions performs quite different. And the best performance usually is achieved at a certain tree index instead of the first one depending on databases. Our method benefits from the data-driven Quad-Tree partition, which can generate the most appropriate partition adapt to different databases automatically.

4.3.3 Comparison with Data-Adaptive Block based Algorithm

The relationship between the data-adaptive block based algorithm (Algorithm 1 in Chapter 3) and this algorithm (Algorithm 2) is as follows: Algorithm 1 is about feature learning from the planar image surface. After Quad-Tree partition, we can get data-adaptive blocks of variant sizes. We treat each block equally and assign the same weight to them for combination. In Algorithm 2, we find that the classification accuracy of each block is actually different from each other. Then, we assign different weights to different blocks according to the tree structure. Apart from this difference, the feature learning architecture is also different. In Algorithm 1, we just uses image planar surface (local blocks) for feature learning, while in Algorithm 2 we use a multi-layer architecture (defined by the Quad-Tree structure), which can learn more abstract features based on deep PCA.

We compare the result of these two algorithms in Table 4.1. We observe that this hierarchical learning algorithm outperforms the data-adaptive block based method for all databases, especially on FERET database, which is largest databases containing a large number of classes. These experimental results demonstrate the strength and effectiveness of utilizing hierarchical learning framework to generate global feature representation. This verifies that the global feature representation formed based on local feature analysis is preferable than holistic methods, which utilize a single-layer network to generate global features.

4.4 Summary

This chapter proposes a novel hierarchical feature learning algorithm using Quad-Tree structure of images. Quad-Tree partitions face images into flexible-blocks according to the density of discriminative power in the local regions, and

learns a data-driven hierarchical architecture that preserves the correlationship between local subregions. We can conclude that our hierarchical learning algorithm using Quad-Tree structure for weight assignment has more accurate description of image structure for recognition. In contrast to exiting works which just use the global feature representation for recognition, we involve all mid-level features in high-level applications. This strategy further improves the recognition accuracy. Extensive experimental results demonstrate that the whole framework outperforms diverse methods using global, local features and canonical deep structure learning algorithm "Deep PCA". Through comparison with the method developed in Chapter 3, we find our method has a better performance than the data-adaptive block based method without hierarchical learning.

Chapter 5

Feature Learning from Enlarged Training Data Encoded by Quad-Tree for Small Sample Size Problem

In chapters 3, 4, we have investigated feature learning with large training data. While exiting visual feature learning algorithms can work well when the training data is large enough, they are not well suited to learn good features from small training data. In order to evaluate the reliability of our method, this chapter investigates feature learning with small training data. The Small Sample Size (SSS) problem arises from the small number of training samples compared to the high dimensionality of the sample space. An extreme case of SSS is single sample per person (SSPP), where only one sample is available for each subject. Existing visual feature learning algorithms suffer from overfitting problem under SSS severely. Even worse, SSPP makes some feature learning methods failure, e.g. linear discriminant analysis (LDA) since we can not generate the within-class scatter matrix under SSPP. Thus, it is much necessary to investigate feature learning with small training data.

Ensemble learning has been acknowledged as an effective technique to alleviate the SSS problem. It benefits from base classifier collaboration, where the overall discriminative power is higher than a single classifier. However, existing ensemble methods are still challenged by two issues: (1) How to generate diverse base classifiers under small data; (2) How to alleviate the Diversity/Accuracy dilemma when involving inappropriate base classifiers in ensemble. To solve these two problems, this chapter develops a novel ensemble learning algorithm from enlarged training data encoded by Quad-Trees. This algorithm solves the

first problem by generating new samples encoded by Quad-Trees and deals with the second problem by selecting appropriate base classifiers for ensemble using a tailored 0–1 Knapsack solution. Our method is more robust and powerful then existing ensemble learning algorithms to deal with the SSS problem. We evaluated our method on the SSS face recognition using several standard databases. Extensive experimental results demonstrate the effectiveness of our method.

5.1 Problem Description and Related Works

5.1.1 Small Sample Size Problem

Face recognition has been widely applied to many applications, such as multimedia monitoring/surveillance, access control, biometric verification, human-computer interaction etc.. Existing appearance based face recognition algorithms such as PCA, LDA, Bayes Matching and their weighted, kernelized and tensorized variants [LPV08, LT10, Yan02] have been extensively exploited and acknowledged as one of the most well-known approaches. However, these methods usually assume that a large number of training samples are available for training purpose. Unfortunately, this assumption may not be met in many real-world applications such as forensic identification, surveillance photo verification, person re-identification in multi-camera networks etc., due to the difficulty in training sample collectionin. The small training data leads to the so-called small sample size problem [DLHG12, Yan12], where the number of training samples is much less than the high dimensionality of the sample space. We illustrate the SSS problem by the manifold surface demonstration in Fig. 5.1. In Fig. 5.1, the red manifold surface represents a face space projected by a large number of samples of a person (all the samples in this figure), while the blue surface denotes the face space generated using a very small number of samples (only three samples of that person). We can see that the estimated face space (in blue) is very likely to overfit to the small training data and can not reveal the actual face space (in red). Conventional object recognition / classification algorithms suffer from the SSS problem, which leads to performance degradation [LPJ11, PWB+12, BKJ13]. The extreme case of SSS is *Single Sample Per Person* (SSPP), where only a single sample is available for each subject in such systems as passport photo and ID card identification, law enhancement, etc. [LTW11].

The exploration of the SSS problem is very interesting to both theoreti-

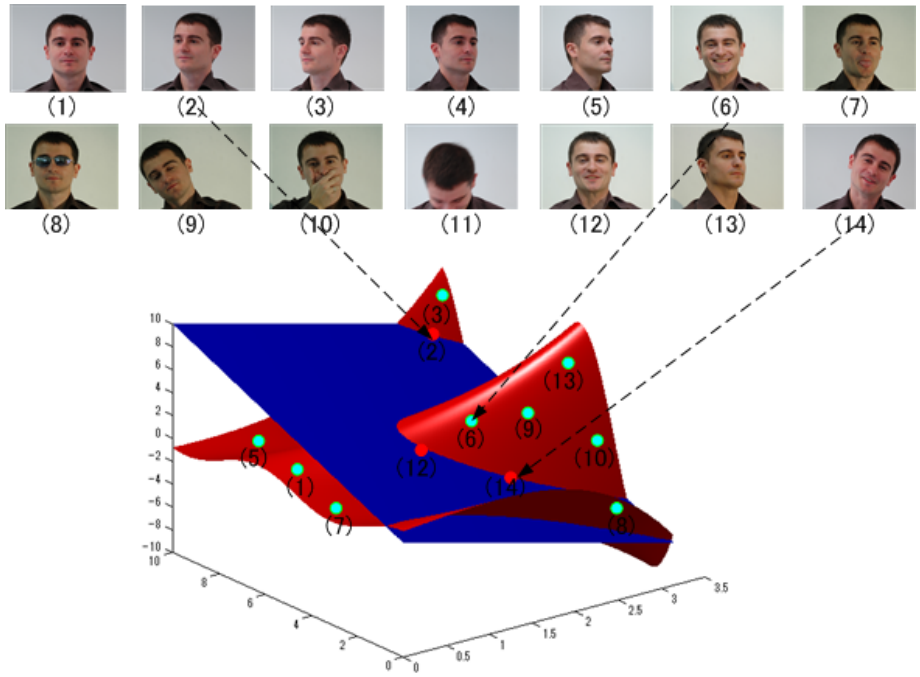


Figure 5.1: The demonstration of two manifold surfaces: the red one represents a full face space projected from all the samples of one person; and the blue one is a face space learned when only three samples are available. (Copyrighted by IET [ZLM13])

cians and practitioners, because it can easily contaminate the design and construction of a face recognition system. In the last two decades, much effort has been devoted to solving the SSS problem. Representative works include [WT06, SJZZ10, OPM+11, Spr11, ZZHS12, KSS+13, LTW13], and so on.

5.1.2 Ensemble Learning for SSS

There have been many attempts in the literature towards the SSS problem, such as the virtual sample generation based methods, multi-subregion combination methods [LTW13]. However, these methods still have limited generalization ability when the training samples are insufficient [ZZHS12]. Recently, a machine-learning technique known as ensemble learning has been recognized as an effective way in alleviating the SSS problem. The basic idea of ensemble learning is that a pool of weak classifiers can have a better classification ability than a strong classifiers under SSS. It generates multiple base classifiers, which offer complementary information for classification. Benefit from the base classifier collaboration, the overall discriminative power can be higher than a single classifier.

Existing ensemble learning algorithms for face recognition can be mainly

classified into three groups according to the type of base classifiers involved in ensemble: (1) global feature selection based on random subspace; (2) multi-subregion based local feature extraction; (3) global and local feature integration [ZLM13, ZLM14a].

The first kind of methods introduce *random subspace* for ensemble. The introduction of random subspace is based on the literature finding that strong and stable base classifiers defined by subspace methods (e.g. PCA, LDA) are not suitable to ensemble rules [SD02]. To solve this problem, Random Subspace (RS) [WT06] was employed through random sampling on feature subspace to generate weak but diverse base classifiers. As RS just focuses on the global rather than local features, local discriminant information can not be guaranteed. Motivated from this observation, the second kinds of methods utilize local feature extraction based on multi-subregions, which are named as multi-subregion methods [ZZHS12]. The first step of these methods is to partition the whole face region to multiple local subregions. An early attempt [MK01] partitioned each face image into six elliptical sub-regions according to the prior-knowledge of human beings. Since it is difficult to model the feature distribution on the whole face region by utilizing separated local subregions such as nose, mouth, and eyes [LTW11], Topcu et al. [TE10] developed an alternative way by partitioning face regions into small patches of the same size. Base classifiers can be trained from each patch separately and the final recognition results were obtained based on the decision fusion rule of base classifiers. This algorithm also has its problems. As patches are usually very small, they have limited ability in dealing with large variations. Considering that both global and local features can provide complementary information, the third category of methods integrate both global and local features together for classification. For example, Su et al. [SSCG09] proposed a hierarchical face recognition algorithm, where local features were extracted based on LDA and the global feature was extracted from whole face image using low frequency Fourier coefficients. Zhu et al. [ZZHS12] developed a Multi-scale Patch-based Collaborative Representation (MPCRC), which integrates the complementary information obtained at variant scales of patches. Spreeuwers et al. [Spr11] introduced a 30 region method. This method defined 30 regions of variant sizes according to experimental experience. The largest one of the 30 regions covers almost the whole face region.

Apart from the SSS problem, there are also some special attempts in dealing with the SSPP problem. Unsupervised learning methods employed variant exten-

sions of the classical PCA to solve the SSPP problem, such as 2DPCA [YZFY04], (PC)2A [WZ02], and (2D)2PCA [DZZP10]. Virtual sample generation methods such as SVD-LDA [ZCZ05] generate some virtual samples for each person so that LDA can be used for feature extraction. Generic learning methods such as Adaptive Generic Learning (AGL) [SSCG10] and Adaptive Discriminant Analysis (ADA) [KSS+13] generate an auxiliary training set from existing face databases to learn assist the recognition model to identify the people with a single sample. Patch (block) based methods such as Block PCA [GA04], Block LDA [CLZ04], and Discriminative Multi-Manifold Analysis (DMMA) [LTW13] aimed to learn base classifiers from local subregions to form a final classifier.

5.2 Motivation

An ensemble system has two important parameters, which relates to its performance: accuracy and diversity. The accuracy refers to the final classification accuracy of an ensemble system and diversity measures the disagreement degree in the output of multiple base classifiers [TSY06]. While high accuracy is the objective of most ensemble systems, diversity also plays an existential role in developing ensemble systems. Intuitively, a set of identical base classifiers cannot lead to a clear accuracy gain in ensemble. The ideal situation is that a set of base classifiers can bear different errors and the combination of them can minimize the final classification error. Existing ensemble methods can not generate diverse base classifiers from insufficient training data. Base classifiers learned from insufficient training data are very likely to be tightly correlated and bear similar errors. Having less diverse base classifiers has become a bottleneck of many existing ensemble methods.

In order to make base classifiers more diverse, data augmentation is employed in many approaches before ensemble learning. Generic learning (GL) is considered as an effective technique for data augmentation. Conventional GL approaches utilize an auxiliary set (named as generic set), which contains a large number of samples, to assist in ensemble learning. This idea is based on an intuitive observation that faces of all human beings look alike. This implies that different people may have similar faces. These methods try to grab some similar face images from the generic set and put into the small training dataset to augment the training samples in the small dataset. However, no matter how similar these face images are, they actually do not belong to the small dataset. These

methods are very likely to make wrong decisions for the subjects, who actually belong to the generic set other than the small dataset.

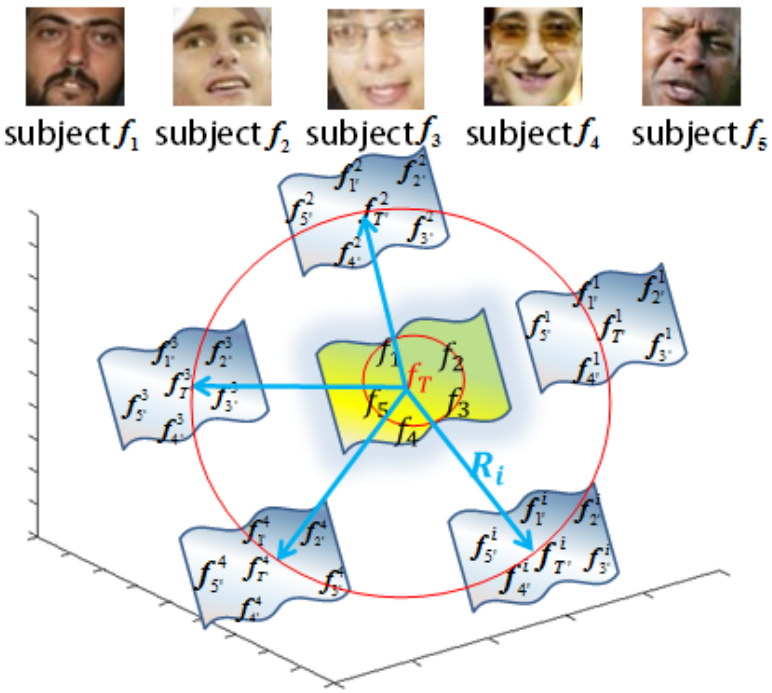


Figure 5.2: Face space expansion by generating new samples. This figure is from author's publication [ZLM14a] in an open access journal: Sensors.

To overcome the problem of conventional GL methods, this chapter develops a novel small data augmentation method by generating new samples encoded by Quad-Tree other than grabbing samples from the generic set directly [ZLM13, ZLM14a]. The novel idea of this work is illustrated in Fig. 5.2. The original small training sample set locates at the center of the face space in the figure. To explore new possibilities of face space, we introduce a set of random matrices $R_i, i = 1, 2, ..., L$, the elements of which are generated based on uniform distribution. Random matrices are not added to the original face images directly. Instead, we first model the feature distribution of the a small sample set by a template image f_T and then R_i is added to f_T to transform it to a new place $f^i_{T'}$. After that, Quad-Tree decomposition [ZLM12] is performed on each $f^i_{T'}$ to generate an image encoding pattern. Original face images are re-organized according to each Quad-Tree partition to generate a new training sample set. For each random matrix, we can have a Quad-Tree partition, and generate a new training sample set. Given N random matrices, we can have N Quad-Tree partitions, and N new training

sample set accordingly. Base classifier are learned from each new training set for ensemble. Since the new samples are generated using the original face images according to the feature distribution in the small sample set, these new samples should locate around the original small sample set. Our data augmentation strategy expands the face space to a large extent.

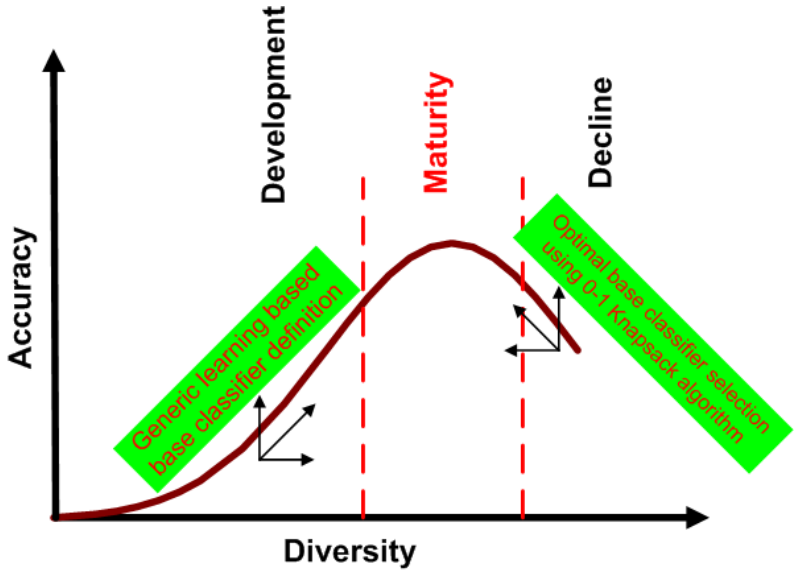


Figure 5.3: The relationship between diversity and accuracy, which is summarized into three stages: development, maturity, and decline [ZLM14a]. (This figure is plotted based on the experimental results shown in Fig. 5.8)

Even having diverse base classifiers, we still need to consider another problem. Ensemble learning algorithms aim to achieve higher accuracy and higher diversity meanwhile. It is difficult to achieve these two objectives at the same time. Since some base classifiers learned from generated samples may have are not appropriate for classification. Integrating such inappropriate base classifiers may reduce the accuracy of an ensemble system, even the diversity can be improved. Figure 5.3 illustrates the relationship between accuracy and diversity, where the growth of accuracy and diversity is classified into three stages: *development*, *maturity*, and *decline*. In the development stage, the accuracy grows up with the increase of diversity. It achieves the highest value at the maturity stage, and then decreases with the increase of diversity in the decline stage. We can see that the accuracy and diversity are not linearly related. The increase of diversity can improve the accuracy in a certainty degree, but not always so. This observation is known as the Diversity/Accuracy dilemma [TSY06]. It is a very challenging problem to existing ensemble methods. Many algorithms were proposed to allit-

erate this problem, but their results are not satisfactory. In this work, we find that the key point in addressing the Diversity/Accuracy dilemma is to get the trad-off between accuracy and diversity. In this chapter, we convert the trad-off investigation problem as an optimization problem, which aims at selecting an optimal subset of base classifiers for ensemble. Motivated by this finding, we introduce an amended 0–1 Knapsack solution for the optimal base classifier selection.

The main contribution of this work can be summarized as follows: in this chapter, we develop a novel Quad-Tree based ensemble framework (QT-E), which can argument the small training data by generating news samples encoded by Quad-Trees during the development stage and alleviate the Diversity/Accuracy dilemma by introducing a tailored 0–1 Knapsack solution [ZLM12]. Compared to existing algorithms [WT06, SJZZ10, OPM+11, Spr11, ZZHS12, KSS+13, LTW13], this framework is characterized by the following items:

- QT-E augments the small training data by generating new samples encoded by Quad-Trees. More diverse base classifiers can be generated from the expanded training sample space than existing ensemble methods;
- Conventional ensemble algorithms integrate all base classifiers without selection, which may lead to the Diversity/Accuracy dilemma. Our framework selects just appropriate base classifiers for ensemble based on an optimal solution. This strategy alleviates the Diversity/Accuracy dilemma.
- While conventional multi-subregion based ensemble methods partition face images into small patches without considering the correlation between subregions, the Quad-Tree structure partitions the face image into overlapping regions and preserves the geometric correlation among subregions.

5.3 Algorithm Architecture

The developed Quad-Tree based ensemble framework QT-E is illustrated in Fig. 5.4, which involves the following three steps:

- **Base classifier definition after data expansion:** the training data is expanded by generating new samples encoded by Quad-Trees. After data expansion, more diverse base classifiers can be generated subsequently.
- **Base classifier selection using an optimal solution:** in order to solve the Diversity/Accuracy dilemma, we employ an optimal solution motivated

from a tailored 0–1 Knapsack problem, which can select just appropriate base classifiers for ensemble.

- **Base classifier integration through majority voting:** the selected base classifiers can be integrated based on a majority voting.

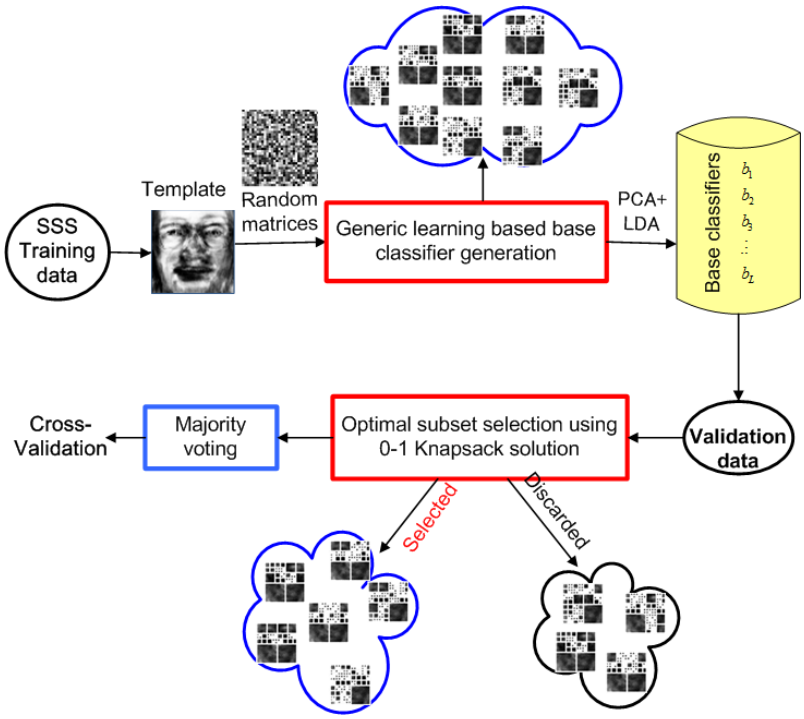


Figure 5.4: Illustration of Quad-Tree based ensemble framework (QT-E) [ZLM14a].

5.3.1 Quad-Tree based Image Data Expansion

The procedure of base classifier definition is illustrated in Fig. 5.5, which consists of three operations: (1) template image generation and random matrix introduction; (2) new sample generation based on Quad-Tree decomposition, and (3) the definition of accuracy and diversity of base classifiers.

Template Image Generation

The template image is generated in the same way as that in chapter 3. Motivated by the idea of linear discriminant analysis (LDA), which encodes discriminant information by maximizing the between-class scatter matrix S_b and minimizing the within-class scatter matrix S_w (See Eq. (5.1)), we define a template face f_T by

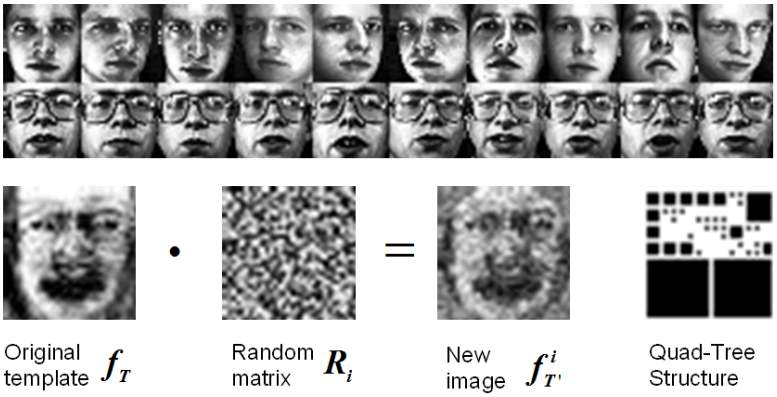


Figure 5.5: Example of template face generation and random matrix introduction on ORL database [ZLM14a].

the diagonal vector of $\frac{S_b}{S_w}$ (see Eq. (5.2)). The entries of f_T represent the variances at each pixel of the face region across all the images in a database.

$$\begin{cases} S_b = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T, \\ S_w = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T, \end{cases} \tag{5.1}$$

$$f_T = diag(\frac{S_b}{S_w}), \tag{5.2}$$

where c is the number of classes in the dataset, μ denotes the mean face of all face classes, μ_i is the mean face of class X_i, N_i denotes the sample number of class X_i, and x_k represents the k-th sample of class X_i. For the face images of size $m \times n$, the covariance matrix $\frac{S_b}{S_w}$ is of size $(m \times n)^2$. And f_T if of the same size as the original face images. Please note that the generation of f_T using Eq. (5.2) fails under SSPP since we can not construct S_w using just one ample per person. In this case, f_T is defined by the diagonal vector of S_b under SSPP.

Since the template image f_T is generated from the small training data, it is of weak ability to represent the whole face space. To solve this problem, we introduce a set of random matrices $R = \{R_1, R_2, ..., R_i, ..., R_L\}$ and add them to f_T to generate several new template images $f_{T'}^i = \{f_{T'}^1, f_{T'}^2, ..., f_{T'}^i, ...f_{T'}^L\}$. Here, each random matrix R_i, $i = 1, 2, ..., L$ if of the same size as f_T and its elements are randomly generated based on a uniform distribution in [0, 1]. The new image $f_{T'}^i$ is

generated by the dot product of f_T and R_i as:

$$f^i_{T'} = f_T \cdot R_i. \tag{5.3}$$

Quad-Tree Decomposition

Quad-Tree decomposition is performed on each $f^i_{T'}$ to find local regions of high density of discriminant features across the face area. The decomposition is performed based on a criterion function $doSplit(r)$ defined by Eq. (5.4) [ZLM12]. If the variance of a region r (r starts from the whole region of $f^i_{T'}$) is higher than or equal to a threshold variance (t_v), then this region is split into four sub-blocks of equal size. The partition carries on until no blocks satisfy the criterion function.

$$doSplit(r) = \begin{cases} true, while(var(r) \geq t_v), \\ false, otherwise, \end{cases} \tag{5.4}$$

The image $f^i_{T'}$ is partitioned into less and larger blocks for a large t_v, but into more and smaller blocks for a small t_v. In this chapter, the random matrix is the main variations introduced to face images, which influence the feature distribution over the image region. We had better keep t_v as an invariant parameter in order for better investigation the influence of the random matrix to the face region partition. In this dissertation, we define $t_v = 0.5 * var(wholeR)$ without loss of generality, where $wholeR$ denotes the whole region of $f^i_{T'}$.

Each Quad-Tree partition refers to a face encoding pattern (depicted by blocks of variant sizes) as illustrated in Fig. 5.6 (a). Given L random matrices, we can have L encoding patterns. For each Quad-Tree partition, we re-organize the face images in the original image dataset according to the partition result. The image re-organization is performed in the same way as that in Chapter 3. Since larger partitions (blocks) mean that the density of discriminative features in them is low, then we donot need to keep its original size. We will do down-sampling on large blocks (the size is larger than the smallest size) by resizing them to a smaller size $((d/2) \times (d/2))$, where d is the width of large blocks (in pixel). For smallest blocks mean they the density of their discriminant features are high. Thus, we will not do down-sampling on them. Finally, block resizing result in a set of new images whose sizes are smaller than original face images. For L Quad-Tree partitions, we can have L new training sample set. Then, we train a base classifier b_i from each new training sample set using PCA+LDA.

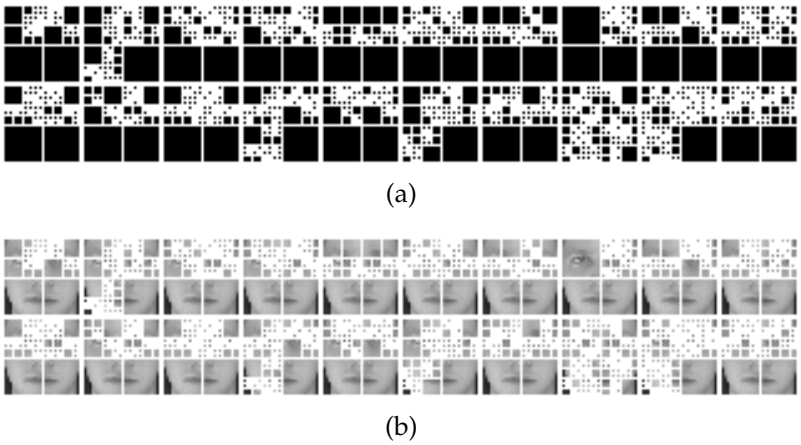


Figure 5.6: An example of 20 Quad-Trees (a) Quad-Trees (b) Quad-Tree partitions on a face image. (Copyrighted by IET [ZLM13])

Figure 5.7: Example periocular images from two different subjects [PJR11]: (a), (b) without eyebrows and (c), (d) with eyebrows.

We have a very interesting observation that our Quad-Tree partitions can prove an important finding in biometric recognition field. In the recent years, researchers find that the periocular region (the area around eyes) is a very important region for biometric recognition [PJR11]. It even has better performance than the whole face region. The advantages involve these tree main aspects: (1) the periocular region is less affected by local deformations such as facial expressions, aging changes, and the variations caused by the growth of male facial hair than the whole face recognition; (2) this region is more robust to head pose changes than the whole face region. It remains effective in case of extreme head pose changes where the full face based recognition may fail. (3) this region is more reliable than the full face region when most of the lower face region is occluded and when only one eye is visible. These finding are observed based on experimental results. However, existing works cannot give specific reasons on why the

periocular region has such advantages. In contrast, our Quad-Tree (see Fig. 5.6) can prove these findings by demonstrating that the periocular region has higher density of discriminate information than the other regions of the face.

5.3.2 Base Classifier Definition and Calculation

The performance of an ensemble system E is mainly related to two important parameters: (1) the accuracy ($acc(E)$); and (2) the diversity ($div(E)$). We also use these two parameters to evaluate base classifiers. The accuracy of each base classifier b_i is defined by the ratio of correctly classified samples against the total number of samples in Eq. (5.5). And the accuracy of an ensemble $acc(E)$ is defined by this ratio based on the majority voting of a set of base classifiers in Eq. (5.6).

$$acc(b_i) = \frac{num(correctSamples(b_i))}{num(totalSamples)}, \tag{5.5}$$

$$acc(E) = acc(majorityVote(b_i)), i = 1, 2, ..., L. \tag{5.6}$$

The measurement of diversity ($div(E)$) is investigated in many literature works. For example, six commonly used measurements are employed in [TSY06], namely *disagreement measure, KW variance, inter-rater agreement, double fault measure, measure of difficulty* and *generalized diversity*. These measurements are designed for different recognition/classification problems. None of them can be recognized as universal for all applications. In this chapter, we use the first one: the *disagreement measurement* to calculate $div(E)$ since it was originally designed for ensemble problem [Aro05] and its intuition was coincide with ours, which suggest that two diverse base classifiers should perform differently on the same training data. The specific definition of disagreement degree is as follows:
For a test sample set, which contains N samples, each base classifier b_i, $1 \leq j \leq L$ predict a label for each test sample x_i, $1 \leq i \leq N$. The diversity between two base classifiers b_i and b_j is defined by the number of samples on which they have different labels against the total number of samples:

$$div_{i,j} = \frac{n(a,b)}{N}, a \neq b \tag{5.7}$$

where a and b denote the labels assigned by b_i and b_j, respectively; $n(a,b), a \neq b$ denotes the number of samples, on which b_i and b_j have different decisions.

Diversity among all base classifiers is calculated by the average of all pairwise diversities in Eq. (5.8). The total diversity measures the total disagreement among all base classifiers.

$$div(E) = \frac{2}{L(L-1)} \sum_{i=1}^{L} \sum_{j=1}^{L} div_{ij}. \tag{5.8}$$

5.3.3 Base Classifier Selection and Ensemble

As mentioned in the motivation, not all generated new samples are within the face space as we expected. Even the samples within the face space may perform differently according to the discriminant features involved in them. That is, some base classifiers learned from the expanded training data may not be appropriate for classification. Integration of such inappropriate base classifiers is not even to say improve the final recognition accuracy but may reduce the performance instead, which leads to the Diversity/Accuracy dilemma as illustrated in Fig. 5.8.

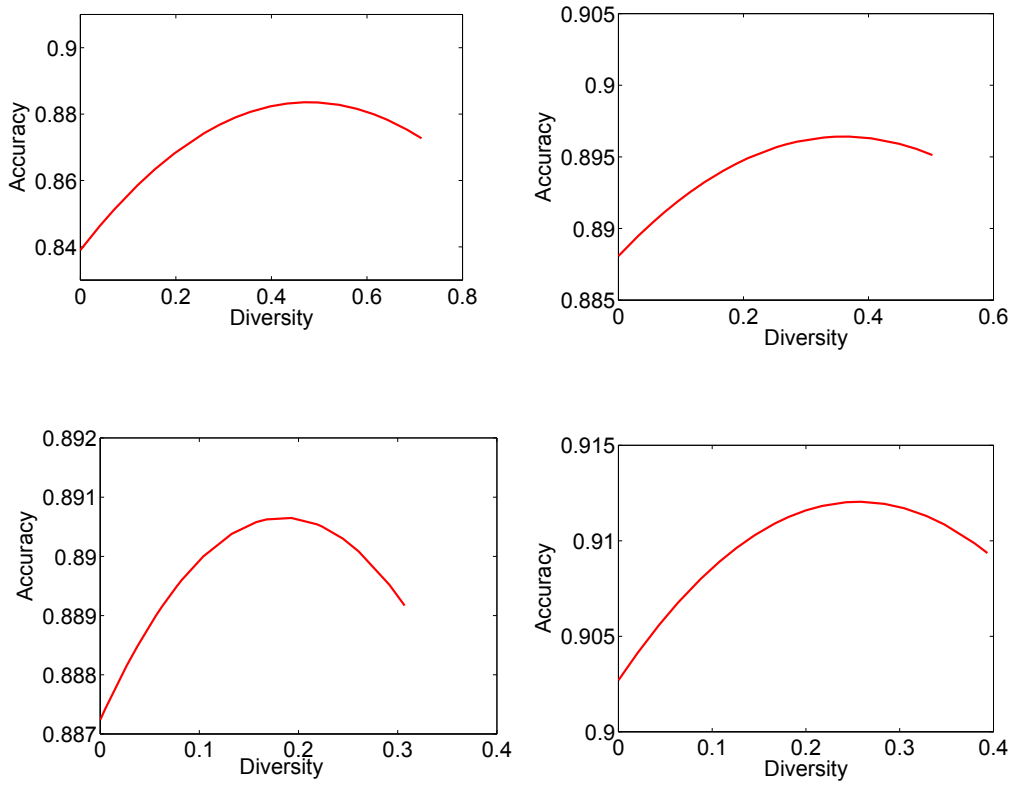


Figure 5.8: A demonstration of Diversity/Accuracy dilemma on four public face databases [ZLM14a]: (a) ORL, (b) Yale2, (c) AR, and (d) FERET.

To solve this problem, we develop a base classifier selection algorithm using

a tailored 0–1 Knapsack solution. The selection algorithm can be considered as on optimal combinatorial task. The conventional 0–1 Knapsack problem is depicted as follows: we have a Knapsack K, which has its own capacity. Given a set of items, each item has two parameters: a mass and a value. We would like to select some items and put into the Knapsack so that the total value of K can be maximized while the total weight is still not beyond the capacity of K. The mathematical interpretation is as:

$$\begin{aligned} &max(value(K)), \\ &subject\ to \quad mass(K) \leq t_c, \end{aligned} \tag{5.9}$$

where t_c denotes the capacity of K; $value(K)$ and $mass(K)$ represent the total value and total mass of selected items in K, respectively.

This is very similar to our problem. We have a set of base classifiers. Each base classifier are with two parameters: (1) recognition accuracy, and (2) the diversity with other base classifiers in making decisions. Our goal is to find an optimal subset of base classifiers E which can maximize the final accuracy of E and the total diversity should be still higher than or equal to a threshold diversity t_d. Our problem can be interpreted in a mathematical way as:

$$\begin{aligned} &max(acc(E)), \\ &subject\ to \quad D(E) \geq t_d, \end{aligned} \tag{5.10}$$

where t_d is the diversity threshold and $D(E) = div(E)$ denotes the diversity of E.

Comparing the optimization functions (5.10) and (5.9), we can see that these two problems are very similar. The $acc(E)$ and $D(E)$ in our problem are much alike the $value(K)$ and $mass(K)$ in the conventional 0–1 Knapsack problem.

However, there are also some differences between these two problems. The first different is in the constraint interpretation. While the conventional Knapsack problem requires the total weight to be *less than or equal to* the capacity and our problem requires the diversity of ensemble to be *higher than or equal to* T_d. To make our problem be more coincide with the conventional 0–1 Knapsack problem, we modify the constraint as: the *inverse* of the total diversity of E should be *less than or equal to* the *inverse* of a diversity threshold t_d.

Another difference is that the conventional Knapsack problem assumes items are different in mass or value, and thus we can have a unique optimal solution. However, in our problem, base classifiers may have the same accuracy and di-

versity at the same time. That means, multiple subsets of base classifiers may get the optimal solution meanwhile. Under such cases, we add another constraint by defining the complexity of ensemble (depreciated by the number of base classifiers) and select the subset of the least number of base classifiers as the final optimal solution.

Finally, our problem can be formulated as:

$$\begin{aligned} & max(acc(E)), \\ & subject\ to \quad \frac{1}{D(E)} \leq \frac{1}{t_d}, \\ & min(num(E_1), num(E_2)), where\ acc(E_1) = acc(E_2), \end{aligned} \tag{5.11}$$

where $num(E)$ is the number of base classifiers in E. We define the diversity threshold t_d as such a diversity, at which the final recognition accuracy of an ensemble achieves at its highest value. As illustrated in Fig. 5.8, such t_d has different values at different databases. It should be defined according to both the application requirement and data property. For applications where high diversity is preferred, we need to assign a relatively high value to t_d (e.g. 0.9). And for other applications, we can assign a relatively low value. In this chapter, in order to make the definition of t_d adapts to databases, we calculate t_d as the average value of such t_ds on four databases in Fig. 5.8.

After base classifier selection, we need to integrate the classification results of selected base classifiers $B' = \{b_1, b_2, ..., b_{L'}\}$ together. Here, we use majority voting as the integration scheme. For each test sample, we assign the class label that receives the largest vote to it.

5.3.4 Algorithm Complexity Analysis

The skeleton of the proposed ensemble system is demonstrated in Algorithm 1:

The computational complexity of the whole framework is analyzed according to the tree main processing stages: Generic learning based base classifier definition, optimal base classifier selection, and majority voting . The first stage involves three sub-procedures: template image calculation, Quad-Tree decomposition, and the performance analysis of base classifiers based on PCA+LDA. For a given training sample set containing m samples, the size of each face image is d. The generation of original template image f_T, new template image $f_{T'}^i$ and Quad-Tree decomposition can be performed in linear time $O(d)$. The clas-

Algorithm: 1 Feature learning from extended training data encoded by Quad-Trees

Input: Gallery set $X^{\mathbb{G}} = \{x_{ij}^{\mathbb{G}}; i = 1, 2, ..., N, j = 1, 2, ..., M_G\}$, probe set $X^{\mathbb{P}} = \{x_i^{\mathbb{P}}; i = 1, 2, ..., N * M_P\}$, parameter L.

Output: Decision vector of $X^{\mathbb{P}}$: $Y^{\mathbb{P}} = \{y_i^{\mathbb{P}}; i = 1, 2, ..., N * M_P\}$.

Initialization:

Set $y_i^{\mathbb{P}} = 0; i = 1, 2, ..., N * M_P$.

Step 1: Base classifier definition based on training data expansion using Quad-Tree

Generate a template image f_T using Eq. (5.1).

For $r = 1, 2, ..., L$, repeat

1.1: Add a random matrix R_r to f_T to get a new template image $f_{T'}^r$ defined by Eq. (5.3);

1.2: Perform Quad-Tree decomposition on $f_{T'}^r$ to get an encoding pattern P_r by Eq. (5.4) and utilize P_r to generate a new gallery dataset $X_r^{\mathbb{G}}$;

1.3: Perform PCA+LDA on $X_r^{\mathbb{G}}$ to define a base classifier b_i, the output decision vector of b_r is Y_r, the accuracy of b_r is denoted by $acc(b_r)$.

end

Step 2: Optimal base classifier selection:

Select a subset of base classifiers to compose the ensemble E using the optimal solution in Eq. (5.11).

Step 3: Majority voting:

Integrate the selected base classifiers in E using majority voting.

Output decision vector:

Output the labels of samples in the probe set $Y = max(\sum Y_r)$.

sification algorithm PCA requires $O(d^3 + d^2m)$ computations and LDA needs $O(mnt + t^3)$ [CHH+07], where n is the number of features and $t = min(m, n)$. Since the dimension of feature subspace is usually smaller than that of the original face image, we have $d > n$. For an ensemble system, which contains L base classifiers, the complexity of the first stage is summarized as $O(L(d^3 + d^2m))$. The second stage is the base classifier selection using a tailored 0–1 Knapsack problem, which can be solved by dynamic programming requiring $O(LlogL)$ computations. The third stage based on majority voting takes $O(L)$ computations. Therefore, the total computational complexity of our Quad-Tree based ensemble system is $O(L(d^3 + d^2m + LogL + 1))$.

5.4 Experimental Evaluation

To evaluate the developed system, we use four standard face recognition datasets, namely: ORL [SH94], Extended Yale (Yale2) [GBK01], AR [MB98], and FERET [PMRR00]. In this section, we first analyze the influence of one key parameter to our method: the number of base classifiers. Then, we evaluated our method by comparing with a large number of face recognition algorithms on both SSS problem and SSPP problem. Finally, we tested the performance of the base classifier selection algorithm. For the better comparison of our method with existing methods, we conduct the SSS problem with different number of training samples per subject. For the training and test data partition, we randomly select p samples for each subject as training data and the rest samples as test data. For each database, we perform 10 splits in this way. For each split, we use k-fold cross validation ($k = 5$) for the evaluation of perform with and without base classifier selection, respectively. Let me use an example for explanation. In Fig.5.9, we can see that both the training data and test data are divided into k subsets. These subsets have the same number of subjects. The holdout validation is performed k times. For each time, one of the k subsets is selected as training data and validation data. We use them to train the base classifier selection. The other $k-1$ subsets are used as training data and test data, which are used to report the performance of our method with and without base classifier selection (denoted by *'Our-Sel.'* and *'Our-Org.'*), respectively. The final recognition accuracy of our method is reported by averaging over $10 * k$ trials(10 random splits by k folds-cross validation). In our method, we use PCA+LDA for feature extraction and use the nearest neighbor classification with L_2-norm for matching. For each databases, we define the feature dimension of PCA+LDA subspace as $subjectNumber - 1$.

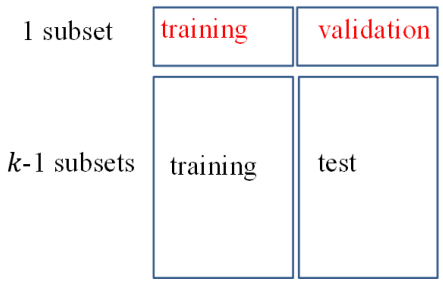


Figure 5.9: k-fold cross validation for the perform evaluation.

5.4.1 Datasets

- **ORL database**: this database contains 40 subjects, each subject has 10 samples. Face images vary at facial expressions, head poses, open or closed eyes, with glasses or without glasses and have some scale changes (up to about 10 percent).
- **Extended Yale (Yale2) database**: this database contains more than 20,000 single light source images of 38 subjects which are taken under 576 viewing conditions (9 poses in 64 illumination conditions). In order to evaluate the robustness of our method to illumination changes, we use 64 near frontal images per subject under different illuminations for the experiments.
- **AR database**: this database contains more than 4,000 color face images of 126 people (70 men and 56 women). Each subject has 26 samples, which are taken in two sessions (separated by two weeks). Each session contains 13 samples, which are vary on facial expressions, illumination conditions, and occlusions (with sunglasses and scarves). Similar to the comparison method [LTW13], we select eight subsets (A to H) of 800 images (8 images per subject) from 100 different subjects (50 men and 50 women) for the experiments. These samples are taken from two separate sessions and with different expressions. Figure 5.10 (c) illustrates sample images from Subsets A to H, where subset A is used for training and the remaining seven subsets for test.
- **FERET database**: this database contains 13,539 facial images from 1,565 subjects. They are vary on facial expression, gender, ethnicity, and age. We use the grayscale images from GrayFERET (FERET Database, 2001). We use two subsets (FERET-1 and FERET-2) to evaluate the performance of our method on SSS and SSPP problem, respectively. Similar to the comparison method [OPM$^+$11], FERET-1 contains the images of all available subjects that have more than 4 frontal images. There are totally 665 face images of 82 subjects in this subset. Similar to the existing work [LTW13], we use FERET-2 for SSPP problem, which contains 400 frontal face images of 200 persons (71 women and 129 men). Each subject has two images (labeled by *Fa* and *Fb*), which are varying on expressions, illuminations, races, genders, ages, and scales, etc..

Face images in the first three databases (ORL, Yale2, and AR) are aligned to size 32×32 using the method in [CHH$^+$07]. Face images in the FERET database are

Figure 5.10: Four databases used in the experiments: (a) ORL, (b) Extended Yale (Yale2), (c) AR, (d) FERET.

normalized to the same size using the *CSU Face Identification Evaluation System 5.1* [CSU03]. Sample images of these four databases after histogram equalization are shown in Fig. 5.10.

5.4.2 Parameter analysis

We first investigate the influence of a key parameter in our method: the number of base classifiers L. We use a relatively large database as Yale2 database ($p = 5$) and a relatively small database as ORL ($p = 2$) for analysis. The recognition accuracy of our method on these two databases is illustrated in Figs. 5.11 (a) and (b), respectively, where L ranges from 20 to 50. We can see that QT-E has a stable performance for a wide range of base classifier numbers on these two databases. In our method, in order to save computational cost, we set L to a relatively small value as 20 without performance lost.

5.4.3 Experimental Results

We compared our method with several existing face recognition algorithms on SSS problem, including: conventional face recognition algorithms without ensemble and several existing ensemble methods [WT06, SJZZ10, OPM$^+$11, Spr11, ZZHS12, KSS$^+$13, LTW13]. The results on ORL, Yale2, and FERET-1 databases are shown in Tables 5.1–5.3, respectively. In these tables, the fist column are the comparison methods while the rest columns report the rank-one recognition accuracy of comparison methods using p samples per subject as training data. The ROC curves of our method on all databases are shown in Fig. 5.12 for better illustration.

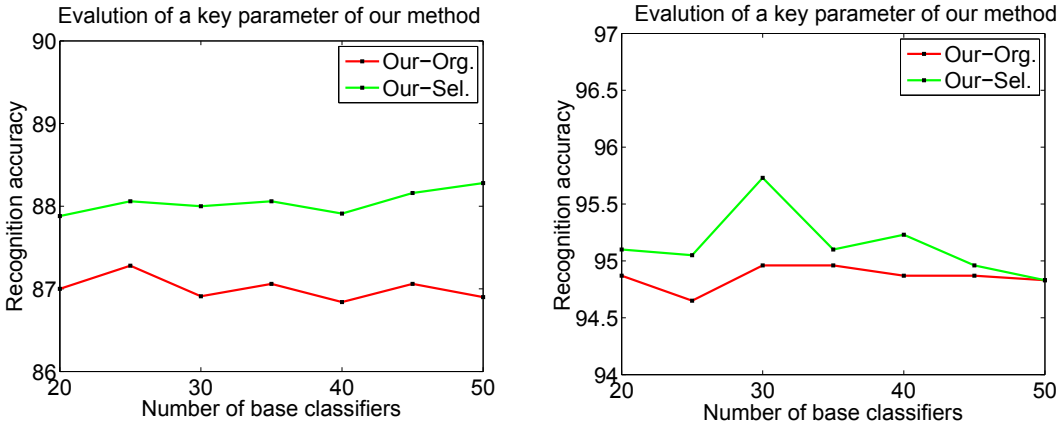


Figure 5.11: Influence of a key parameter (base classifier number) to our method on two databases [ZLM14a]: (a) ORL, (b) Yale2.

Comparison with conventional face recognition algorithms without ensemble

The comparison methods without ensemble include: PCA family [OPM+11], LDA family [CHH+07], LPP family (Locality Preserving Projections) [CHH+07], CCA family (canonical correlation analysis) [SJZZ10], and several other representative methods such as SVM, neural network based methods (MLP (multilayer perceptron) and RBF (radial basis function network)) [OPM+11]. For fair comparison, we reported their performances presented in existing works. As the existing works did not implement these methods on all four databases, we compare with different methods on different databases.

From Tables 5.1–5.3, we got the following five observations:

(1) **Performance reduces on SSS problem:** We find that conventional face recognition methods obtain relatively high performance when the gallery data are moderate or large. However, their performances reduces as the training samples per subject (p) decreases on all databases. This implies that existing face recognition algorithms are challenged by the SSS problem.

(2) **Supervised learning outperforms unsupervised learning:** The LDA based supervised learning methods performs better than the PCA based unsupervised learning methods in almost all experiments. This demonstrates the superiority of applying supervised learning algorithms in dealing with the SSS problem.

(3) **Global and local based methods have their both advantages and disadvantages:** Local feature based method (the classical LPP) outperforms the global based method (the classical LDA) on ORL database, which contains several vari-

Table 5.1: Evaluation on the ORL database

Method		p=2	p=3	p=4	p=5
PCA		66.9	76.6	82.1	86.3
LDA family	LDA	72.5	84.0	89.4	92.8
	PCA+LDA	77.7	86.1	90	92.7
	R-LDA	79.1	89.0	93.7	96.4
	S-LDA	82.9	91.9	95.9	97.7
CCA family	PCA+CCA	81.3	87.5	89.2	91.8
	CCA+Perturbation	81.3	87.8	89.5	92.7
	KPCA+CCA	81.5	88.8	92.8	93.5
	2DCCA	85.0	89.5	93.3	95.0
LPP family	LPP	78.0	86.2	90.3	93.2
	R-LPP	79.1	89.1	93.6	96.4
	S-LPP	82.9	91.9	95.9	97.7
	OLPP	79.5	89.2	93.6	96.2
Ensemble	RS	76.8	83.6	87.8	93.3
	PCRC	70.6	81.8	87.9	89.5
	MPCRC	78.4	84.3	87.1	91.5
	30Region	80.6	87.8	90.7	94.8
	Data-Adaptive Block	79.1	87.5	93.3	96.0
Our	Org.	86.3	92.2	95.4	97.5
	Sel.	**87.1**	**93.1**	**96.2**	**98.1**

Table 5.2: Evaluation on the Yale2 database

Method		p=5	p=10
PCA		36.4	53.6
LDA family	LDA	75.5	87.5
	PCA+LDA	76.3	87.0
	R-LDA	77.2	89.6
CCA family	PCA+CCA	73.0	86.0
	CCA+Perturbation	73.0	87.0
	KPCA+CCA	75.0	88.0
	2DCCA	87.5	91.5
LPP family	LPP	67.9	81.5
	Tensor-LPP	71.7	82.9
	OLPP	71.6	83.7
Ensemble	RS	84.3	95.8
	PCRC	91.0	98.8
	MPCRC	92.8	**99.1**
	30Region	90.3	97.8
	Data-Adaptive Block	91.9	98.7
Our	Org.	95.0	98.8
	Sel.	**95.5**	98.9

Table 5.3: Evaluation on the FERET-1 database

Method		p=2	p=3	p=4
PCA family	PCA	82.4	86.6	89.4
	2DPCA	81.9	86.4	89.2
	KPCA	82.3	87.8	91.6
SVM family	SVM	68.8	91.7	95.1
	PCA+SVM	91.5	95.8	97.2
Neural Networks	MLP	72.9	83.4	85.9
	RBF	85.3	93.2	96.8
Ensemble	RS	75.7	81.5	85.7
	LBP-5 $\times$ 5	89.4	92.1	94.2
	LBP-7 $\times$ 7	91.5	94.4	96.0
	LBP-7 $\times$ 7w	**92.9**	95.1	96.6
	30Region	73.0	92.3	82.1
	Data-Adaptive Block	84.3	93.9	94.5
Our	Org.	85.1	96.9	96.4
	Sel.	91.9	**96.9**	**98.2**

ations caused by facial expressions, head poses, open/close eyes, head scale changes, with/without glasses etc. However, it is not so on the Yale2 database, which contains illumination variations. This suggests that both local and global feature based methods have their advantages and disadvantages. Feature representation is not the key issue in solving SSS.

(4) **Generating more diverse samples is important:** We have an interesting observation between 2DCCA (on Tables 5.1, 5.2) and 2DPCA (on Table 5.3). The method 2DCCA is an extension version of CCA algorithm targeting at SSS problem. It directly extracts features from image matrix other than from matrix to vector transformations. It consistently outperforms other versions of CCA on Tables 5.1 and 5.2. However, 2DPCA does not have performance gain on Table 5.3. This implies that the 2D-matrix based feature representation is not always effective in solving the SSS problem. The key point for SSS is to expand the training sample set by generating new samples other than changing the feature representation.

(5) **Our method outperforms others:** Our method obtains higher performance in all experiments compared to all conventional methods without ensemble. This thanks to two main reasons: (1) the original training sample set can be enlarged effectively by generating new samples from QT-E such that the face space can be represented more accurately by our method; (2) bases classifiers learned from enlarged training data are more diverse for collaboration such that the overall discriminative power is much greater than a single classifier. Our method performs

better than conventional methods in dealing with the SSS problem thanking to both new sample generation and ensemble learning.

Comparison with existing ensemble methods

Ensemble learning based methods have been widely employed in solving the SSS problem. We compared our method with three kinds of representative ensemble methods: (1) global feature selection based on random subspace [WT06]; (2) patch based local feature extraction such as PCRC [ZZHS12] and a patch based method using LBP (Local Binary Pattern) [OPM$^+$11]; (3) global and local feature integration methods such as the 30 region method [Spr11] which defines 30 regions with large overlaps based on experimental experience and the multi-scale patch based method MPCRC [ZZHS12] which integrates the collaboration of multi-scale patches to construct the global feature representation. The performance of LBP methods are reported in [OPM$^+$11]. We implemented the 30 region method and used the supplied source code of PCRC, MPCRC in [ZZHS12]. This source code [ZZHS12] requires all subjects have the same number of training samples and test samples. We can only utilize it on ORL and Yale2 databases.

The results are presented in Tables 5.1–5.3. From these tables, we got the following four observations:

(1) **Ensemble can improve the performance:** The ensemble methods outperform conventional algorithms without ensemble on Table 5.2. This verifies the effectiveness of ensemble in alleviating the SSS problem. However, as shown on Table 5.1, higher performances are achieved by CCA and LPP families rather than existing ensemble algorithms. This means existing ensemble methods are not always effective in addressing the SSS problem.

(2) **Local based ensemble outperforms global based ensemble:** Compared to RS, the local patch based method PCRC performs better than all conventional methods without ensemble on Yale2 database (see Table 5.2) and obtains comparable results with S-LDA and S-LPP on ORL database (see Table 5.1) which performs best among all conventional methods without ensemble. This demonstrates the superiority of local based ensemble over global based ensemble in dealing with local variations. However, the patch based methods are very sensitive to the patch size. They have the performance degradation using inappropriate patch size. This has been illustrated on Table 5.3. Among the three versions of LBP, we can see that the one with patch size $7 \times 7w$ performs better than the others at $p = 2$. However, it is not so for the other two versions (LPP 5×5 and

LPP 7×7).

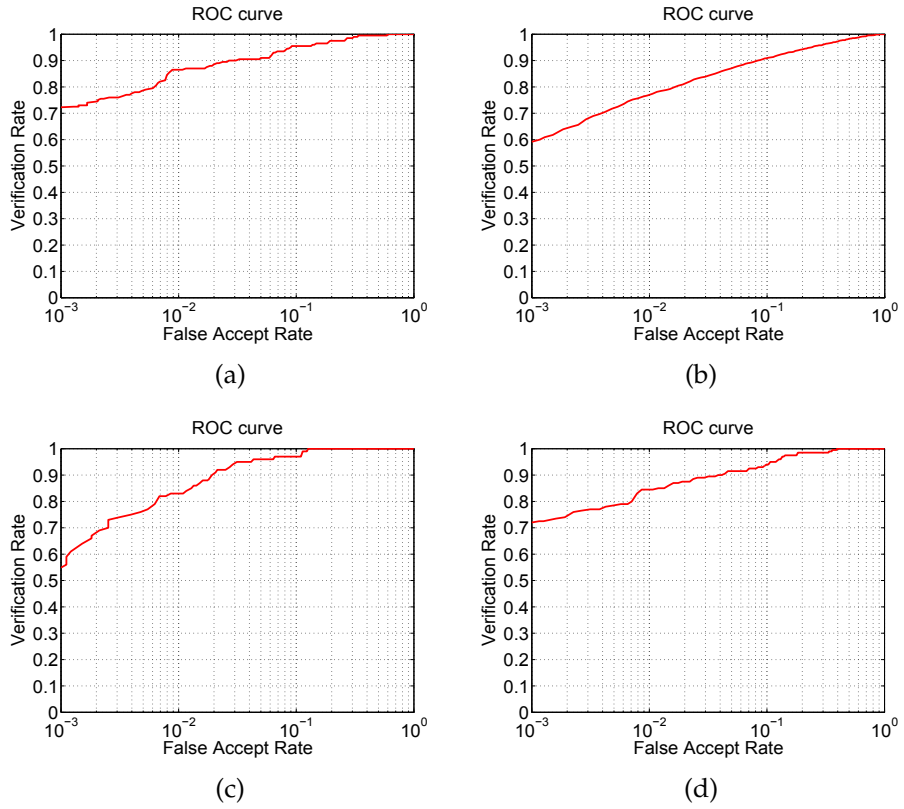


Figure 5.12: ROC curves of our method on four databases [ZLM14a]: (a) ORL($p = 2$), (b) Yale2 ($p = 5$), (c) AR (subset B), (d) FERET-1.

(3) **Global and local feature integration has better performance:** To overcome the disadvantages of global and local patch based methods separately, global and local feature integration methods were proposed. We observe that the 30 region method outperforms RS and PCRC on Table 5.2 and MPCRC performs best among all existing methods on Tables 5.1 and 5.2. This shows the effectiveness of integration of multi-scale local patches to generate the global feature representation. However, we can see that the 30 region method performs not well on the ORL and FERET databases. The main reason is that the 30 regions are designed after data registration based on experimental experience. For well-registered databases, e.g. Yale2, the 30 region method performs well. But for not well-registered databases, such as ORL and FERET, which contain variations of head poses, gender, ethnicity, and age, it performs not so well. Apart from this, we also find that MPCRC performs not well on ORL database which contains several variations (see Table 5.1). There are two main reasons: (a) MPCRC was

proposed on the small gallery data, while the test sample set may contain much more variations. Although the patch-based method can deal with some local deformations, it has limited ability to reconstruct the expected face space with large variations such as head poses etc.; (b) the multi-scale scheme just considers the complementary information at different scales of patches but not the geometric co-relationship between local patches. The literature finding in [LTW13] suggest that there is high overlapping between manifolds of local patches, which means the geometric information between local patches is also important.

(4) **Our method has better performance than existing ensemble methods:** Our method outperforms the existing ensemble methods in almost all experiments thanking to three benefits: (a) Our method expands the training sample set by generating new samples. Then, we can generate more diverse and accurate base classifiers to estimate the face space; (b) In contrast to existing methods, which divide face regions into separated patches, QT-E divides face region into blocks of variant sizes according to a tree-structure, which preserves the geometric correlationship between local blocks; (c) Other than existing methods, which involve all base classifiers for ensemble, our method develops a base classifier selection algorithm, which just selects appropriate base classifiers for ensemble. From Tables 5.1–5.3, we can see that Our-Sel. has better performance than Our-Org. for almost all splits of databases, which verifies the effectiveness of base classifier selection.

Comparison with State-of-the-arts on SSPP problem

We compared our method with 15 state-of-the-arts on SSPP problem, including PCA, (PC)2A [WZ02], E(PC)2A [CZZ04], 2DPCA [YZFY04], (2D)2PCA [DZZP10], SOM [TCZZ05], LPP [HYH$^+$05], SVD-LDA [ZCZ05], Block PCA [GA04], Block LDA [CLZ04], UP [DHG$^+$10], 30region [Spr11], MPCRC [ZZHS12], ADA [KSS$^+$13], and DMMA [LTW11]. Methods except the 30 region method [Spr11], MPCRC [ZZHS12], ADA [KSS$^+$13] were implemented by the authors of DMMA [LTW11]. ADA utilized two additionally databases to construct a generic set: XM2VTS and CAS-PEAL. Since XM2VTS is not a public database, we can not use it to do experiments. We just reported its performance on the FERET-2 database presented in [KSS$^+$13] for comparison. Table 5.4 shows the rank-one recognition accuracy of these methods on the AR and FERET-2 databases. We have the following five observations.

(1) **Block-wise supervised learning does not always perform better than**

Table 5.4: Evaluation on the AR and FERET-2 databases for SSPP face recognition

Method	AR							FERET	Year
	B	C	D	E	F	G	H		
PCA	97	87	60	77	76	67	38	84.0	1991
$(PC)^2A$	97	87	62	77	74	67	40	84.5	2002
$E(PC)^2A$	97	87	63	77	75	68	41	85.5	2004
2DPCA	97	87	60	76	76	67	37	84.5	2004
$(2D)^2PCA$	98	89	60	71	76	66	41	85.0	2005
SOM	98	88	64	73	77	70	42	91.0	2005
LPP	94	87	36	86	74	78	20	84.0	2005
SVD-LDA	73	75	29	75	56	58	19	85.5	2005
Block PCA	97	87	60	77	76	67	38	84.5	2004
Block LDA	85	79	29	73	59	59	18	86.5	2004
UP	98	88	59	77	74	66	41	90.0	2010
30region	91	94	37	91	66	81	22	86.0	2012
MPCRC	87	95	25	**96**	80	**88**	9	79.0	2012
ADA	N/A	N/A	N/A	N/A	N/A	N/A	N/A	92.6	2013
DMMA	**99**	93	**69**	88	**85**	79	45	93.0	2013
Our-Sel.	98	**96**	55	90	83	80	**48**	**94.5**	2014

block-wise unsupervised learning: We observe that Block-LDA does not perform better than block-PCA on the AR database. This is mainly because block-wise approaches assuming features are distributed in blocks uniformly. Block-LDA partitions face images into blocks of the same size and treats each block as an independent sample to estimate within-class scatter matrix. However, the literature finding [LTW13] suggests that there is high overlapping between manifolds of local blocks. Integrating separated blocks without considering any correlation between each other is not reliable to estimate the within-class matrix of LDA.

(2) **The key issue of virtual sample generation is to generate diverse samples:** SVD-LDA is a virtual sample generation method. However, it performs worst on the AR database. This is mainly because the virtually generated new samples are obtained just by discarding just some smaller singular values of the original image. The virtual samples are highly related to the original sample. Thus, using such samples, the within-class scatter matrix cannot be accurately estimated. From this finding we can draw that the key issue of applying virtual sample generation to take advantage of supervised learning is to generate new diverse samples, which are less correlated with the original samples.

(3) **Generic learning needs further investigation:** Compared with generic learning-based method ADA, we find both DMMA and QT-E achieve better performance than ADA. This further proves that generic learning needs to generate news samples other than grabbing samples from the generic set.

(4) **Global and local based ensemble performs not very well under SSPP:** MPCRC and 30 region method performs not stably under SSPP. MPCRC reduces to the original patch-based method PCRC under SSPP without any collaboration of multi-scale patches. As aforementioned, PCRC is very sensitive to the design of patch size, especially when database contains many local deformations. That is why it performs still well on subset E of AR database containing just neural frontal faces without any expressions but degrades severely to 79% on FERET-2 database which contains several local deformations e.g. facial expressions etc.. As mentioned before, the 30 region method degrades on not well-registered databases. Thus, it performs well on some subsets of AR database which contains only frontal images, but it is not true for FERET-2 database.

(5) **Our method shows promising results on SSPP:** Our method outperforms most of the comparison methods and obtains comparable performances with the recently developed DMMA algorithm on AR database and outperforms DMMA with an accuracy gain of 1.0 percent on the FERET-2 database. The reason why our algorithm is comparable to these state-of-the-arts thanking to the strategy of new sample generation which does expand the face space in the training stage. Benefit from introduction of random matrices, our new generated samples are quite diverse and different from the original training data compared with existing virtual sample generation methods. In addition, our method not only encodes discriminant features but also geometric information, which is also an important cue for recognition. Although our method has obtained promising results, we should acknowledge that DMMA outperforms ours on part of AR database. The main reason is that DMMA extracts features in a person-specific way rather than in a generic way. Through modeling the manifold surface for each subject, it specifies personal characteristics (e.g. age, hair style, etc.) during recognition, which contributes to the recognition. However, we argue that this subsject-specific strategy is challenged by the number of subjects involved in databases. The increase of the number of subjects makes the maximization of the manifold margins between different subjects more difficult. Then errors raises. That is why DMMA reduces its performance on the FERET-2 database which contains much more subjects than the AR database (about twice).

Evaluation on the optimal base classifier selection

For the better illustration of the effectiveness of base classifier selection, we use Kappa-error diagram [ZLC09] in Fig. 5.13, which visualizes pairwise diversity of

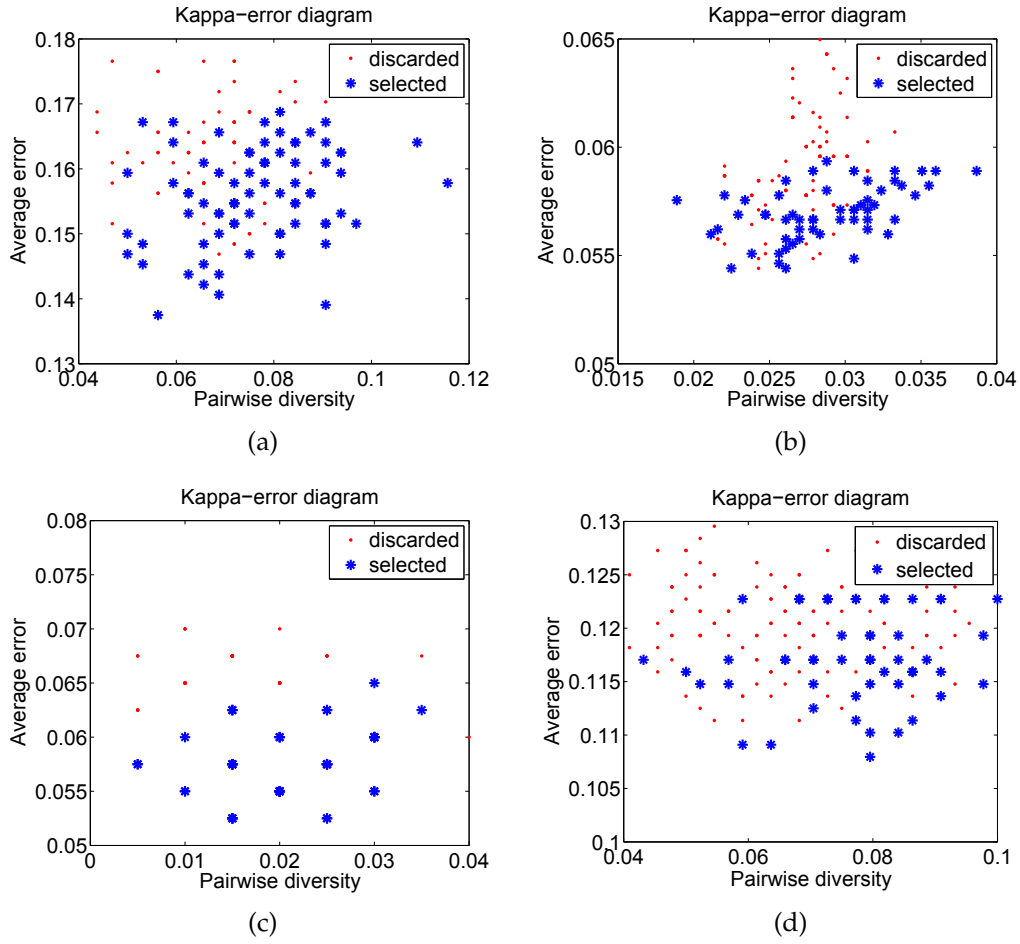


Figure 5.13: Kapapa-error diagrams on four databases (red points represent the discarded base classifiers while blue stars represent the selected base classifiers) [ZLM14a]: (a) ORL ($p = 2$), (b) Yale2 ($p = 5$), (c) AR (subset B), (d) FERET-1 ($p = 2$).

base classifiers against their mean error. In this figure, x-axis denotes the diversity div between each pair of base classifiers and y-axis represents the averaged individual error Err of pairwise classifiers. Desirable pairs of classifiers locate in the bottom right side, where div is high and Err is low. In Fig. 5.13, we use different colors and marks to differentiate base classifiers. Red points represent discarded base classifiers while blue stars represent selected base classifiers. We can see that both the number of base classifiers and the average errors are reduced obviously after selection. The elected base classifiers mainly locate at the right bottom side as we expected. This demonstrates the effectiveness of base classifier selection in alleviating the Diversity/Accuracy dilemma.

5.4.4 Comparison with Data-Adaptive Block based Algorithm

We compared the feature learning algorithm with large training data, which was developed in Chapter 3 for further discussion. Since the data-adaptive block based method utilize mutli-resolution Quad-Trees to generate multiple base classifiers, we consider this method is also a kind of ensemble method. The main difference between the data-adaptive block based method and the data-generation based method is that the former one utilize the original small data to generate base classifiers while the latter one generate base classifiers after data argumentation. From the experimental result in Tables 5.1–5.3, we observe that when the training data is very small for example (P=2,3 for ORL database, p=5 for Yale 2 database, and p=2 for FERET-1 database) the data-adaptive block based method has a very low-performance. The difference between the data-adaptive block based method and the data-generation based method is very significant. However, along with the increasing of the training data (e.g. p=5 for ORL database, p=10 for Yale 2 database, and p=3, 4 for FERET-1 database) the difference between these two method becomes nearer and nearer. Thus suggest that, the data-adaptive block based method can work well when the training data is large enough. It perform better than many other feature extraction method. However, it suffers from performance degradation when the training data is small. Although it can improves the performance of traditional PCA+LDA method, it performs worse than many other feature extraction methods, with or without ensemble.

An interesting observation is that although the difference between these two methods become small when the training data is large enough, the data-generation based method still outperforms the data-adaptive block based method. This is because although we introduce random matrices for data augmentation, the random matrices are not added to the face images directly. Instead, they are only added to template images to change the Quad-Tree partition. That means the introduction of random matrices does not introduce noises to the facial images, it only influence the image encoding patterns (Quad-Trees), according to which original facial images are re-organized to generate new samples.

From these observations, we can conclude that the data-adaptive block based method can improve the performance with large training data but still suffer from the SSS problem under small training data. In contrast, the data-generation based method can improve the performance of recognition algorithms no matter when the training data is large enough or not. It verifies the finding that the key point

of ensemble learning is to generate new samples again.

5.5 Summary

Many visual feature learning algorithms suffer from the small sample size (SSS) problem arising from the small number of training samples compared to the high dimensionality of the sample space. In this chapter, we develop a novel feature learning algorithm from extended training data encoded by Quad-Trees to solve the SSS problem. Our method overcomes the two serious problems of existing ensemble methods for SSS face recognition: (1) base classifiers are not diverse enough using small training data; (2) Diversity/Accuracy dilemma occurs during ensemble. We solve the first problem by generating more diverse base classifiers from the enlarged training data by generating new samples. And the second problem is settled by applying an optimal base classifier selection, which selects a subset of appropriate base classifiers for aggregation. Usage of this solution achieves the trade-off between the diversity and the accuracy of an ensemble. Extensive experimental results on four widely-used databases demonstrate that our method estimates a more accurate and robust ensemble for both SSS and SSPP face recognition.

Compared to the feature learning algorithm with large training data developed chapter 3, we have the following two observations: (1) when the training data is large, the data-adaptive based method outperforms the data generation based method since the template image developed in the data-adaptive block based method is more accurate in modeling the feature distribution while the template image in generic learning method may contain some noised due to the introduction of random matrices; (2)when the training data is small, the generic learning based methods performs better than data-adaptive block based method, since it is more effective in dealing with the SSS problem by taking advantages of ensemble learning and data augmentation.

Chapter 6

Feature Learning in Dynamic Environments using Helmholzt-Hodge Decompsoition and Quad-Tree

In the last three chapters (chapter 3–5), we have investigated feature learning on static images (image dataset). While feature learning in static images is relatively easier, it is much more challenging for dynamic environments (video sequences), especially when the videos are taken by moving cameras. In order to further evaluate the reliability of our method in dealing with more challenging cases, in this chapter, we investigates feature learning in moving camera videos. Video based feature learning is a hot research topic in the recent years thanks to the increasing development of mobile devices such as smart watch, smart phone, wearable devices (e.g. Google glasses, Baidu Eyes), and so on. While feature learning on static images is a relatively easy task, it is not so for video sequences, especially when the videos are taken by a moving camera. In the moving camera videos, camera motion and local object motions are mixed and dependent with other. Local object motions can offer motion features to many video based applications such as action recognition, tracking, content based surveillance, etc.. Since the camera motion and local object motions can influence with each other, if we use the original mixed motion field for these applications, object motion based applications will be very challenging and less accurate. Thus, we need to segment object motions from the scene and recover them to their true values.

Existing works on object motion segmentation model object motions directly.

However, object motions can be very complex and take a variety of forms, e.g. single or multiple, rigid or non-rigid, etc.. It is difficult to segment object motions directly. Our idea is to compensate camera motion first. After camera motion compensation, object motions can be segmented by subtracting camera motions from the original motion field. However, camera motion compensation also has some challenges. The first problem relates to camera motion interpretation. There are three kinds of camera motions (translation, radial motion, and rotation). For the translation, we can find an invariant parameter to interpret it. But it is not so for the others. The second problem relates to depth discontinuities in 3D scenes. Depth discontinuities lead to motion discontinuities, which make camera motion compensation difficult. Under such cases, just Quad-Tree is not enough. This dissertation introduces a tailored Helmholtz-Hodge decomposition (HHD) for assistance. HHD can interpret three kinds of camera motions well. After camera motion compensation, a data-drive Quad-Tree partition is performed for object motion segmentation. Quad-Tree can also detect depth discontinuities from the scene, making our method be available for 3D scenes. That is, the features detected by our Quad-Tree include both object motions and depth discontinuities. We evaluated the developed method on mixed motion segmentation using several benchmark video sequences.

6.1 Problem Review and Related Works

Motion segmentation is relatively "easy" when the background is stationary and videos are captured by a fixed camera. Well-known methods such as background subtraction and frame differencing can be employed in dealing with this problem successfully. However, when it comes to dynamic scenes with an unconstrained and a prior unknown camera motion, it becomes more challenging. The moving camera causes all pixels in the image moving. The 2D image motion observed is caused by both the 3D motion of moving camera, by the changes of internal camera parameters (e.g., camera zoom), and by 3D motions of moving objects. These effects are mixed and interdependent with each other making the motion field rather obscure.

As local object motions can be very complicated, it is difficult to model and segment them directly. An effective approach is to compensate the camera-induce image motion first. And the residue motions must belong to moving objects. To compensate camera-induced image motion, we first need to find an appropriate

way to interpret it. A common theme in representing motions is either by trajectories of key features or by dense motion field (optical flow). Accordingly, literature methods can be broadly divided into: (1) feature based approaches and (2) dense based approaches.

Feature based methods focus on the classification of trajectories of selected features into different groups (subspaces) [ZN11, Ald13]. Representative methods include: factorization-based, algebraic, and statistical based methods. Factorization-based methods [Ich99] attempt to directly factor the trajectory matrix of multiple motions into sub-matrices of different motions. These methods require motions to be independent of one another, and thus cannot deal with dependent motions. Algebraic methods, e.g. Generalized Principal Component Analysis (GPCA) [VMS06] can partially deal with dependent motions. A representative statistical method Multi-Stage Learning (MSL) [SK04] was proposed, which is based on Costeira and Kanade's factorization method (CK) [CK98] and Kanatani's subspace separation method (SS) [KM02]. Feature based methods can handle camera motions in several cases. However, they often assume an orthographic camera model, which is a simplified model of the real perspective camera. It requires the focal length of a camera to be long enough to avoid any perspective distortions on the depth of 3D points, which cannot be met in many applications. Moreover, these methods rely much on the presence of strong features and a robust feature extraction, classification and tracking algorithm to obtain feature trajectories as motion cues [SaXW13, OMB14]. In addition, as they only output a segmentation of sparse key features, postprocessing is required to obtain a dense segmentation.

Dense based methods perform pixel-level segmentation on image plane motion (optical flow). They usually assume the camera-induced image motion by a parametric transformation ranging from translation to perspective transformation using different parameters [SSH05]. Pixels that are consistent with the estimated model are supposed to be inlier, while others are supposed to outliers. Since the inlier estimation is often affected by outliers, several outlier removal methods were proposed. For example, a regression scheme, using gradient descent (GD) [SSH05] or least squares (LS) [SHO00], was applied to refine the inlier model by iteratively excluding outliers. Outlier rejection filter [CB10] explicitly filtered motion vectors by checking their similarity in a predefined window. RANSAC [FB81] is a statistical method, which estimates the inlier on the data containing outliers by iteratively updating the probability of inlier. Recently, a joint

inlier estimation and motion segmentation method [CB11, QB13] was proposed, which performs inlier estimation and outlier rejection simultaneously. In contrast to feature based methods, dense based methods do not rely on the strong key features tracked throughout the scene. However, they suffer from two big problems: (1) the parametric models used for inlier estimation are just approximations of camera motions which only hold for the restricted cases of camera motion. Moreover, they rely on prior-knowledge for model selection. (2) The image plane motion depends on the distance of 3D points from the camera. Objects at different depths may have different optical flow even if they share the same real-world motion, which leads to depth-dependent segmentation. To overcome this problem, a state-of-art algorithm [NHLM13] was proposed, which utilized optical flow direction other than magnitude for coherent motion segmentation. However, it only works for camera translation, while camera zoom and rotation are still challenges.

Drawbacks of existing methods make it clear that existing motion segmentation methods suffer from two severe problems: (1) The first problem is on the camera motion interpretation. The intuitive interpretation of the 2D motion field is generally based on Cartesian coordinate system with two bases x, y. A motion vector can be projected into x and y components, denoted by u and v, respectively. For camera translation, we can use an invariant parameter $\frac{u}{v}$ to interpret it. But for the rotation and radial motion, $\frac{u}{v}$ changes with x, y changing on motion field. They cannot be represented by an invariant parameter using existing methods. (2) The second problem is that depth discontinuities in 3D scenes lead to motion discontinuities, making the segmentation of background motion (inlier) to be incoherent. Thus, we need to develop new algorithms, which can interpret the three kinds of camera motions (translation, rotation, and radial motion) be in an invariant way and be of high ability in dealing with depth discontinuities meanwhile.

6.2 Algorithm Architecture

To solve the two problems mentioned above, in this chapter, we develop a dependent motion segmentation algorithm by introducing an amended Helmholtz-Hodge decomposition (HHD) and a data-driven Quad-Tree partition. HHD is used for camera motion compensation while Quad-Tree is for object motion segmentation. HHD can interpret the three kinds of camera-induced image motions

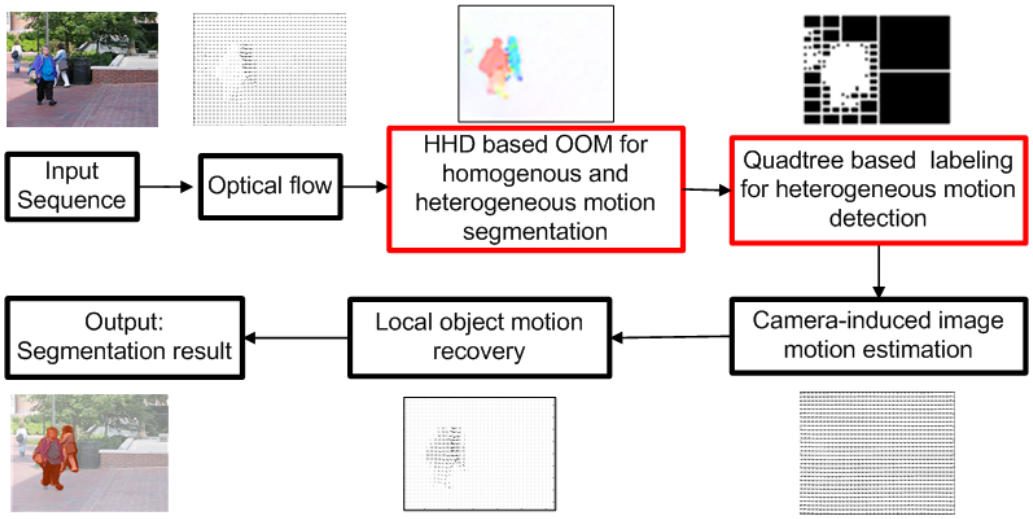


Figure 6.1: Dependent motion segmentation using Helmholtz-Hodge decomposition (HHD) and Quad-Tree.

by its two components: curl-free component and divergence-free component. And the Quad-Tree based object motion segmentation can deal with depth discontinuities in 3D scenes. The outline of our algorithm is illustrated in Fig. 6.1.

In the following, we first describe the motion flow modeling in 2D and 3D scenes and discuss the inlier/outlier representation based on optical flow modeling. Then we introduce the amended HHD based camera motion compensation. Next is about object motion modeling using Quad-Tree labeling. Finally, we explain the inlier estimation and outlier recovery. The segmentation result is obtained accordingly.

6.2.1 Optical Flow Modeling

Camera undergoes two kinds of 3D motions: translation $T = (T_X, T_Y, T_Z)$ and rotation $R = (R_X, R_Y, R_Z)$. The instantaneous image motion of a general 3D scene can be formulated as [IA98]:

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} -f_c(\frac{T_X}{Z} + R_Y) + x\frac{T_Z}{Z} + yR_Z - x^2\frac{R_Y}{f_c} + xy\frac{R_X}{f_c} \\ -f_c(\frac{T_Y}{Z} + R_X) - xR_Z + y\frac{T_Z}{Z} - xy\frac{R_Y}{f_c} + y^2\frac{R_X}{f_c} \end{bmatrix}, \quad (6.1)$$

where f_c is the focal length, Z is the depth of the 3D point, and $(u(x,y), v(x,y))$ denotes the image plane motion at coordinate (x,y). Image motion has more specific representations in different scenes (2D or 3D).

Motion in 2D Scenes

When the scene is parallel to the image plane, all the scene objects are at the same distance from the camera. No depth variations present in the scene ($\Delta Z = 0$). The camera-induced image motion can be modeled by a single parametric transformation, which is of low-order polynomial function:

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy \\ a_6 + a_7 x + a_8 y + a_4 xy + a_5 y^2 \end{bmatrix}, \tag{6.2}$$

where the parameters $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8)$ are functions of camera motion (R, T), f_c, and Z as depicted in Eq.(6.1). Here, Z is constant.

We define the scenes satisfying the above-mentioned condition as 2D scenes. In practice, the scene may contains small depth variations. However, when the depth variation within the scene is much smaller than the overall 3D range of the scene from the camera, these scenes can also be considered as 2D scenes [IA98].

Motion in 3D Scenes

When the scene contains large depth variations, it cannot be approximated by a flat/planar surface which is parallel to the image motion. In this case, we need to use a set of planar surfaces instead of a single one to represent it [IA98]. We define such scenes as 3D scenes. Camera motions under 3D scenes can be conducted in the following cases:

- Camera translation: when the camera is translating, the image motion in Eq.(6.1) becomes:

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} -\frac{T_X}{Z} + x\frac{T_Z}{Z} \\ -\frac{T_Y}{Z} + y\frac{T_Z}{Z} \end{bmatrix}, \tag{6.3}$$

  Camera translation consists of two cases: (1) translation, which is parallel to the image plane, where $T_Z = 0$; and (2) perspective motion, where $T_X = 0, T_Y = 0$. Perspective motion contributes to a radial image motion (expansion or contraction). For both cases, the image motion observed in

the scene depends on the depth Z of scene points. Different planar surfaces have substantially different induced image motions depending on their depths. A single parametric transformation becomes insufficient for modeling the image motion. We need to use a set of parametric transformations $\{P_1, P_2, ..., P_N\}$ to represent it [IA98], where N denotes the number of planar surfaces. P_i is defined by a polynomial function with different parameters ranging from translation (2 parameters) to perspective transformation (8 parameters). The specific definition of $P_i$is usually depending on the prior knowledge of what kind of camera motion is involved in the scene [SSH05].

- Camera Rotation: When the camera undergoes a pure rotation, Eq. (6.1) becomes:

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} -R_Y + yR_Z - x^2R_Y + xyR_X \\ R_X - xR_Z - xy2R_Y + y^2R_X \end{bmatrix}, \tag{6.4}$$

  The contribution of camera rotation to the image plane motion is independent of depth Z of scene points.

Camera-induced image motion C_{IM} can be regarded as the combination of these three basic motions: translation T_{IM}, radial motion (or named as perspective motion) P_{IM}, and rotation R_{IM} by:

$$C_{IM} = \alpha T_{IM} + \beta P_{IM} + \gamma R_{IM}. \tag{6.5}$$

where α, β, γ are three regularization parameters.

Among these three basic motions, translation and radial motion are influenced by depth variations in 3D scenes. Although rotation is independent of depth validations, there is another problem we need to consider. The intuitive interpretation of the 2D motion field is generally based on Cartesian coordinate system with two bases x, y. A motion vector can be projected into x and y components, denoted by u and v, respectively. For camera translation, we can use an invariant parameter $\frac{u}{v}$ to interpret it. But for the rotation and radial motion, $\frac{u}{v}$ changes with x, y changing on motion field. They cannot be represented by an invariant parameter using existing methods.

Inlier/Outlier Representation

In this paper, inlier is the abbreviation of inlier optical flow, which refers to the camera-induced image motion. Outlier is the abbreviation of outlier optical flow, which refers to motion discontinuities due to the relative motion between moving objects and camera, depth discontinuities, and noises. Our method works for dense motion field, depicted by optical flow. Existing optical flow estimators often add some constraints to make the estimation robust and accurate. Commonly used constraints include the intensity constancy assumption, gradient constancy assumption, and local smoothness assumption, which are interpreted as follows:

- **Intensity Constancy Assumption**: which assumes the intensity value of pixels between two adjacent frames $I(x, y, t)$ and $I(x, y, t + 1)$ does not change. It can be interpreted by an energy function as:

$$E_{Intensity}(u, v) = \int_{\Omega} (|I(\mathbf{x} + w) - I(\mathbf{x})|)^2 d\mathbf{x}, \tag{6.6}$$

  where Ω denotes a local search window sliding across the image region, $w := (u, v, 1)$ denotes the searched displacement vector.

- **Gradient Constancy Assumption**: which assumes the gradient of intensity value of pixels between these two frames does not change:

$$E_{Gradient}(u, v) = \int_{\Omega} (|\nabla I(\mathbf{x} + w) - \nabla I(\mathbf{x})|)^2 d\mathbf{x}. \tag{6.7}$$

- **Local Smoothness Assumption**: We may expect some outliers during estimation. To make the estimated motion field smooth, local smoothness assumption was considered:

$$E_{Smoothness}(u, v) = \int_{\Omega} (|\nabla u|^2 + |\nabla v|^2) d\mathbf{x}. \tag{6.8}$$

Optical flow is calculated by minimizing the total energy function: $E = E_{Intensity} + \alpha E_{Gradient} + \beta E_{Smoothness}$, where α, β are two regularization parameters.

Benefit from local smoothness constraint, for 2D scenes with small depth variations, the optical flow field is piece-wise smooth and can be represented by a single parametric transformation of low-order polynomial function (P^L). But for 3D scenes with large depth variations, the local smoothness constraint fails and the estimation error occurs. The optical flow field becomes inaccurate and we

have to use a high-order polynomial function (P^H) to represent it. That is:

$$\begin{aligned} OF_{2D} &= P_1 = P^L \\ OF_{3D} &= \{P_1, P_2, ..., P_N\} \approx P^H. \end{aligned} \tag{6.9}$$

where OF_{2D} means optical flow filed in 2D scenes and OF_{3D} denotes the optical flow field in 3D scenes.

For the inlier, most part of it is caused by camera motion, which is smooth and continuous. Here we use $\vec{V}_{inlier}^{homogeneous}$ to represent this part, where the "homogeneous" refers to the continuity of the motion field. However, motion discontinuities occur at the boundaries of scene objects due to depth discontinuities. Their motion field is heterogeneous. Thus, we use $\vec{V}_{inlier}^{heterogeneous}$ to represent this part, where the "homogeneous" means the motion field is unsmooth and non-continuous. In contrast to inlier, outliers belonging to the motion field of moving objects. It suffers from motion discontinuities because of the relative motion between camera and moving objects. Most part of outliers are heterogeneous and just very small part of them are homogeneous. Thus, we use $\vec{V}_{outlier}^{heterogeneous}$ and $\vec{V}_{ioutlier}^{heterogeneous}$ to represent the heterogeneous and homogeneous part, respectively. Both the inlier and outlier can be represented by the homogeneous part and heterogeneous part as below:

$$\begin{aligned} \vec{V}_{inlier} &= a_1 \vec{V}_{inlier}^{homogeneous} + b_1 \vec{V}_{inlier}^{heterogeneous}, a_1 >> b_1, \\ \vec{V}_{outlier} &= a_2 \vec{V}_{outlier}^{homogeneous} + b_2 \vec{V}_{outlier}^{heterogeneous}, a_2 << b_2. \end{aligned} \tag{6.10}$$

where a_1, b_1, a_2, b_2 are quantity coefficients. For the inlier, most part of it is homogeneous, thus $a_1 >> b_1$. It is opposite for the outliers, thus $a_2 << b_2$. The homogeneous part should be represented by a low-order polynomial function while the heterogeneous part should be represented by a high-order polynomial function.

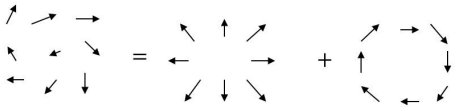
6.2.2 HHD based Camera Motion Modeling in 3D Scenes

Helmholtz-Hodge Decomposition

Helmholtz-Hodge decomposition is one of the fundamental theorems in fluid dynamics. The principle of HHD [BNPB13] is explained as follows: for an arbitrary image motion field $\vec{\xi}$, it decomposes it into two components: a curl-free compo-

nent ∇E and a divergence-free component $\nabla \times \vec{W}$:

$$\vec{\xi} = \nabla E + \nabla \times \vec{W}. \tag{6.11}$$



Here, E and W are 3D potential surfaces defined as: 1. *Scalar potential surface*, whose gradient is the curl-free component ∇E; 2. *Vector potential surface*, whose curl operation denotes the divergence-free component $\nabla \times \vec{W}$ [BNPB13]. The curl operator and the gradient operator have the following relationship:

$$\nabla \times \vec{W} = (\nabla W)^{\perp}. \tag{6.12}$$

The three kinds of camera induced image motions can be interpreted by the two components of HHD in this way:

- Radial motion is irrational, and can be represented by the curl-free component;
- Rotation is incompressible, and can be represented by the divergence-free component.
- Translation is irrational and incompressible, and can be represented by both curl-free and divergence-free components without any change.

In the literature, many algorithms have been proposed for the implementation of HHD. For example, Polthier et al. [PP03] derived a technique for 2D discrete vector fields. Tong et al. [TLHD03] extended it to discrete vector fields on 3D meshes. To ensure HHD be able to identify the inlier as one segment irrespective of depth variations, we add two assumptions to the previously defined HHD and introduce an amended HHD in this chapter. The two assumptions are: (1) the original motion field should be piece-wise smooth; (2) HHD is implemented based on global minimization. The implementation is as follows: since ∇E and $\nabla \times \vec{W}$ are the projection of the original motion field $\vec{\xi}$ to the space of the curl-free field and the divergence-free field, respectively, the distance between $\vec{\xi}$ and two projected components should be minimal. Therefore, we apply energy minimiza-

tion to calculate the two components:

$$\begin{aligned} min(D(E)) &= min(\int_\Omega \parallel \nabla E - \vec{\xi} \parallel^2 d\Omega), \\ min(G(\vec{W})) &= min(\int_\Omega \parallel \nabla \times \vec{W} - \vec{\xi} \parallel^2 d\Omega). \end{aligned} \tag{6.13}$$

where Ω denotes the image domain. According to the definition of HHD, the divergence-free component ($\nabla \times \vec{W}$) does not exist in the curl-free component (∇E). And vice versa. We can derive the following criteria:

$$\begin{aligned} \int_\Omega \nabla \times (\nabla E) d\Omega = \int_\Omega \nabla \times (\vec{\xi} - \nabla \times \vec{W}) d\Omega = 0, \\ \int_\Omega \nabla \cdot (\nabla \times \vec{W}) d\Omega = \int_\Omega \nabla \cdot (\vec{\xi} - \nabla E) d\Omega = 0. \end{aligned} \tag{6.14}$$

In the discrete domain, Eq.(6.14) can be rewritten as:

$$\begin{aligned} \sum_{i \in \Omega} \nabla \cdot (\nabla E_i) = \sum_{i \in \Omega} \nabla \vec{\xi}_i. \\ \sum_{i \in \Omega} \nabla \times (\nabla \times \vec{W}_i) = \sum_{i \in \Omega} \nabla \times \vec{\xi}_i, \end{aligned} \tag{6.15}$$

Since they are linear functions, we can abbreviate them as:

$$S_1 E = B, \qquad S_2 W = C. \tag{6.16}$$

where S_1 and S_2 are $N \times N$ sparse element matrices (N is the total number of the nodes on the image domain Ω), E and W are $N \times 1$ vectors to be calculated, B and C represent the right hand side of Eq.(6.15).

We calculate the potential surfaces E and W by solving Eq.(6.16). Then $\nabla \times \vec{W}$ is subsequently obtained by Eq.(6.12).

Camera Motion Representation by HHD

Based on the preceding discussion (see 'Inlier/Outlier Representation'), the optical flow field can be represented by a low-order polynomial function for $\vec{V}_{inlier}^{homogeneous}$, $\vec{V}_{outlier}^{homogeneous}$, and by a high-order polynomial function for $\vec{V}_{inlier}^{heterogeneous}$, $\vec{V}_{outlier}^{heterogeneous}$. In our algorithm, the polynomial function actually corresponds to the potential surface of HHD. As we added two assumptions to the implementation of HHD, the potential surfaces of the amended HHD are piecewise smooth. They approximate the basic shape of motion field and thus corresponds to a low-order polynomial function. Thus, the homogeneous motion field, which should be represented by a low-order polynomial function can be

interpreted by the two components of HHD as follows:

$$\{\vec{V}_{inlier}^{Homogeneous}, \vec{V}_{outlier}^{Homogeneous}\} = k_1(\nabla E) + k_2(\nabla \times \vec{W}). \tag{6.17}$$

where k_1 and k_2 are two regularization parameters. They will be determined in the next section.

The heterogeneous motion, which corresponds to a high-order polynomial function, cannot be represented by HHD and will be reserved in a reminder. In the next section, we will construct an object-motion oriented map (OOM) from this remainder to detect heterogeneous motions.

6.2.3 Object Motion Modeling in 3D Scenes

To construct the object motion oriented map, we fist need to represent the homogeneous motion according to Eq. (6.17). Then OOM can be calculated by subtracting the homogeneous motion from the original motion field $\vec{\xi}$ as follows:

$$OOM = \vec{\xi} - k_1(\nabla E) - k_2(\nabla \times \vec{W}). \tag{6.18}$$

The key point is the determination of the two parameters k_1 and k_2, which should be discussed according to the type of the camera motion involved in the scene. In this paper, we define two distance functions in Eq.(6.19) for this purpose.

$$\begin{aligned} d_1 &= \frac{\|\xi - \nabla E\|}{\|\xi\|}, \\ d_2 &= \frac{\|\xi - \nabla \times \vec{W}\|}{\|\xi\|}, \end{aligned} \tag{6.19}$$

where d_1 denotes the distance between $\vec{\xi}$ and the curl-free component, and d_2 denotes the distance between $\vec{\xi}$ and the divergence-free component. As aforementioned, radial motion only exists in the curl-free component; rotation only presents in the divergence-free component; and translation can present in both components. We have the following three observations:

- If $d_1 < 0.5$, $d_2 > 0.5$, it means the curl-free component is much similar to the original optical flow field while the divergence-free component differs much. This implies the camera-induced image motion belongs to a radial motion, and exists only in the curl-free component. Thus, $k_1 = 1$, and $k_2 = 0$;

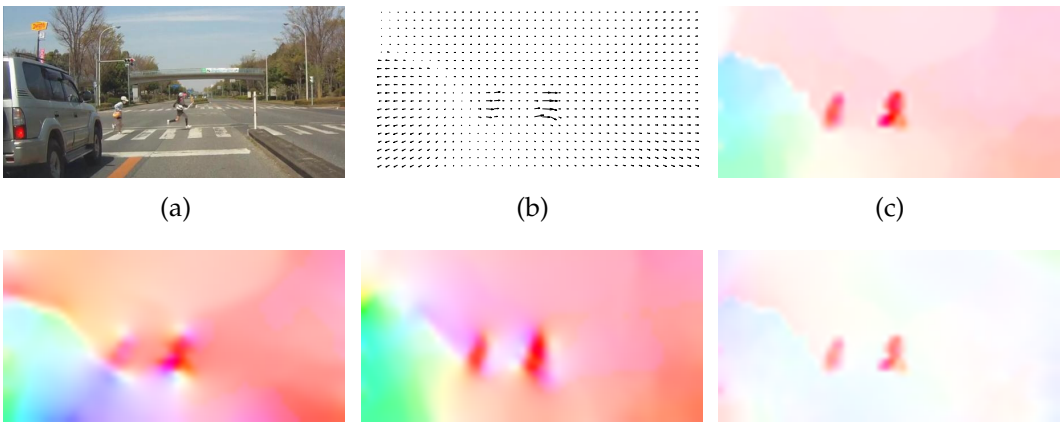


Figure 6.2: An example 3D scene: (a) one frame; (b) the input optical flow; (c) visualization of the input optical flow by color coding [BSL+11]; (d) curl-free component; (e) divergence-free component; (f) OOM. (Copyrighted by Springer LNCS [LZM14])

- If $d_1 > 0.5$, $d_2 < 0.5$, it means the divergence-free component is much similar to the original optical flow field while the curl-free component differs much. This implies the camera-induced image motion belongs to rotation and exists only in the divergence-free component. Thus, $k_1 = 0$, and $k_2 = 1$;

- If $d_1 < 0.5$, $d_2 < 0.5$, it means both components are very similar to the original motion field. This implies the camera-induced image motion is translation and exists in both components. Thus, $k_1 = k_2 = 1/2$.

As the camera motion is a generic motion, all the scene points share the same kind of camera motion. Thus, k_1 and k_2 are invariant parameters across the whole scene. We can use a unique k_1 and k_2 for all the places of the scene. After determining k_1, k_2, we can calculate OOM by Eq. (6.18).

We use an example 3D scene in Fig. 6.2 to illustrate the procedure of OOM construction. In the image sequence (a), two people are walking from left to right, a car is stopping at the left side. A camera is moving behind the car. From the original optical flow field in (b) and its color image in (c), we can see that the car shows larger motion than other pure flat scene area in distance. After HHD decomposition, the OOM in (f) demonstrates that most part of homogeneous motion field of static scene objects has been removed from OOM. It only contains heterogeneous motion caused by depth discontinuities and moving objects.

6.2.4 Quad-Tree based Motion Segmentation

To detect the heterogeneous motion, we need to label them from OOM. As depth discontinuities and moving objects vary at different locations, it is rather difficult to label them based on global thresholding. To this end, we introduce a data-driven Quad-Tree scheme, which casts the heterogeneous motion labeling to local subregions. If a region is determined as heterogeneous according to a criterion function in Eq.(6.20), it will be divided into four subregions. The criterion function considers the following two conditions:

- The variance of pixel values in a region R is higher than or equal to a threshold variance T_{var};
- The mean pixel value in R is higher than or equal to a threshold mean value T_{mean};

Mathematically it can be formulated as:

$$doSplit(R) = true, while \begin{cases} var(R) \geq T_{var} \quad or, \\ mean(R) \geq T_{mean}. \end{cases} \tag{6.20}$$

We apply the first condition to detect the heterogeneous motion caused by depth discontinuities ($\vec{V}_{inlier}^{heterogeneous}$), where the values changes violently in local regions and second condition to detect the heterogeneous motion caused by moving objects ($\vec{V}_{outlier}^{heterogeneous}$), where higher than the threshold implies the local object motions occupy larger part of the region. Partition performs till no more regions can be split. Regions of smallest size $smallR$ are labeled as foreground regions containing heterogeneous motions and will be excluded from the inlier estimation in the next procedure. The rest larger regions $largeR = wholeR - smallR$ represent the homogeneous region and will be evolved in inlier estimation, where $wholeR$ represents the whole region of OOM.

Usually, OOM is split to less and larger blocks while the thresholds T_{var}, T_{mean} are large, but more and smaller blocks while they are small. Too large thresholds imply not all heterogeneous motion can be detected, and too small ones mean heterogeneous motion is over-segmented. By defining a universal threshold, we may suffer from such under-segmentation or over-segmentation problems. To avoid this, we propose a data-adaptive threshold definition algorithm based on the Quad-Tree structure. The entire procedure is as follows:

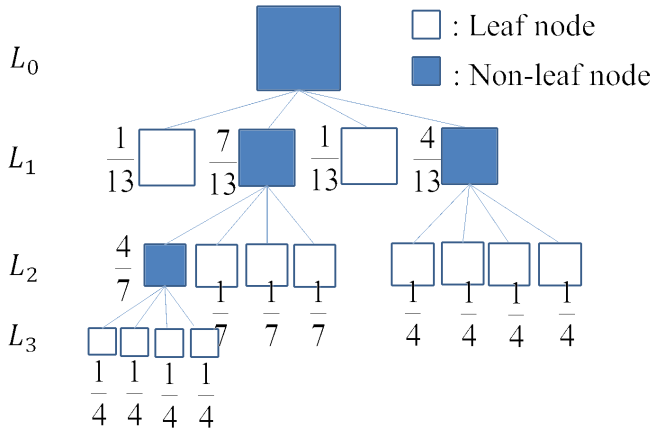


Figure 6.3: An example Quad-Tree partition.

1. Assign an initial estimate for the thresholds: $T_{var} = var(wholeR)$, $T_{mean} = mean(wholeR)$;

2. Partition OOM using T_{var} and T_{mean} based on Eq.(6.20). This will produce a hierarchical Quad-Tree structure, which contains leaf and non-leaf nodes. Please see Fig. 6.3 for example. For each leaf node $subR_i$, we calculate its variance and mean value: $var(subR_i)$, and $mean(subR_i)$;

3. From bottom to upper side, we calculate the variance and mean value of each non-leaf node $subR_j$ by averaging its four children based on a weighted sum rule. The weight of each child node $subR_k$ is defined by the number of leaf nodes in the subtree rooted at $subR_k$ against that number rooted by its father node $subR_j$. A higher weight implies the higher probability a sub-region is foreground. An example of the quad-tree structure with weight assignment is shown in Fig. 6.3. When we reach the first level, we can calculate the variance and mean value of the root node $var(root)$ and $mean(root)$ and assign them to the new thresholds T'_{var} and T'_{mean}, respectively;

4. Repeat Steps 2 through 3 until the difference between two quad-tree partitions is smaller than a threshold distance. Specifically, we encode each quad-tree by a 0–1 sequence according to a full quad-tree partition, where leaf nodes are assigned one and non-leaf nodes are assigned zero. The distance between two quad-trees is defined by the Hamming distance of two 0–1 sequences. If this distance is smaller than a threshold distance denoted by $\varepsilon * M$, where ε is the tolerance error and M is the length of the 0–1 sequence, then the iterative process stops. Here, we define $\varepsilon = 5\%$.

(a) (b)

Figure 6.4: Quad-Tree partition on the example 3D scene: (a) OOM; (b) Quad-Tree partition. (Copyrighted by Springer LNCS [LZM14])

Our algorithm updates the thresholds according to the last Quad-Tree structure in each repetition. The new thresholds are closer to the value of foreground regions containing heterogeneous motions. Finally, our algorithm can find the most appropriate thresholds for each OOM automatically. Figure 6.4 shows the Quad-Tree partition on the example 3D scene. We can see that Quad-Tree is effective in labeling both moving objects and depth discontinuities from OOM. Here, the features learned by Quad-Tree includes both local object motions and depth discontinuities. Next, we will use surface fitting based camera motion estimation to compensate depth discontinuities and rest object motions only.

6.2.5 Camera Motion Estimation using Surface Fitting

After Quad-Tree based heterogeneous motion labeling on OOM, we will use the rested regions, which correspond to homogeneous motion field for inlier estimation. We first illustrate the procedure on scalar potential surface E. The procedure on $\vec{W}$ is analogous.

As aforementioned, the potential surface of HHD should be smooth and represented by a low-order polynomial function. Thus, we formulate the problem of inlier estimation from E as construction of a new smooth surface E', which approximates the smooth basic shape of E. In [SSH05], several parametric models were conducted for inlier estimation. These models are designed for camera motions ranging from simple translation to complex perspective transformations. But, the limitation is that a prior-knowledge of motion structure is required to select an appropriate model. We employ a surface fitting solution using a low-order

polynomial function as follows:

$$z = a_{d0}x^d + a_{0d}y^d + \cdots + a_{ij}x^i y^j + \cdots + a_{10}x + a_{01}y + a_{00}, \tag{6.21}$$

It has been known that the higher the degree of d, the more details the surface approximates, but it can potentially evolves some local deformations. On the contrary, lower degree polynomial yields a smoother and simpler surface which only approximates the inlier positions but maybe not accurate enough. In our method, a polynomial of $d = 5$ is employed so as to produce a smooth and accurate surface E'. Finally, we will have a smooth surface E' which can fit fits the base of E. Similarly, we get a new smooth potential surface $\vec{W}'$ which approximate the base of $\vec{W}'$. The inlier of curl-free component is calculated by $G_1 = \nabla E'$. The inlier of divergence-free component is computed by $G_2 = \nabla \times \vec{W}'$. The final inlier optical flow is estimated by linear combination of G_1 and G_2 using Eg.(6.22), where k_1, k_2 have been determined in Sect.4.1.

$$G = k_1 G_1 + k_2 G_2, \tag{6.22}$$

Although the heterogeneous part of inlier (motion discontinuities caused by depth discontinuities) was not involved in estimation, since most part of inlier has been involved, its heterogeneous part can be approximated by its homogeneous part. In this way, the surface fitting separates the depth discontinuities from the true moving objects. Figure 6.5 (a), (b) shows the estimated inlier of the example 3D scene. We can see that our method estimates the inlier accurately.

6.2.6 Object Motion Recovery

After inlier estimation, outliers can be recovered by subtracting the inlier from the original motion field subsequently. The segmentation is obtained by assigning binary labels on the pure outlier motion field. Since the surface fitting used in inlier estimation has a defect that it better fits the data in the middle but goes wild at the edge of the $x - y$ domain of the original data. To refine the segmentation map, the raw result is filtered by the mean-curvature of the original potential surfaces. Figure 6.5 shows the recovered outlier motion field in (c) and the final segmentation map in (d).

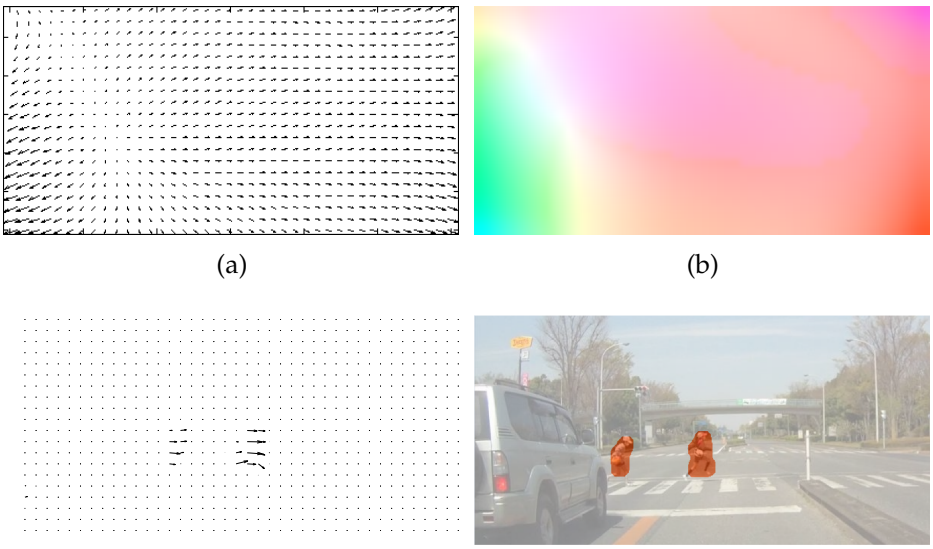


Figure 6.5: Inlier estimation and outlier recovery on the example 3D scene: (a) the estimated inlier optical flow; (b) color visualization of (a); (c) the recovered outlier optical flow; (d) segmentation result. (Copyrighted by Springer LNCS [LZM14])

6.3 Experimental Evaluation

6.3.1 Datasets and Experiments Design

The system's performance is evaluated on four benchmark datasets: Hopkins [TV07], Berkeley Motion Segmentation [BM10], Complex Background [NHLM13], and SegTrack [TFM10]. The Hopkins dataset contains three category video sequences: checkerboard, car, and people. It provides ground truth segmentation on selected features tracked throughout the sequence. Since checkerboard sequences do not correspond to natural scenes. We just use one sequence (1R2TCR) to show the effectiveness of our method in dealing with cameras rotation. The Berkeley dataset is derived from the Hopkins dataset, which consists of 26 moving camera videos: car, people, and Marple sequences. This dataset has full pixel-level annotations on multiple objects for a few frames sampled throughout the video. Since Marple sequences mainly contain static scenes or the objects are static, it is little challenging for our method. We did not use them in the experiments. In this paper, we choose the car and people sequences, which are contained in both Hopkins and Berkeley datasets to compare our method with dense based and feature based methods, respectively. We use the other two datasets: Complex Background and SegTrack, which contain extremely challenging scenes

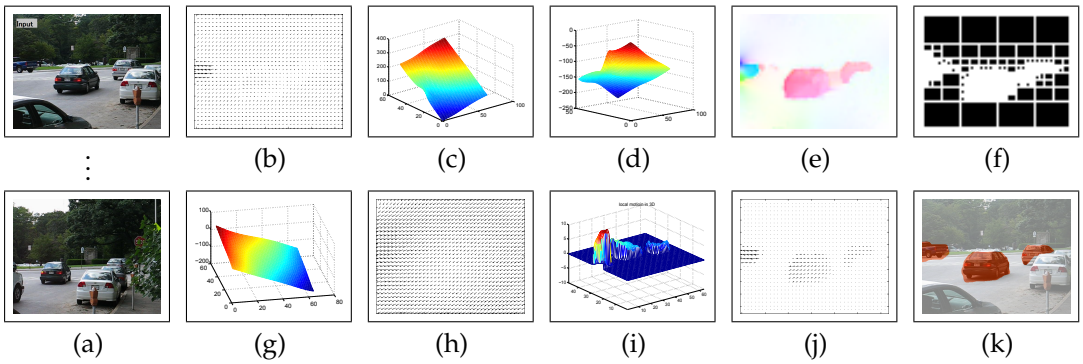


Figure 6.6: Scenario 1: cars2 sequence: (a) image sequence from frame 1 to frame 20; (b) optical flow of one frame; (c) potential surface E; (d) potential surface $\vec{W}$; (e) the object-motion oriented map (OOM); (f) Quad-Tree partition on OOM; (g) the estimated inlier potential surface; (h) inlier optical flow field; (i) the recovered local motion shown in 3D; (j) outlier optical flow field; (k) segmentation result.

to highlight the strength of our method. They provide full pixel-level annotations on multiple objects at each frame within each video.

In this section, we first use four sequences to illustrate the performance of our method in dealing with different camera motions. Then we compare our method with several existing methods. Optical flow is calculated using Brox's method [BBPW04] and optimized by [LMJ11].

6.3.2 Experimental Results on Challenging Scenes

We performed experiments on four representative sequences: *cars2*, *1R2TCR*, *store*, and *drive* to evaluate the performance of our method in dealing with varied camera motions. Before the name of each sequence, we add a prefix to notify the motions involved in each scene. The local objects are identified by natural numbers (e.g. 1, 2, 3, ..., N) and the camera is identified by the letter 'C'. The type of motions of objects and camera is indicated by a letter: 'R' for rotation, 'T' for translation, and 'P' for radial motion. For example, if a sequence is called 1R2TCRT it means that the first object rotates, the second translates and the camera motion consists of both rotation and translation.

[1] 1T2T3TCT-Cars2 sequence : This sequence is from the Berkeley dataset. Three cars are translating in the scene: one is near the center of the view, another is on the left side near to the image boundary, and the third one is on the right side partially occluded by another car. The camera is translating, see Fig.6.6 (a). From the original optical flow field (b), we can find only one car, while the others

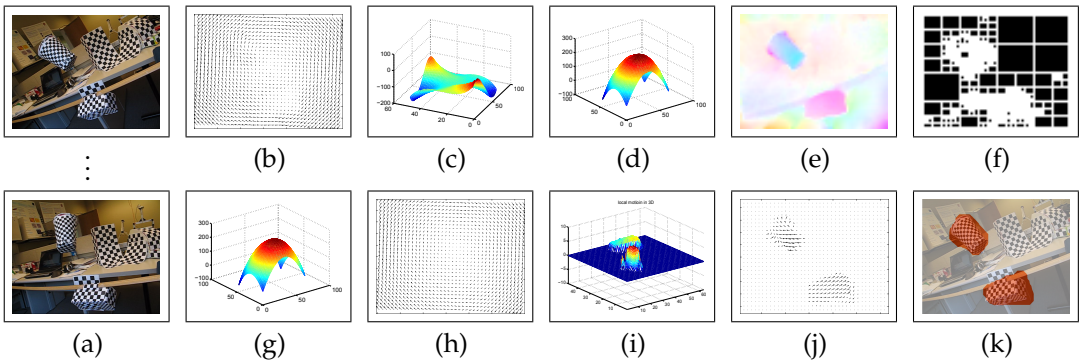


Figure 6.7: Scenario 2: checkerboard sequence. Please refer to Fig. 6.6 for the description of subfigures.

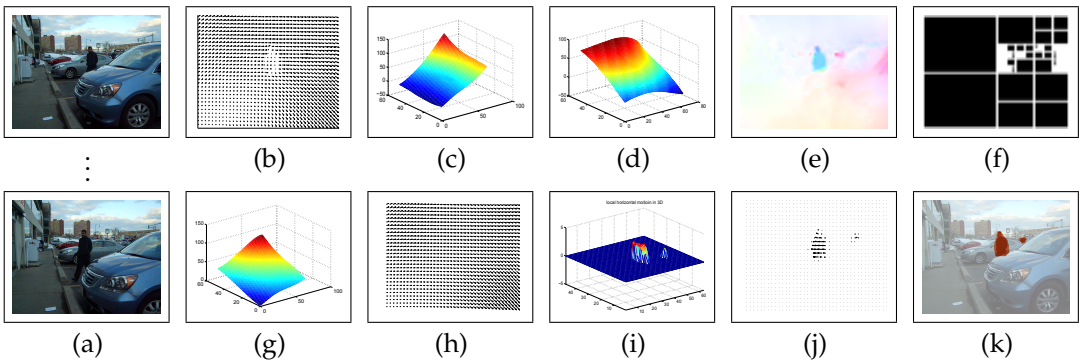


Figure 6.8: Scenario 3: store sequence. Please refer to Fig. 6.6 for the description of subfigures.

are drowned out by the inlier. The potential surfaces (c) and (d) showed the the homogeneous part of both inlier and outliers. In the reconstructed OOM (e), all the three moving cars are revealed clearly. Quad-Tree labels the heterogeneous part in (f). The estimated inlier potential surface (g) and its motion field (h) are quite smooth. Local object motions are recovered accurately in (i) and (j). This example demonstrates that our method cannot only segment motions well but also recover them to their true values. The OOM and Quad-Tree scheme enable our method to deal with challenging scenes.

[2] 1R2TCR-Checkerboard sequence : This data is from the Hopkins dataset. This scene involves three motions: a rotating view (inlier), in which a basket is rotating in the top left scene and a box is translating from left to the right in the bottom scene. This is also a very challenging scene, which comprises of multiple different motions. We use this scene to evaluate the performance of our method in dealing with camera rotation. The segmentation results are shown in Fig. 6.7.

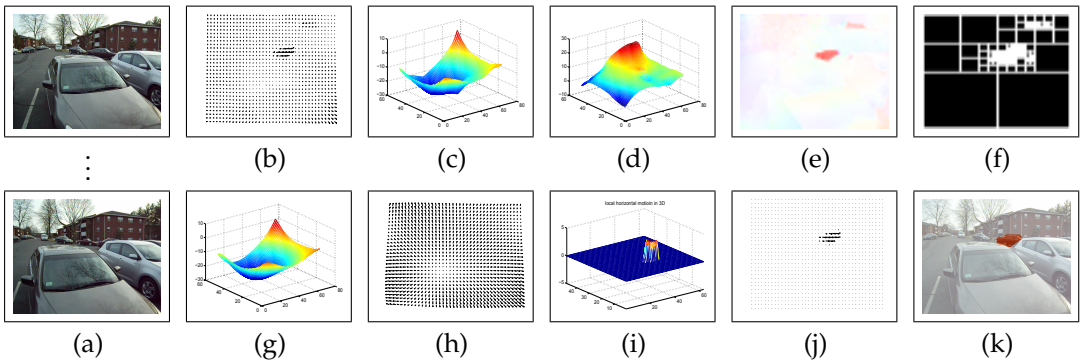


Figure 6.9: Scenario 4: drive sequence. Please refer to Fig. 6.6 for the description of subfigures.

From (a), we can see that the feature based methods place several checkerboards in the scene to obtain prominent feature points for motion segmentation. Our method, in contrast, does not rely on the strong features (corners and edges). From the original motion field (b), it is impossible to figure out what exact motions are involved in the scene. The OOM, however, has showed the foreground motions in (e) and the Quad-Tree detects them well in (f). From the estimated inlier potential surface (g) and its motion field (h), we can clearly find the rotation shape of the camera motion. This scene demonstrates the good performance of our method in dealing with camera rotation.

[3] 1T2TCRT-Store sequence : This sequence is from the Complex Background dataset. We use this sequence to evaluate the performance of our method in dealing with extremely challenging scenes with complex background and complex camera motions. It involves three motions: a translating and rotating camera motion, in which two people are walking across several cars: one is towards the store, and the other is in opposite direction. The segmentation results in Fig.6.8 show that the OOM represents the moving objects in (e) even when the object is very small and the background scene is complex with complex camera motions. The inlier estimated in (h) is quite smooth and the two object motions are recovered precisely in (j). The segmentation result (k) demonstrates the good performance of our method in dealing with extremely challenging scenes.

[4] 1TCP-Drive sequence : This sequence is also from the Complex Background dataset. We use this sequence to evaluate the performance of our method in dealing with radial motion. It involves two motions: a camera is moving towards the picture generating a radial motion; a car is turning around a corner generating a translation. The segmentation results are shown in Fig.6.9. The perceptive mo-

tion is revealed in the inlier potential surface (g) and its motion field (h) clearly. The OOM in (e) demonstrates object motions successfully, which verifies the effectiveness of our method in dealing with perspective motion.

6.3.3 Comparison with Existing Works

In this part, we compare our method with two types of motion segmentation methods: (1) dense based ones and (2) feature based ones. For the first type, we select six recent developed methods including: (1) joint inlier estimation and segmentation (GME-SEG) [CB11], (2-3) iterative estimation based on least-square (LS) [SHO00], and gradient decent (GD) [SSH05], (4) outlier rejection filter (Filter) [CB10], (5) RANSAC [FB81], and (6) the latest developed FOF [NHLM13], which is considered as the state-of-art algorithm in motion segmentation. In [NHLM13], they present two versions: (1) FOF, which uses optical flow only, and (2) FOF+color+prior, which combines optical flow, color appearance and a prior model together. The first five methods were implemented using the source code in [CB11]. We reported the performances of FOF and FOF+color+prior [NHLM13] directly. For the second type, we compare our method with three well-known algorithms including: GPCA with spectral clustering [VMS06], Local Subspace Affinity (LSA) [YP06] and RANSAC [FB81]. The source codes of these methods are provided in [TV07].

We used the F-measure to evaluate the performance of each algorithm, which is an important performance analysis parameter in binary classification.

$$F = \frac{2 \times R_c \times P_r}{R_c + P_r}. \tag{6.23}$$

F-measure considers both the precision P_r and the recall R_c [DJK+13]. For dense-based methods, as segmentation is performed on each individual pixel, F-measure is calculated using all pixels. While for feature based methods just selected key feature points are involved in F-measure calculation. Table 6.1 reports the F-measure of dense based methods on three benchmark datasets: Berkeley, Complex Background, and SegTrack. Table 6.2 reports the F-measure of feature based methods on Hopkins dataset. As less pixels are involved in F-measure determination in Table 6.2, our method obtains higher performance in Table 6.2 than that in Table 6.1 on the same image sequence.

Table 6.1: F-measure of existing dense based methods and ours.

Sequences	GME-SEG	LS	GD	Filter	RANSAC	FOF	FOR+color	Our
Cars1	78.33	**86.18**	18.01	82.87	60.42	47.81	50.84	76.38
Cars2	55.90	65.97	15.70	78.28	34.21	46.37	56.60	**83.13**
Cars3	65.21	79.43	22.29	74.56	35.80	67.18	73.57	**87.37**
Cars4	45.69	49.78	22.96	77.22	22.81	38.51	47.96	**84.42**
Cars5	54.67	61.93	33.08	81.17	25.24	64.85	70.94	**84.82**
Cars6	33.01	51.23	28.77	57.53	13.40	78.09	84.34	**85.71**
Cars7	37.89	36.36	36.92	60.47	13.79	37.63	42.92	**86.10**
Cars8	62.20	81.24	8.57	78.44	37.02	87.13	87.61	**90.68**
Cars9	72.99	**80.99**	17.97	68.19	54.69	68.99	66.38	77.42
Cars10	60.01	66.04	14.34	**90.95**	81.78	53.98	50.84	54.83
People1	34.11	38.32	40.76	71.84	12.06	56.76	69.53	**80.03**
People2	78.16	84.45	69.30	81.70	37.53	85.35	88.40	**89.81**
drive	8.14	6.30	41.18	32.40	5.76	30.13	61.80	**83.03**
forest	15.42	11.01	15.41	19.87	10.67	19.48	31.44	**35.71**
parking	33.20	21.57	43.84	62.29	17.47	43.47	73.19	**83.47**
store	14.39	10.94	32.10	29.32	9.68	28.46	70.74	**80.10**
traffic	14.77	15.80	34.55	15.04	15.49	66.08	71.24	**71.67**
birdfall2	9.39	3.84	0.99	64.00	3.25	68.68	75.69	**76.23**
girl	22.51	20.26	15.36	18.21	12.33	75.73	**81.95**	78.06
parachute	23.01	18.97	12.88	16.30	44.03	51.49	54.36	**86.72**
cheetah	21.33	14.93	43.59	12.05	9.85	12.68	22.31	**55.67**
penguin	10.34	18.84	15.34	5.53	18.66	14.74	20.71	**21.61**
monkeydog	22.29	20.74	16.46	18.93	12.31	10.79	18.62	**45.44**

Comparison with Dense Motion based Methods

From Table 6.1, we observe that our method achieves at the highest performance for almost all videos: it raises the F-measure by 10% – 30% on *Cars* 2, 3, 4, 7, and *People* 1 sequence in the Berkely dataset; around 10% on the *drive, parking*, and *store* sequences in the Complex Background dataset; and more than 20% on the *parachutte* and *monkeydog* sequences in the SegTrack dataset. This result is quite appealing even when videos contain extremely challenging scenes, such as the ones with complex camera motions, complex backgrounds, occlusions, etc.. The good quantitative results are confirmed by the good visual quality of the segmentation results. Some examples are shown in Fig. 6.10, where the last column shows the ground truth segmentation. In most cases, our segmentation agrees with the true object regions more than existing methods.

Compared to existing methods, the accuracy of our algorithm increases thanks to the introduction of the OOM and the Quad-Tree scheme. As shown in Figs.

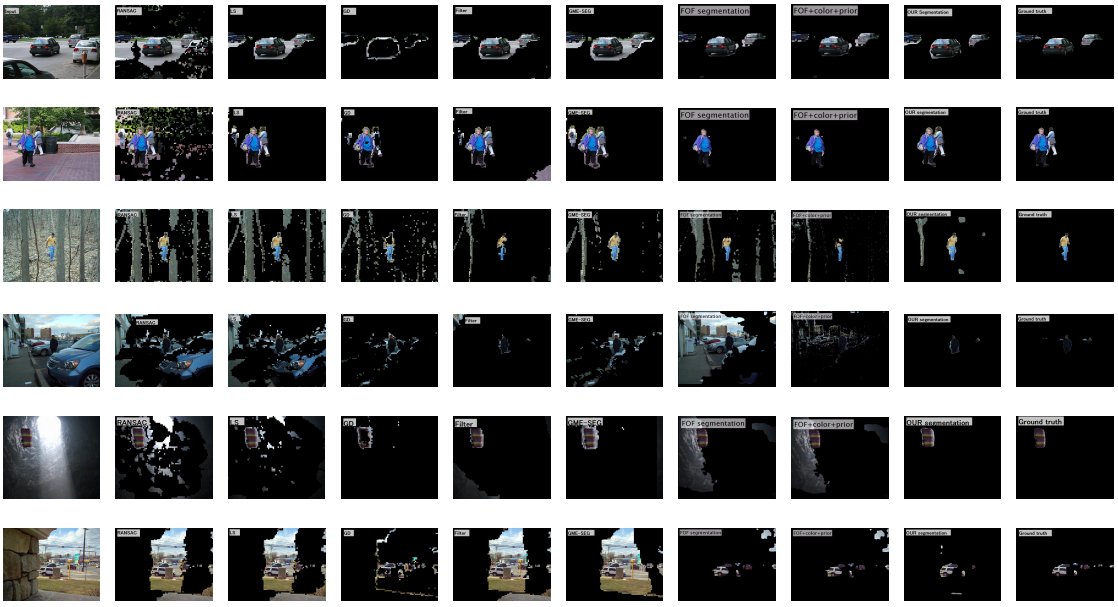


Figure 6.10: Segmentation results of six existing dense based methods and ours on challenging scenarios: (a) input sequences, from top to bottom: cars2, people2, forest, store, parachute, traffic, segmentation by (b) RANSAC [FB81], (c) LS [SHO00], (d) GD [SSH05], (e) Filter [CB10], (f) GME-SEG [CB11], (g) FOF [NHLM13], (h) FOF+color+prior [NHLM13], (i) our segmentation, (j) ground-truth segmentation.

6.6–6.8, all the regions containing moving objects and depth discontinuities haven been highlighted on OOM in (e) and labeled by Quad-Tree in (f). OOM separates inlier and outliers into homogeneous and heterogeneous part making the segmentation much easier and the Quad-Tree scheme based heterogeneous motion labeling ensures the high ability of our method in estimating smooth and accurate camera-induced image motion as well as to recover local object motions.

From Table 6.1, we can also find that our method obtains weak performance on the *cars* 1, 9, 10, and *girl* sequences. The main reason is that the *cars* 1, 9 and 10 sequences contain some weak and smooth objects motions, which make the inlier and outliers are very similar. For example, a big truck appears in cars 10 sequence, which is almost static for some frames. When these motions are mixed with the inlier, the motion field is still very smooth. Thus, they can only be presented in the curl-free and divergence-free components and not show up in the OOM, which makes our method fail to detect them, and the performance decreases. Another thing need to point out is that our method is performed based on the optical flow field. Its performance is influenced by the accuracy of the optical flow estimation. The *girl* sequence shows such an example. It captures a running girl in the sports

yard. Since the girl is moving very fast, some frames taken by a moving camera have been blurred terribly. The calculated optical flow fields become too noisy to identify moving objects. In this case, just using optical flow information is not enough. That's why FOF+color+prior, which utilizes additional information including color appearance and some prior models performs better than ours. In addition, all the methods appear less accurate on the three videos (*cheetah, penguin, monkeydog*) of the SegTrack dataset. That is because these videos actually contain multiple moving objects, but the ground truth intended for tracking analysis marks only one primary object as the foreground, causing all segmentation methods appear less accurate.

Comparison with Feature based Methods

From Table 6.2, we observe that our method performs better than or equal to existing methods on most sequences (*1R2TCR*, *Cars* 2, 3, 4, 5, 6, 7, 9, and *People* 1). However, for some sequences (*Cars* 1, 8), existing methods slightly outperform ours. That's because these methods use a special initialization procedure to segment the objects in the first frame. Our method, which makes no assumptions on the first frame, becomes less accurate. For example, in *Cars* 1 sequence, a heavy penalty is laid, where the optical flow of our method fails to detect the object in the first frame. As mentioned in the introduction, we argue that feature based methods suffer from several limitations: (1) they rely much on the existence of strong key features in the scene and depend on a robust feature extraction, tracking, and classification algorithms; (2) they are very sensitive to the incomplete trajectories and tracking errors caused by occlusion problems and noises; (3) The segmentation of sparse features are difficult to reveal the entire motion region in details. In contrast, our method neither cares about whether strong key features exist in the scene nor relies much on the feature extraction, tracking, or classification. Benefit from HHD based OOM and Quad-Tree scheme, our method is more robust to cope with complex scenes with complex camera motions, even when occlusions and noises present in the scene. What's more, our method preserves the shape of moving objects well and recovers their true motions, which are much valuable in vision based applications.

Table 6.2: F-measure of three feature based methods and ours.

Sequences	GPCA	RANSAC	LSA	Our
1R2TCR	84.83	92.54	97.05	**97.80**
Cars1	**100**	**100**	98.09	99.35
Cars2	84.79	92.91	92.09	**96.45**
Cars3	93.27	94.55	52.15	**96.30**
Cars4	**100**	**100**	98.85	**100**
Cars5	45.78	88.89	63.61	**98.06**
Cars6	94.12	94.12	92.31	**96.00**
Cars7	**83.33**	**83.33**	78.00	**83.33**
Cars8	**99.39**	96.89	98.78	98.77
Cars9	61.18	59.57	49.02	**72.73**
Cars10	52.83	53.46	52.49	**90.63**
People1	91.53	91.53	91.53	**96.43**
People2	**100**	84.75	97.96	**100**

6.3.4 Discussion

Although our algorithm achieves better performance for many scenarios in the experiments, it is not always effective. It is prone to failure in the following conditions:

- Camera is moving, the background is texture free and far away from the camera, while no moving objects present in the scene. In this case, our method may misidentify static scene objects as moving objects.
- Local object motions are very weak/small while the camera-induced image motion is strong. In this case, our method is difficult to identify between the true moving objects and static scene objects.
- The inlier occupies smaller region than outliers. This violates one of the two basic assumptions of HHD, which makes HHD fail to identify the homogeneous motion as one segment.

Example scenes for these three cases are shown in Fig. 6.11.

Next, we discuss the relationship between the first three algorithms (in Chapter 3–5) and this algorithm. While the first three algorithms are highly related with each other, this algorithm is a little bit different them. First, the research background of the first three algorithms and this algorithm is different. The first three algorithms uses face recognition as research background while the last algorithm is employed on motion segmentation. From the point view of feature

(a) (b) (c)

Figure 6.11: Three failure example scenes: (a) the first case; (b) the second one; (c) the third one.

learning, the first three algorithms are on feature learning from an image dataset while the last algorithm is about feature extraction from a video sequence. Thus, the problem definition and algorithm description are different from each other. But from the point view of image encoding, these four algorithms share the same Quad-Tree based image region partition, which is the core aspect of this dissertation.

6.4 Summary

In this chapter, we have presented a Quad-Tree based motion segmentation algorithm for feature learning in moving camera videos, where the object motions and camera motion are mixed and dependent with each other making the segmentation of moving objects difficult. As the Quad-Tree structure is sensitive to dynamic background, it is difficult to apply Quad-Tree based foreground and background segmentation in the mixed motion field directly. To solve this problem, we introduce an amended Helmholtz-Hodge Decomposition for camera motion compensation. HHD can represent three kinds of camera motions in an invariant way. After camera motion compensation, the heterogeneous motion caused by moving objects, depth discontinuities and noises are rested in a remainder. We construct an object-motion oriented map from this remainder and apply a data-driven Quad-Tree scheme to detect outliers on OOM. The Quad-Tree can not only detect object motions but also depth discontinuities in 3D scenes. After that, surface fitting based on a low-order polynomial function is employed for camera motion estimation and object motions can be recovered by subtracting camera motion from the original motion field. Extensive experimental results showed that our method obtained more robust segmentation results with much higher

accuracy than state-of-art algorithms.

This method can segment and recover camera motion and object motions from the mixed motion field, which is a fundamental task to many video based applications, such as the content-based retrieval, surveillance, human-computer interaction, action recognition, robot learning, etc.. However, this method is mainly on how to segment camera motion and object motions form the scene other than finding good features to represent them. It is more like a feature extraction algorithm rather than a feature learning algorithm. This is the limitation of this work. For the extension of this algorithm, we need to consider how to learn features to represent camera motion and object motions so that they can be more effectively employed in video based applications.

Chapter 7

Conclusion

7.1 Summary

This dissertation develops four Quad-Tree based image encoding methods towards four visual feature situations, which are highly related with each other and occupy a wide range of feature learning issues. Features can be extracted from the planar image surface or using hierarchical models, with large training data or small data, from static images or video sequences. The first three algorithms are developed for feature learning from static images (image dataset) while the last algorithm is for object motion segmentation in video sequences. We use face recognition as research background for the first three algorithms and motion segmentation for the last algorithm. The Quad-Tree partition is performed on a template image of an image dataset instead of raw images. The template image is generated according to the feature distribution of all the images in an image dataset. For each image dataset, we can generate a template image and then get a Quad-Tree partition for a given threshold. Since the generation of template image is data-adaptive, the Quad-Tree partition is also data-adaptive. In the following content, I will summary each algorithm and explain the relationship between four algorithms.

Algorithm 1: Feature learning from data-adaptive blocks decomposed by Quad-Tree

We develop a feature learning algorithm using data-adaptive blocks decomposed by Quad-Tree in Chapter 3. The features extracted from data-adaptive blocks can be directly transferred to machine learning applications or act as low-level fea-

tures and input to more sophisticated feature learning algorithms (e.g hierarchical learning). Instead of arbitrarily dividing the image region into fixed-size image blocks (patches), Quad-Tree partitions the image region into local subreigons of variant sizes according to the feature distribution in an image dataset. As it is difficult to find an appropriate threshold for Quad-Tree partition, we define a set of thresholds and get a set of Quad-Tree partitions accordingly. Good Quad-Tree partitions, which can provide high accuracy are selected by a 0–1 Knapsack algorithm for final decision fusion. Since the Quad-Tree partition is performed on the template image instead of raw images, the Quad-Tree based image region partition is adaptive to the image dataset.

Algorithm 2: Hierarchical feature learning using Quad-Tree structure of images

After local feature analysis, the next step is on how to combine local features together. New trends on feature learning utilize hierarchical models, which can generate high-level features form local features according to a hierarchical architecture. In this dissertation, we develop a hierarchical feature learning algorithm using Quad-Tree structure of images. In contrast to image pyramid based hierarchical models, which are complete tree structure for all images, Quad-Tree defines a data-adaptive image tree structure for hierarchical feature learning. Moreover, the parameters of local subregions can be defined according to the tree structure effectively make hierarchical feature learning more efficient. Through comparison with multiple methods using global, local features, canonical deep structure learning algorithm "Deep PCA", and the data-adaptive block based method (Algorithm 1) we observe that the Quad-Tree based hierarchical learning algorithm outperform the others.

Algorithm 3: Feature learning from enlarged training data encoded by Quad-Tree for Small Sample Size problem

While the first two algorithms are on feature learning with large training data, in some practical applications, the training data is very small due to the difficulty in data collection. The Small Sample Size (SSS) problem arises from the small training data compared to the large dimensionality of the feature space. To solve the SSS problem, we extend the training data by generating new samples encoded by Quad-Trees and then perform base classifier ensemble to improve the final recognition accuracy. In contrast to existing ensemble methods which are

performed using the original small training data, our method can generate more diverse base classifiers. Moreover, since existing ensemble methods suffer from the Diversity / Accuracy dilemma by integrating all base classifier together, we develop a base classifier selection algorithm using a tailored 0–1 Knapsack solution, which alleviates the dilemma effectively. We compare our method with many stat-of-the-art methods face recognition methods to demonstrate the strength of our method.

Algorithm 4: Feature learning in dynamic environments using Helmholtz-Hodge decomposition and Quad-Tree

While exiting visual feature learning algorithms can be applied to static images relatively easily, it is not so for video based applications, especially when the video is captured by a moving camera. In the last algorithm, we investigate feature learning in moving camera videos, where local object motions and camera motion are mixed and interdependent with each other making the segmentation of object motions difficult. We develop an object motion segmentation algorithm, which can segment and recover object motions from the scene effectively. As Quad-Tree is sensitive to the dynamic background, it is rather difficult to apply Quad-Tree based image region partition directly in dynamic environments. To solve this problem, we introduce an amended Helmholtz-Hodge Decomposition first for camera motion compensation first. After camera motion compensation, a data-driven Quad-Tree partition can be performed in the rested motion field for object motion segmentation. We evaluated our method on several motion segmentation benchmarks. Experiential results demonstrate that our method improves the segmentation accuracy of state-of-the-art method by 5%–20%.

Relationship between Four Algorithms

We need to explicitly mention that the first three algorithms are highly related with each other while the fourth algorithm is different from them in some sense. The first three algorithms share the same research background (face recognition). Their problem definition and algorithm description are also similar. However, the research background of the last algorithm is on motion segmentation. It investigates feature extraction from video sequences rather than image dataset. Thus, the problem definition and algorithm description are also different from the first three algorithms. However, these four algorithms share the same Quad-

Tree based image encoding, which is the core-aspect of this dissertation.

Specifically, Algorithm 1 is like a warm-up algorithm. It is the beginning of using Quad-Tree based image region partition to learn data-adaptive blocks for feature learning. Algorithm 2 is the improvement of Algorithm 1. In Algorithm 2, we employ a hierarchical architecture encoded by Quad-Tree structure to learn more abstract features and to assign different weights to local subregions for combination. We can say that the Quad-Tree based method becomes mature in Algorithm 2. The first two algorithms are on feature learning from large training data. Algorithm 3 extends feature learning to another challenging situation, where the training data is very small. Algorithm 3 is the extension of Algorithm 1. It deal with the Small Sample Size problem and makes the Quad-Tree based method more robust. While the first three algorithms are on feature learning from static image dataset, the fourth algorithm is about object motion segmentation in video sequences. It is the first try of extending feature learning from static images to video sequences since motion segmentation is the fundamental task to many video based applications, e.g. action recognition. However, we need to mention that we didnot investigate how to represent object motions in a more effective way for advanced applications. Thus, the fourth algorithm is more like a feature extraction algorithm rather than feature learning. I will explain this limitation and the possible extensions in the future work.

7.2 Future Directions

Although our methods can be employed in many feature learning situations, they still have some limitations. I will discuss the limitations and the possible extensions of these methods, which will be the future work of this dissertation.

More Variations to Bear

For the feature Learning from image dataset, we use face recognition as research background. We find that our Quad-Tree based image encoding methods can deal with local deformations well e.g. facial expressions, illumination changes, occlusion, etc.. However, it is not true for large variations (e.g. head positions). Thus, our method still needs some data registration to help get the face area for face recognition. This limits the ability of our method in dealing with object recognition and classification problems. For the future work, we need to improve our

methods by adding more constraints to Quad-Tree partition make them more robust to large variations. We expect our methods can work for wild databases in the future.

More Informative Features and Classifiers

Our implementation of visual feature learning algorithms mostly relies on appearance-based visual feature extraction. While appearance-based features can supply discriminant information to describe natural objects with stable and distinctive appearance, they are still insufficient to represent a large number of object classes with varying appearance but distinctive shapes. Therefore, it is desirable to utilize some local feature descriptors such as SIFT to encode image features reflecting local shape information. These local shape descriptors can describe local structures of images and can be widely used in object categorization, recognition, classification applications.

Another possible issue for improvement are classifiers. This dissertation focuses on Quad-Tree based image encoding methods. The developed image encoding methods are independent with classifiers. Currently, we just use PCA+LDA as classifiers for feature extraction. Actually, we can also use some other kinds of feature extraction methods. The reason why we use PCA+LDA is that it is recognized as a base line algorithm for face recognition and has been extensively studied for many object recognition/classification tasks. Using such base line classifiers without any other constraints can better demonstrate the effectiveness of our image encoding methods. For the future work, we can try more sophisticated classifiers to make the learning procedure more robust and powerful.

More Sophisticated Hierarchical Models for Feature Learning

In this dissertation, we explores the possibilities of employing image spatial structure in hierarchical learning. Our Quad-Tree based image spatial structure changes the local connectivity of subregions and makes the high-level feature integration closely related to image inputs. However, we currently mainly incorporate the Quad-Tree structure into the deep PCA algorithm to examine more precisely the impact of the image spatial structure in hierarchical learning itself. Although our hierarchical model has demonstrate impressive results in real-world image classification tasks. This is just the beginning of a full exploration of how

and why image spatial structure can help hierarchical learning. For the future work, we will explore the benefits or Quad-Tree based image structure in other hierarchical models such as Convolutional Neural Networks (CNNs) and Dynamic Belief Networks (DBNs). We aim to build more robust and flexible deep architectures to make hierarchical learning more powerful.

Learning Issues for Ensemble Learning

The ensemble method developed in this dissertation provides a new direction of applying ensemble to deal with the SSS problem. That is the data augmentation based ensemble learning by generating new samples. To generate new samples, our method relies on the feature distribution for data generation, which acts as the prior-knowledge for random matrix introduction. However, this feature distribution is usually unknown for given face databases, which may make our method generate inappropriate face images and thus influence the performance of our method. For the future work, we are going to develop a data-adaptive feature distribution learning algorithm that can estimate the feature distribution of a given database accurately and data-deceptively.

Another potential feasible way to improve the developed ensemble learning algorithm is to improve the ensemble rule for base classifier integration. Currently, we just utilize the majority voting for ensemble, which is a very basic rule for base classifier integration. Although it is the most commonly used integration rule thanking to its simplicity, it is still of limited ability to take advantage of different base classifier in classification. In the future, we will utilize more sophisticated ensemble rules for base classifier integration, such as weighted voting.

Video based Feature Learning

In Chapter 6, we developed a motion segmentation method using HHD and Quad-Tree, where HHD is used for camera motion compensation and Quad-Tree for object motion segmentation. Quad-Tree can segment both object motions and depth discontinuities from the scene, which can be further used for video based applications, such as action recognition, content based surveillance, etc.. However, the Quad-Tree strategy can not specify between the object motions and depth discontinuities. Although in Chapter 6, we use a low polynomial function based surface fitting for camera motion estimation, which can compensate depth discontinuities from the scene, it is still hard to define an appropriate degree for

the polynomial function. This is the limitation of this work. For the extension of this algorithm, we need to consider to develop a more effective way to specify between object motions and depth discontinuities and learn how to represent object motions using more salient features so that they can be more effectively employed in video based applications. Towards this goal, the possible directions are as follows: we are planning to develop object motion descriptors and to construct spatial-temporal models to represent them so that they can be specified from the scene and used for video based applications. Towards this goal, we can first model the spatial dynamics by using the tree structure. Then the spatial dynamics can be transformed to temporal dynamics (structured time signals) based on Quad-Tree comparison along time axis. Finally, the time signals can be transformed, formalized and classified for video based applications such as action recognition, etc..

Other Applications

Face recognition and motion segmentation are investigated in this dissertation for algorithm description and performance evaluation. Thanking to the data-adaptive property, the developed visual feature learning algorithms can be widely applied to many computer vision applications. A few examples are listed as follows:

- **Biometric Recognition** In the dissertation, we uses faces as cues for subject recognition. In the field of biometrics, there are many other kinds of cues that can be used to recognize humans, such as finger print, periocular regions, etc.. The biometric recognition systems using these cues are very similar with face recognition. They also need to consider how to extract effective visual features, how to generate classifiers, and how to utilize classifiers for recognition. The visual feature learning algorithms developed in this dissertation, such as the region-specific feature extraction algorithm, the hierarchical feature learning algorithm, and the ensemble learning algorithm can be applied to these recognition applications naturally.

- **Other Small Sample Size Classification Problems** Small sample size problem is a common problem in the field of machine learning based applications, such as pattern recognition, computer vision etc.. The data augmentation based ensemble algorithm algorithm developed in this dissertation provides a new direction of utilizing ensemble learning to deal with SSS

problem. Besides SSS face recognition, it can be widely applied to deal with other SSS classification/recognition problems.

- **Video based Applications**: Recently, many video data are taken by a moving observer, such as hand-held or vehicle-mounted cameras. The camera motion plays an important role in global view based applications such as video stabilization, structure-from-motion, etc., while local object motions are what we expected in many content based applications such as action recognition, content-based retrieval, etc.. Our Quad-Tree based motion segmentation can estimate the camera motion accurately and recovers object motions to their true values. Extending the current framework to video analysis problems will be interesting directions to explore. Next, we use video stabilization, video based face recognition and action recognition as examples to explain such applications.

  - **Video Stabilization:** Video stabilization is an important video enhancement technology, which aims at improving the quality of moving camera videos by removing undesirable shaky camera motions. It has been gaining stable importance thanking to the increasing development of mobile devices. The first step of video stabilization is the camera motion estimation, and then we can remove some shaky camera motion by smoothing the camera path and use image warping method to generate stabilized new videos. The Quad-Tree based motion segmentation method (developed in Chapter 6) can estimate the camera path and thus can be used for video Stabilization in the future.
  - **Video based Face Recognition:** Static image based face recognition has been extensively studied in the last two decades. Many researchers devote themselves on video based face recognition, which can be used for facial expression recognition and human emotion understanding. Face images in videos can have more variations since they can be taken in more complex situations. For example, in some cases, we just want to have face areas for facial expression recognition and human emotion understanding. However, the face motions can be mixed with head motions, which we would like to remove. In this case, the head motion can be considered as the global motion while the face motion can be considered as local object motions. We can use the motion segmentation algorithm to remove head motions from the mixed motion field

and recover the face motion to their true value. After that face motions can be used for facial expression recognition and human emotion understanding.

– **Action Recognition:** The performance of action recognition algorithms usually relies on the extraction and representation of object motions from video sequences. In moving camera videos, we also need to compensate camera motion from the scene so that object motions can be represented more accurately. The Quad-Tree based motion segmentation method can be used to remove camera motions and to recover object motions to their true values from the scenes. As mentioned in the extension of 'video based feature learning', we can also consider to use Quad-Tree to build up spatio-temporal models for action recognition.

we believe the proposed method can contribute to a large number of video based application including but not limited to the situations where we mentioned above.

Bibliography

[AGA09] N. Alyüz, B. Gökberk, and L. Akarun. Regional registration and curvature descriptors for expression resistant 3D face recognition. *SIU*, pages 544–547, 2009.

[Ald13] A. Aldroubi. A review of subspace segmentation: Problem, nonlinear approximations, and applications to motion segmentation. *ISRN Signal Process*, pages 1–13, 2013.

[Aro05] Tomasz Arodz. Margin-based diversity measures for ensemble classifiers. *Advances in Soft Computing Volume*, 30:71–78, 2005.

[Ase07] Agarwala Aseem. Efficient gradient-domain compositing using quadtrees. *ACM Transactions on Graphics*, 26(3), 2007.

[ASS+] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Suesstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11).

[Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, 1996.

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36, 2004.

[BG14] Amrita Biswas and M K Ghose. Expression invariant face recognition using DWT SIFT features. *International Journal of Computer Applications*, 92(2):30–32, 2014.

[BHK97] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *PAMI*, 20(7):711–720, 1997.

[BKJ13] K. Bonnen, B. F. Klare, and A. K. Jain. Component-based representation in automated face recognition. *IEEE Transactions on Information Forensics and Security*, 8(1):239–253, 2013.

[BL07] Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. *MIT Press*, 2007.

[BM10] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010.

[BNPB13] Harsh Bhatia, Gregory Norgard, Valerio Pascucci, and Peer-Timo Bremer. The helmholtz-hodge decomposition - a survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 19(8):1386–1404, 2013.

[BPF09] C. Boehnen, T. Peters, and P. Flynn. 3D signatures for fast 3D face recognition. *ICB*, pages 12–21, 2009.

[BSL+11] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliskiv. A database and evaluation methodology for optical flow. *Internations Journal of Computer Vision*, 92:1–31, 2011.

[CB10] Y. M. Chen and I. V. Bajic. Motion vector outlier rejection cascade for global motion estimation. *IEEE Signal Processing Letters*, 17(2):197–200, 2010.

[CB11] Y. M. Chen and I. V. Bajic. A joint approach to global motion estimation and motion segmentation from a coarsely sampled motion vector field. *IEEE Transactions on Circuits Systtem and Video Technology*, 21(9):1316–1328, 2011.

[CFCNL14] C. Couprie, F F. Clment, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. In *International Conference on Learning Representations (ICLR)*, 2014.

[CHH+07] D. Cai, X. He, Y. Hu, J. Han, and T. Huang. Learning a spatially smooth subspace for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[CK98] J. Costeira and T. Kanade. A multibody factorizationmethod for independently moving objects. *International Journal of Computer Vision (IJCV)*, 29(3):159–179, 1998.

[CLZ04] S. Chen, J. Liu, and Zhi-Hua Zhou. Making FLDA applicable to face recognition with one sample per person. *Pattern Recognition*, 37(7):1553–1555, 2004.

[CMES10] G. R. Conde-Marquez, H. J. Escalante, and E. Sucar. Simplified Quadtree image segmentation for image annotation. In E. Sucar and H. J. Escalante, editors, *Proceedings of the 2010 Automatic Image Annotation and Retrieval Workshop*, volume 719, pages 24–34. 2010.

[CSU03] CSU. CSU face identification evaluation system, 2003.

[CW08] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML)*, pages 160–167. ACM, 2008.

[CZZ04] S. Chen, D. Zhang, and Zhi-Hua Zhou. Enhanced (PC)2A for face recognition with one training image per person. *Pattern Recognition Letters*, 25(10):1173–1181, 2004.

[DHG+10] W. Deng, J. Hu, J. Guo, W. Cai, and D. Feng. Robust, accurate and efficient face recognition from a single training image: A uniform pursuit approach. *Pattern Recognition*, 43(5):1748–1762, 2010.

[DJK+13] K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hullermeier. Optimizing the F-measure in multi-label classiiňĄcation: Plug-in rule approach versus structured loss minimization. In *International Conference on Machine Learning (ICML)*, pages 1130–1138, 2013.

[DJL+10] M. N. Dailey, C. Joyce, M. J. Lyons, M. Kamachi, H. Ishi, and J. Gyoba. Evidence and a computational explanation of cultural differences in facial expression recognition. *Emotion*, 10(6):874–893, 2010.

[DLHG12] W. Deng, Yebin Liub, Jiani Hua, and Jun Guoa. The small sample size problem of ICA: A comparative study and analysis. *Pattern Recognition*, 2012.

[DZZP10] Wei Di, Lei Zhang, David Zhang, and Quan Pan. Studies on hyperspectral face recognition in visible spectrum with feature band selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(6):1354–1361, 2010.

[FB81] M.A. Fischler and R.C. Bolles. RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 26:381–395, 1981.

[FBL09] S. Fidler, M. Boben, and A. Leonardis. Hierarchical compositional representations of object structure. In S. Dickinson, Leonardis A., B. Schiele, and M.J. Tarr, editors, *Object Categorization: Computer and Human Vision Perspectives*, pages 196–215. Cambridge University Press, 2009.

[FCNL13] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1915–1929, 2013.

[GA04] R. Gottumukkal and V. K. Asari. An improved face recognition technique based on modular PCA approach. *Pattern Recognition Letters*, 25(4):429–436, 2004.

[GBK01] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(6):643–660, 2001.

[GDDM14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[Has86] Johan Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18the annual ACM Symposium on Theory of Computing*, pages 6–20, 1986.

[HES+08] Raia Hadsell, Ayse Erkan, Pierre Sermanet, Marco Scoffier, Urs Muller, and Yann LeCun. Deep belief net learning in a long-range vision system for autonomous off-road driving. In *Proceeding of Intelligent Robots and Systems (IROS08)*, pages 628–633, 2008.

[HLLM12] G. B. Huang, H. Lee, and E. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2518–2525, 2012.

[HO00] A. Hyvarinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13:411–430, 2000.

[HO06] G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[HOT06] Goeffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[HYH+05] X. He, S. Yan, Y. Hu, P. Niyogi, and H. J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(3):328–340, 2005.

[IA98] Michal Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):577–589, 1998.

[Ich99] Naoyuki Ichimura. Motion segmentation based on factorization method and discriminant criterion. In *ICCV*, pages 600–605, 1999.

[JDaHW14] Yi Hong Jifeng Dai and, Wenze Hu, and Ying Nian Wu. Unsupervised learning of dictionaries of hierarchical compositional models. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[JM] Vidit Jain and Amitabha Mukherjee. The indian face database. *http://vis-w.cs.umass.edu/ vidit/IndianFaceDatabase/*.

[KG12] H.C. Kim and Z. Ghahramani. Bayesian classifier combination. In *Proceedings of International Conference on Artificial Intelligence and Statistcs (AISTATS)*, 2012.

[KM02] K. Kanatani and C. Matsunaga. Estimating the number of independent motions for multibody motion segmentation. In *ECCV*, pages 25–31, 2002.

[KPB+13] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, C. Gulcehre, R. Memisevic, P. Vincent, A. Courville, and Y. Bengio. Combining modality specific deep neural networks for emotion recognition in video. In *15th ACM International Conference on Multimodal Interaction (ICMI)*, 2013.

[KPS+04] Daniel A. Keim, Christian Panse, Jorn Schneidewind, Mike Sips, and Universiat Konstanz. Geo-spatial data viewer: From familiar land-covering to arbitrary distorted geo-spatial quadtree maps. In *In WSCG 2004, The 12-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2004.

[KSH12] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classi cation with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*, 2012.

[KSS+13] Meina Kan, Shiguang Shan, Yu Su, Dong Xu, and Xilin Chen. Adaptive discriminant learning for face recognition. *Pattern Recognition*, 46(9):2497–2509, 2013.

[LBBH98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2234, 1998.

[Lee10] Honglak Lee. *Unsupervised Feature Learning via Sparse Hierarchical Representations*. PhD thesis, Stanford University, 2010.

[LMJ11] X.F. Liang, P.W. McOwan, and A. Johnston. A biologically inspired framework for spatial and spectral velocity estimations. *Journal of the Optical Society of America A*, 28(4):713–723, 2011.

[LPJ11] Z. Li, U. Park, and A. K. Jain. A discriminative model for age invariant face recognition. *IEEE Transactions on Information Forensics and Security*, 6(3):1028–1037, 2011.

[LPV08] H. Lu, K. Plataniotis, and A. Venetsanopoulos. MPCA: Multilinear principal component analysis of tensor objects. *IEEE Transactions on Neural Networks*, 19(1):18–39, 2008.

[LT10] J. Lu and Y. Tan. A doubly weighted approach for appearance-based subspace learning methods. *IEEE Transactions on Information Forensics and Security*, 5(1):71–81, 2010.

[LTW11] Jiwen Lu, Yap-Peng Tan, and Gang Wang. Discriminative multimanifold analysis for face recognition from a single training sample per person. In *International Conference on Computer Vision (ICCV)*, 2011.

[LTW13] Jiwen Lu, Yap-Peng Tan, and Gang Wang. Discriminative multimanifold analysis for face recognition from a single training sample per person. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(1):39–51, 2013.

[LZM14] Xuefeng Liang, Cuicui Zhang, and Takashi Matsuyama. Inlier estimation for moving camera motion segmentation. In *Proceeding 12th Asian Conference on Computer Vision (ACCV)*, 2014.

[MB98] A. M. Martinez and R. Benavente. The AR face database. *CVC Technical Report*, 1998.

[MGKR12] K. Manikantan, V. Govindarajan, V. V. S. Sasi Kiran, and S. Ramachandran. Face recognition using block-based DCT feature extraction. *Journal of Advanced Computer Science and Technology*, 1(4):266–283, 2012.

[MK01] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(2):228–233, 2001.

[MMCS11] Jonathan Masci, Ueli Meier, Dan Ciresan, and Juergen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, 2011.

[MS12] B. Mitchell and J. Sheppard. Deep structure learning: beyond connectionist approaches. In *The 11th International Conference on Machine Learning and Applications (ICMLA'12)*, 2012.

[MSB+10] D. Marpe, H. Schwarz, S. Bosse, B. Bross, P. Helle, T. Hinz, H. Kirchhoffer, H. Lakshman, T. Nguyen, S. Oudin, M. Siekmann, K. Suhring, M. Winken, and T. Wiegand. Video compression using nested quadtreestructures, leaf merging, and improved techniques for motion representation and entropy coding. *IEEE Transactions on Circuits System and Video Technology*, 20:1676–1687, 2010.

[NH98] A. V. Nefian and M. H. Hayes. Hidden markov models for face recognition. *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, pages 2721–2724, 1998.

[NHLM13] Manjunath Narayana, Allen Hanson, and Erik Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. In *ICCV*, 2013.

[OMB14] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014.

[OPM+11] Milos Oravec, Jarmila Pavlovicova, Jan Mazanec, Lubos Omelina, Matej Feder, and Jozef Ban. Efficiency of recognition methods for single sample per person based face recognition. *Refinements and New Ideas in Face Recognition, Dr. Peter Corcoran (Ed.), ISBN: 978-953-307-368-2, InTech*, pages 1885–1896, 2011.

[PJR11] U. Park, R. Jillela, and A. Ross. Periocular biometrics in the visible spectrum. *IEEE Transactions on Information Forensics and Security*, 6:96–106, 2011.

[PMRR00] P. Jonathon Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(10):1090–1104, 2000.

[PP03] K. Polthier and E. Preu. Identifying vector fields singularities using

a discrete Hodge decomposition. *In: Hege, H.C.,Polthier, K. (Eds.), Visualization and Mathematics III.*, pages 123–134, 2003.

[PWB+12] V. M. Patel, T. Wu, S. Biswas, P. J. Philips, and R. Chellappa. Dictionary-based face recognition under variable lighting and pose. *IEEE Transactions on Information Forensics and Security*, 7(3):954–956, 2012.

[QB13] C. Qian and I. V. Bajic. Global motion estimation under translation-zoom ambiguity. In *Proc. IEEE PacRim*, pages 46–51, 2013.

[QSBS10] C. C. Queirolo, L. Silva, O. R. P. Bellon, and M. P. Segundo. 3D face recognition using simulated annealing and the surface interpenetration measure. *PAMI*, 32:206–219, 2010.

[RGEW86] D. E. Rumelhart, Hinton G. E., and R. J. Williams. Learning representations by backpropagating errors. *Nature*, 333:533–536, 1986.

[RMM11] V. Ramanathan, S.S. Mishra, and P. Mitra. Quadtree decomposition based extended vector space model for image retrieval. In *IEEE Winter Conference on Applicaitons of Computer Vision (WACV)*, 2011.

[RPCL07] Marc Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann LeCun. Efficient learning of sparse representations with an energy-based model. In B. Scholkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19 (NIPS06)*, pages 1137–1144. MIT Press, 2007.

[Sam06] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. 2006.

[SaXW13] Feng Shi, Zhong Zhou andJiangjian Xiao, and WeiWu. Robust trajectory clustering for motion segmentation. In *ICCV*, 2013.

[SD02] M. Skurichina and R. P. W. Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis and Applications*, 5(2):121–135, 2002.

[SEZ+14] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection

using convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[SH94] Ferdinando S. Samaria and Andy C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.

[SHO00] A. Smolic, M. Hoeynck, and J.-R. Ohm. Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 application. In *ICIP*, pages 271–274, 2000.

[SHR10] M. Saquib Sarfraz, Olaf Hellwich, and Zahid Riaz. Feature extraction and representation for face recognition. In Milos Oravec, editor, *Face Recognition*, pages 1–20. InTech., 2010.

[SJZZ10] N. Sun, Z. Ji, C. Zou, and L. Zhao. Two-dimensional canonical correlation analysis and its application in small sample size face recognition. *Neural Computing and Applications*, 19:377–382, 2010.

[SK04] Yasuyuki Sugaya and Kenichi Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. *Workshop on Statistical Methods in Video Processing*, pages 1–2, 2004.

[SMH07] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey E. Hinton. Restricted Boltzmann machines for collaborative filtering. In Zoubin Ghahramani, editor, *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML07)*, pages 791–798. ACM, 2007.

[Spr11] L. Spreeuwers. Fast and accurate 3D face recognition using registration to an intrinsic coordinate system and fusion of multiple region classifiers. *International Journal of Computer Vision (IJCV)*, 93(3):389–414, 2011.

[SSCG09] Y. Su, S. Shan, X. Chen, and W. Gao. Hierarchical ensemble of global and local classifiers for face recognition. *IEEE Transactions on Image Processing*, 18(8):1885–1896, 2009.

[SSCG10] Y. Su, S. Shan, X. Chen, and W. Gao. Adaptive generic learning for face recognition from a single sample per person. In *IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), pages 2699–2706, 2010.

[SSH05] Y. Su, M.-T. Sun, and V. Hsu. Global motion estimation from coarsely sampled motion vector field and the applications. *IEEE Transactions on Circuits Systtem and Video Technology.*, 15(2):232–242, 2005.

[TCZZ05] X. Tan, S. Chen, Zhi-Hua Zhou, and F. Zhang. Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft k-NN ensemble. *IEEE Transactions on Neural Networks*, 16(4):875–886, 2005.

[TE10] Berkay Topcu and Hakan Erdogan. Decision fusion for patch-based face recognition. In *International Conference on Pattern Recognition (ICPR)*, pages 1348–1351, 2010.

[TFM10] David Tsai, Matthew Flagg, and James M.Rehg. Motion coherent tracking with multi-label mrf optimization. In *BMVC*, 2010.

[TLHD03] Y. Tong, S. Lombeyda, A.N. Hirani, and M. Desbrum. Discrete multiscale vector field decomposition. *ACM SIGRAPH*, pages 27–31, 2003.

[TP91] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1), 1991.

[TSY06] E. K. Tang, P. N. Suganthan, and X. Yao. An analysis of diversity measures. *Machine Learning*, 65(1):247–271, 2006.

[TV07] Roberto Tron and R. Vidal. A benchmark for the comparison of 3D motion segmentation algorithms. In *CVPR*, 2007.

[VLBM08] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Andrew McCallum and Sam Roweis, editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML)*, pages 1096–1103. Omnipress, 2008.

[VMS06] R. Vidal, Yi Ma, and Stefano Soatto. Two-view multibody structure from motion. *International Journal of Computer Vision (IJCV)*, 68(1):7–25, 2006.

[WOY14] W. Wang, B. C. Ooi, and X. Y. Yang. Effective multi-modal retrieval based on stacked auto-encoders. In *International Conference on Very Large Data Bases*, 2014.

[WT06] Xiaogang Wang and Xiaoou Tang. Random sampling for subspace face recognition. *International Journal of Computer Vision (IJCV)*, 70(1):91–104, 2006.

[WUKZ12] I Gede Pasek Suta Wijaya, Keiichi Uchimura, Gou Koutaki, and Cuicui Zhang. Robust face recognition using Wavelet and DCT based lighting normalization, and shifting-mean LDA. In *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 343–350, 2012.

[WZ02] J. Wu and Zhi-Hua Zhou. Face recognition with one training image per person. *Pattern Recognition Letters*, 23(14):1711–1719, 2002.

[Yan02] M. H. Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *IEEE International Conference on Face and Gesture Recognition (FG)*, pages 215–220, 2002.

[Yan12] Wuyi Yang. Regularized complete linear discriminant analysis for small sample size problems. *Communications in Computer and Information Science*, 304:67–73, 2012.

[YP06] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and nondegenerate. In *ECCV*, pages 94–106, 2006.

[YZFY04] J. Yang, D. Zhang, A. Frangi, and J. Yang. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(1):131–137, 2004.

[ZCZ05] D. Zhang, S. Chen, and Zhi-Hua Zhou. A new face recognition method based on SVD perturbation for single example image per person. *Applied Mathematics and Computation*, 163(2):895–907, 2005.

[ZLC09] Y. Zhu, J. Liu, and S. Chen. Semi-random subspace method for face recognition. *Journal of Image and Vision Computing*, 27:1358–1370, 2009.

[ZLM12] Cuicui Zhang, Xuefeng Liang, and Takashi Matsuyama. Multi-subregion face recognition using coarse-to-fine Quad-tree decomposition. In *International Conference on Pattern Recognition (ICPR)*, pages 1004–1007, 2012.

[ZLM13] Cuicui Zhang, Xuefeng Liang, and Takashi Matsuyama. Small sample size f ace recognition using random quad-tree based ensemble algorithm. In *The 5th International Conference on Imaging for Crime Detection and Prevention (ICDP)*, 2013.

[ZLM14a] Cuicui Zhang, Xuefeng Liang, and Takashi Matsuyama. Generic learning-based ensemble framework for small sample size face recognition in multi-camera networks. *Sensors*, 14(12):23509–23538, 2014.

[ZLM14b] Cuicui Zhang, Xuefeng Liang, and Takashi Matsuyama. Multi-depth deep feature learning for face recognition. In *International Conference on Informatics, Networking and Intelligent Computing (INIC)*, 2014.

[ZN11] Vasileios Zografos and Klas Nordberg. Fast and accurate motion segmentation using linear combination of views. In *BMVC*, 2011.

[ZUKZ10] Cuicui Zhang, Keiichi Uchimura, Gou Koutaki, and Cai Ming Zhang. 3D face recognition using multi-level multi-feature fusion. *PSIVT*, 2010.

[ZZHS12] P. Zhu, L. Zhang, Q. Hu, and S. Shiu. Multi-scale patch based collaborative representation for face recognition with margin distribution optimization. In *European Conference on Computer Vision (ECCV)*, 2012.

Related Publications

Journal Papers

- Cuicui Zhang, Xuefeng Liang, and Takashi Matsuyama, "Mixed-Motion Segmentation using Helmholtz Decomposition", IPSJ Transactions on Computer Vision and Applications, Vol.5 55-59, 2013.

- Cuicui Zhang, Xuefeng Liang, and Takashi Matsuyama, "Generic Learning-Based Ensemble Framework for Small Sample Size Face Recognition in Multi-Camera Networks", Sensors, Vol. 14, No. 12, pp: 23509-23538, 2014.12. (IF: 2.048).

International Conference Papers

- Cuicui Zhang, Xuefeng Liang, and Takashi Matsuyama, "Multi-subregion Face Recognition using Coarse-to-fine Quad-tree Decomposition", International Conference on Pattern Recognition (ICPR) (*Oral presentation, Acceptance rate: 15%*), Tsukuba, Japan, Nov. 2012.

- Cuicui Zhang, Xuefeng Liang, and Takashi Matsuyama, "Small Sample Size Face Recognition using Random Quad-Tree based Ensemble Algorithm" International Conference on Imaging for Crime Detection and Prevention (ICDP), London, UK, 2013.

- Xuefeng Liang, Cuicui Zhang, and Takashi Matsuyama, "Inlier Estimation for Moving Camera Motion Segmentation", Proceeding 12th Asian Conference on Computer Vision (ACCV 2014), (*Oral presentation, Acceptance rate: 3.8%*), Singapore, November, 2014.

- Cuicui Zhang, Xuefeng Liang, Takashi Matsuyama, "Multi-Depth Deep Feature Learning for Face Recognition", International Conference on Informatics, Networking and Intelligent Computing (INIC 2014), Shenzhen, China, November, 2014.

Awards

- Cuicui Zhang, Xuefeng Liang, and Takshi Matsuyama, "What are you looking for on faces?". **Outstanding Research Award** at Kyoto University ICT innovation 2013.
- Cuicui Zhang, Xuefeng Liang, and Takshi Matsuyama, "Mixed-Motion Segmentation using Helmholtz Decomposition, **Outstanding Paper Award** at the 16th Meeting on Image Recognition and Understanding (MIRU), Tokyo, Japan, August, 2013.