3D FACE RECOGNITION WITH WIRELESS TRANSPORTATION

A Dissertation

by

LE ZOU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2007

Major Subject: Computer Engineering

3D FACE RECOGNITION WITH WIRELESS TRANSPORTATION

A Dissertation

by

LE ZOU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Mi Lu
	Zixiang Xiong
Committee Members,	Scott Miller
	Xi Zhang
	Andreas Klappenecker
Head of Department,	Costas N. Georghiades

August 2007

Major Subject: Computer Engineering

ABSTRACT

3D Face Recognition with Wireless Transportation. (August 2007)

Le Zou, B.S., Southeast University;

M.S., Southeast University

Co–Chairs of Advisory Committee: Dr. Mi Lu
Dr. Zixiang Xiong

In this dissertation, we focus on two related parts of a 3D face recognition system with wireless transportation. In the first part, the core components of the system, namely, the feature extraction and classification component, are introduced. In the feature extraction component, range images are taken as inputs and processed in order to extract features. The classification component uses the extracted features as inputs and makes classification decisions based on trained classifiers. In the second part, we consider the wireless transportation problem of range images, which are captured by scattered sensor nodes from target objects and are forwarded to the core components (i.e., feature extraction and classification components) of the face recognition system. Contrary to the conventional definition of being a transducer, a sensor node can be a person, a vehicle, etc. The wireless transportation component not only brings flexibility to the system but also makes the "proactive" face recognition possible.

For the feature extraction component, we first introduce the 3D Morphable Model. Then a 3D feature extraction algorithm based on the 3D Morphable Model is presented. The algorithm is insensitive to facial expression. Experimental results show that it can accurately extract features. Following that, we discuss the generic face warping algorithm that can quickly extract features with high accuracy. The

proposed algorithm is robust to holes, facial expressions and hair. Furthermore, our experimental results show that the generated features can highly differentiate facial images.

For the classification component, a classifier based on Mahalanobis distance is introduced. Based on the classifier, recognition performances of the extracted features are given. The classification results demonstrate the advantage of the features from the generic face warping algorithm.

For the wireless transportation of the captured images, we consider the location-based wireless sensor networks (WSN). In order to achieve efficient routing performance, a set of distributed stateless routing protocols (PAGER) are proposed for wireless sensor networks. The loop-free and delivery-guaranty properties of the static version (PAGER-S) are proved. Then the performance of PAGER protocols are compared with other well-known routing schemes using network simulator 2 (NS2). Simulation results demonstrate the advantages of PAGER.

To my parents

ACKNOWLEDGMENTS

The first person I would like to thank is my advisor Prof. Mi Lu. She gave me the opportunity to come to the U.S. to start my Ph.D. study. Without her support, guidance, encouragement and trust, it would have been impossible for me to finish this dissertation. She is more a mentor than a mere advisor to me. She has set a high standard for me that really helps me grow. I also want to thank Prof. Zixiang Xiong for leading me into the face recognition project and for his guidance and support during the project period. I would like to thank Professors Scott Miller, Xi Zhang, and Andreas Klappenecker for spending their valuable time and effort in serving on my committee.

I feel a deep sense of gratitude to my parents for imbuing me with a love for books and for giving me the freedom to pursue my dream. I am really indebted to my sister and brother-in-law, who not only gave valuable advice on my Ph.D. study, but also took the responsibility of taking care my parents for me.

I would like to thank all the former labmates in the Computer Engineering group at Texas A&M University. I want to thank Dr. Zhili Zhao and Dr. Xiang Lu for reviewing my writings. I want to thank Dr. Yong Liu for discussing routing algorithms with me. Haiyun You and Chuan He helped me settle down when I first came to College Station and gave me significant help. I want to thank Dr. Shu Yan, Dr. Chin Ngai Sze and Dr. Zhuo Li for discussing their research works with me, which inspired me a lot. Cheng-Ta Chiang and Zhan Liu give me some valuable advice, so I also want to thank them.

I am indebted to all my groupmates in the multimedia lab at Texas A&M University. I want to thank Dr. Yong Sun and Dr. Jianping Hua for their help when I first started the face recognition project. I had the pleasure to work with Yang Yang,

Qian Xu and Zhixin Liu when I was in the multimedia lab. My life would have been awful without them.

I am grateful for Dr. Kenneth Castleman. Without his supervision and support of the face recognition project, it would have been impossible to finish this dissertation. I also want to thank Dr. Qiang Wu for his encouragement and help when I was in my difficult time. A special thanks goes to my mentor in ADIR, Dr. Samuel Cheng, for his contribution to the face recognition project. I really benefited from his insightful view and solid programming skills. I also want to thank my fellow interns in ADIR, Dr. Hyohoon Choi, Amit Paranjape and Sina Jahanbin for their help when I was doing my internship in the summers.

Additionally, I want to thank Dr. Dagang Wu, Dexing Dai, Thomas Tisserant, Yu Bai, Hao Li, Kun Men, Kai Jiang and Dr. Tie Liu for their friendship and companionship. There are numerous others without whom my Ph.D. years would have been both more difficult and less pleasant. I would like to thank all of them. Unfortunately, it is impossible to name them all; I sincerely apologize to those I have omitted.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

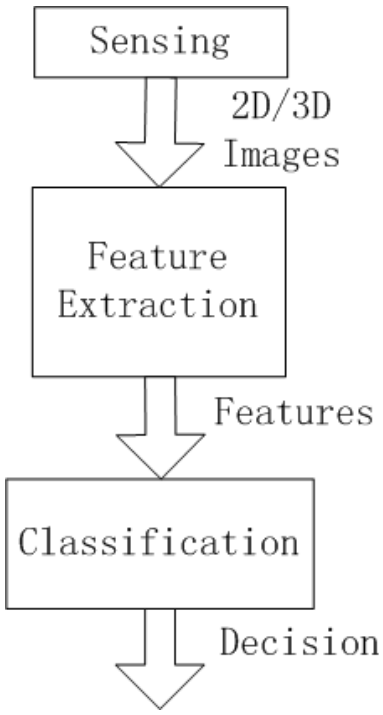


Fig. 1. Conventional 2D/3D face recognition system.

Like all pattern recognition systems, face recognition systems can also be divided into three components, as shown in Fig. 1. The first component is the sensing component that captures 2D/3D images containing human faces to be recognized. The

This dissertation follows the style of *IEEE Transactions on Information Forensics and Security*.

images are then passed to the feature extraction component, where their features are extracted. Using the extracted features as inputs, the classification component evaluates the features and makes final decisions regarding the classes to which the images belong.

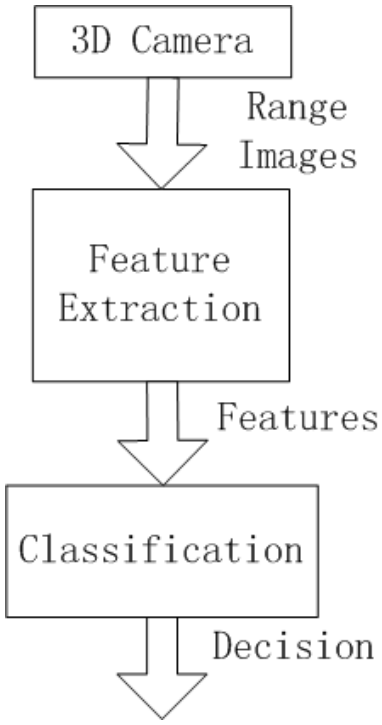


Fig. 2. Conventional 3D face recognition system.

Different to 2D face recognition systems that use 2D texture images as inputs, 3D face recognition systems utilize 3D range images to recognize a human face. Compared to 2D texture images, 3D range images contain 3D geometric information of human faces, and are insensitive to make-ups, lighting condition and pose variations. Hence it is commonly believed [9] that 3D face recognition systems have the advantage of achieving reliable recognition performance. In this dissertation, we restrict ourselves

to 3D face recognition systems.

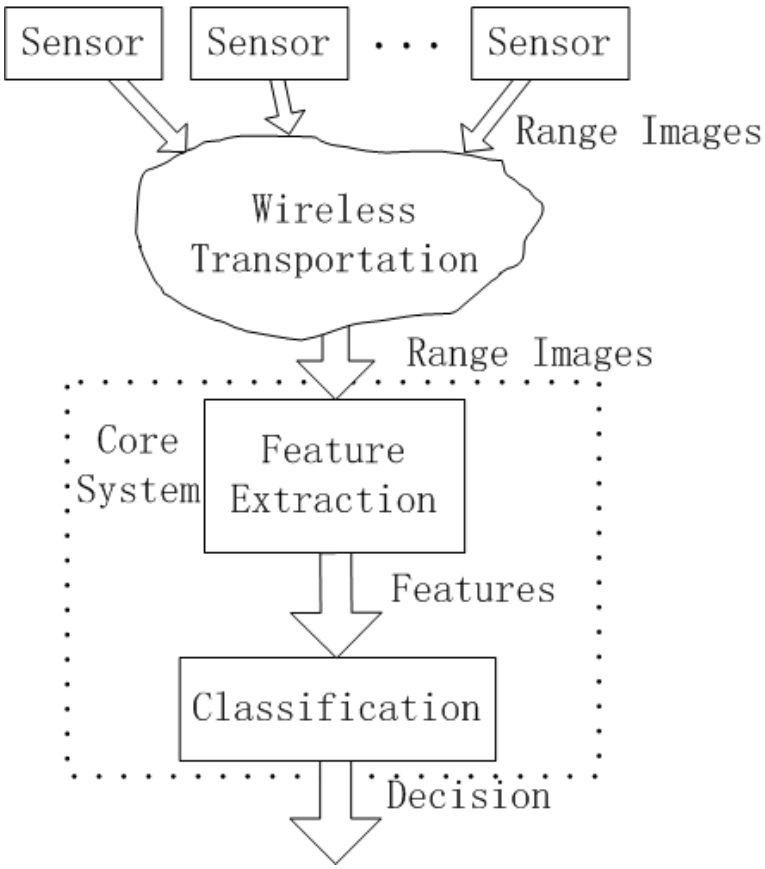


Fig. 3. The proposed 3D face recognition system with wireless transportation.

For a conventional 3D face recognition system, as shown in Fig. 2, range images captured by a local 3D camera system are forwarded directly to the feature extraction component. In our proposed 3D face recognition system shown in Fig. 3, unlike conventional systems, sensors capture the range images and forward the data to the feature extraction component via the wireless transportation component.

Compared to conventional 3D face recognition systems, our proposed framework

has several advantages. First, surveying coverage of the 3D face recognition system is more flexible. This is because the deployment of the sensors can be configured for a given specific task. Second, with mobile sensors, range images can be captured in a "proactive" way in our system. Rather than waiting for human subjects to enter a certain area to capture images, the sensors in our system has the mobility to capture images without being noticed. While the sensing component is beyond the scope of this dissertation, we only focus on the wireless transportation, feature extraction and classification components. The feature extraction and classification components can be considered together as the core part of our 3D face recognition system.

A. Feature Extraction and Classification of 3D Face Recognition

1. Feature Extraction Using 3D Morpahble Model

2D face recognition has been dogged by the problem of being sensitive to the viewpoint and lighting variations for a long time. These problem occurs when the lighting and viewpoint variations dramatically alter the appearance of a human face in 2D texture images. In order to obtain the actual texture and shape information of a human face in a texture image, the lighting and viewpoint parameters of the 2D image have to be estimated accurately. With the 3D Morphable Model [6], [7], this estimation becomes possible.

The 3D Morphable Model estimates the lighting and viewpoint parameters starting from several manually-defined landmark points, i.e., the eye corners, tip of the nose and the mouth corners. A cost function is defined to measure the position differences between the manually-defined landmark points in the input image and the corresponding landmark points in the 3D Morphable Model. Minimization of the cost function by optimizing the viewpoint and shape parameters of the 3D Morphable

Model not only gives a coarse estimation of viewpoint parameters of the face in the image, but also the initial shape parameters of the 3D Morphable Model. Based on the initial viewpoint and shape parameters, minimization of another cost function measuring the color differences between the input texture image and the 3D Morphable Model will give the refined estimation of viewpoint parameters along with the lighting condition parameters. In the meantime, while obtaining the lighting condition and viewpoint parameters of the face in the input texture image, the optimized texture and shape parameters of the 3D Morphable Model are also achieved to make the rendering of the model similar to the appearance of the face in the texture image. The shape and texture parameters are then become features of the human face in the input image. Obviously, the drawback of the extracting features from 2D texture images using 3D Morphable Model is that it depends on human assistance. In addition to estimation of viewpoint and lighting conditions, 3D Morphable Model also made a solid progress towards the hybrid face recognition, which utilizes both detailed and overall information and is considered as a promising direction.

Following this direction, we consider applying the 3D Morphable Model to extract features from the range images [72]. Unlike 2D texture images, 3D faces in texture images can be scaled, shifted and rotated to a standard pose. Therefore, the viewpoint parameters of the 3D model are fixed in this scenario. Additionally, lighting condition variations do not change geometric structures of the faces in range images. Therefore, it is only necessary to consider optimizing the shape parameters. We define a cost function measuring the depth differences between the 3D face and the input range image. Minimizing the cost function by optimizing faces results in optimized shape parameters of the 3D Morphable Model, which become the extracted features of the input range image. Compared to using 3D Morphable Model to extract features from 2D texture images, the advantage of this scheme is: 1) the extraction procedure is

automatic, 2) the feature extraction procedure is not affected by the lighting condition and viewpoint variations and 3) the extraction procedure is faster. Compared to other 2D facial feature extraction methods, the scheme also has some advantages: 1)it is not sensitive to holes, facial expressions and hair, and 2) as a hybrid approach, it extracts both detailed and overall information from range images.

2. Generic Face Warping

Initially designed for reconstructing 3D faces from texture images, although the 3D facial extraction scheme [72] based on 3D Morphabel Model has advantages, it suffers from its slow feature extraction and manual initialization. In order to speedup the extraction and perform the feature extraction automatically without losing the advantages, we propose the generic face warping scheme [73].

The 3D facial feature extraction scheme based on generic face warping [73] can automatically identify faces from range images. In our scheme, a set of carefully selected range images constitute a gallery of example faces, and another range image is chosen as a "generic face". The generic face is then warped to match (in the least mean square (LMS) sense) each of the example faces. Each such warp is specified by a vector of displacement values.

In operation, when a target face[1] comes in, the generic face is warped to match it. The geometric transformation used is a linear combination of the example face warping vectors. The coefficients of the various terms in the sum are adjusted to minimize the root mean square (RMS) error. This minimization is achieved by using two versions of Newton's method. One of them is the conventional Newton's method, where a fixed set of points are chosen from the downsampled generic face to measure

[1]A target image contains the human face that is going to be recognized.

the range difference between the warped generic face and the input face. The other version is a stochastic version. During each iteration, the comparing point set includes randomly selected points from the genetic face. The holes in an input face are dynamically avoided during the matching process by adaptive selection of matching points in the generic face.

After the matching process is complete, the coefficients of the composite warp are used as features. In order to choose representative example faces from the data set, the principle component analysis (PCA) technique is used to select range faces with large deviations from the average face. Furthermore, in order to match input faces with facial expressions, example faces with facial expressions are added to the gallery.

The accuracy of the feature extraction procedure is first demonstrated by the RMS errors between the warped generic faces and the input faces. Then by mapping the landmark points of the warped generic face to the input faces, we show the precision of the extracted features.

Compared to the 3D feature extraction process based on 3D Morphable Model, the feature extraction based on generic face warping is faster. This is due to two facts: 1) the 3D Morphable-Model-based feature extraction scheme has to calculate the occlusion and shadow in order to choose proper triangle during the feature extraction procedure while the generic-face-warping scheme does not have such requirement, and 2) the generic face warping scheme does not have the 3D geometric transformations required in the 3D Morphable Model.

3. Classification Based on Mahalanobis Distance

The features extracted from generic face warping procedure are forwarded to a classifier based on Mahalanobis distance. Before the classification, the features are prepro-

cessed by using the PCA technique in order to reduce the dimensionality and prevent numerical problems from happening during the calculation of Mhalanobis distances. Using the classifier, we consider building a verification system that differentiates a person from others. The classification process can be divided into the training stage and the testing stage. During the training stage, the cross validation is used to obtain the covariance matrix and the mean from the training sets of the person. In the testing stage, the trained classifier uses the testing set of the person to generate a series of false alarm rate and hit rate based on different Mahalanobis thresholds. These false alarm rate and hit rate are rendered as the receiver operating characteristic (ROC) curve. To obtain the accurate overall performance of the verification system, ROC curves of verifying different persons are averaged to obtain the average ROC curves, which show the superiority the verification system.

B. Distributed Routing for Wireless Sensor Networks

We then consider transferring the range images from sensors to a base station following a multi-hop fashion. Furthermore, we restrict ourselves to the network layer and consider only routing issue of the located-based sensor network. Since the sensors are energy-restricted, in order to prolong the battery life of sensors, energy efficient routing protocols are highly desired. Furthermore, other routing metrics, including delivery-ratio, path length, control overhead are also important issues for use to consider.

In order to achieve the desired properties, the dead-end problem is an inevitable issue. The dead-end problem arises when the packets in a location-based network are forwarded following the greedy forwarding, which is the foundation of all location-based routing schemes. Without handling the dead-end problem, packets being for-

warded to the base station may experience long path, enormous delays and severe losses.

The dead-end problem occurs when a message falls into a local minimum using greedy forwarding. Current solutions to this problem are insufficient in either eliminating traffic/path memorization or finding satisfactory short paths. In this paper, we propose a novel algorithm, called Partial-partition Avoiding GEographic Routing (PAGER) [74]–[76], to solve the problem. The basic idea of PAGER is to divide a sensor network graph into functional sub-graphs and provide each sensor node with message forwarding directions based on these sub-graphs. PAGER results in loop-free short paths without memorization of traffics/paths in sensor nodes. It does not require planarization of the underlying network graph. Further, the mobility adaptability of PAGER makes it suitable for use in mobile sensor networks with frequent topology changes. We implement the PAGER algorithm in two protocols and evaluate them in sensor networks with different parameters. Experimental results show the advantage of PAGER in the context of sensor networks.

C. Applications

There are many potential applications of the proposed 3D face recognition system with wireless transportation. One scenario is the mobile criminal identification system [71], a unit of police officers is deployed in a certain surveying area. Each officer carries a mobile 3D camera to capture 3D faces of human objects that are believed to have potential danger. The range images that contain the 3D faces of suspects are transferred to a central database at a base station via a wireless network. In order to maximize the battery life of the mobile devices carried by the police officers, the range images are forwarded to the base station following a multiple-hop fashion. The

wireless network can be considered as the wireless transportation component. At the base station, range images of the suspects will be analyzed by the automatic feature extraction component in order to produce features that can effectively differentiate a person from others. The extracted features are then processed by the classification component to determine whether the range image contains a face belongs to a criminal in the database or not. Unlike conventional identification system, in this mobile criminal identification system, range images are captured in a "proactive" way. Rather than having a certain number of fixed check points, the system has a large number of check points with great mobility, which largely increases the flexibility and enlarges the surveying area. Furthermore, the centralized database at the base station contains records of thousands of criminals, which are impossible for any police officer to memorize. As a result, the search ability of police officers will be greatly enhanced.

D. Brief History of 3D Face Recognition and Routing in Wireless Sensor Network

The earliest 3D face recognition systems can be traced back to more than one decade ago, when Cartoux *et al.* [12] presented a 3D face recognition system based on classifying profiles extracted from range face images. Following that, Lee and Milios [42] proposed a scheme that calculated Gaussian curvatures from range face images and created extended gaussian images (EGI) for classification. Trough the 1990s, however, relatively little work [28], [52] appeared in the literature. In recent years, due to 3D face recognition's practical relevance to homeland security, it has received broad attention in both academia and industry. Similar to 2D face recognition, 3D face recognition can also be divided into three categories, namely, holistic, feature-based and hybrid approaches. Holistic approach considers a 3D face image as a whole

without distinguishing different parts of face, and is a popular category constitutes many schemes [1], [14]–[17], [28], [32], [47]. Compared to 2D face recognition, fewer schemes [18], [42], [61] in the literature belong to the feature-based approach. This may due to the fact that it is more difficult to find out features (*e.g.*, iris, eye corners, mouth corners) in a range face than in a texture face. The schemes [65], [72], [73] belonging to the hybrid category extract both detailed and overall information from a range image. In the 2D face recognition, hybrid approach is considered as a promising direction. And because of the similarity between 2D and 3D face recognition, it is reasonable to believe that hybrid approach is also a research direction with great potential in 3D face recognition.

The earliest work on the dead-end problem of location-based wireless networks appeared in the literature in late 1980s. In his 1987 paper [27], Finn reported the dead-end problem in the large metropolitan-scale internetworks. However, not until the early 2000s did researchers turn their attention to this problem. The main reason behind the research direction turning is because of the emergence of the large-scale wireless ad-hoc networks and, further, the wireless sensor networks. In large-scale wireless ad-hoc networks and wireless sensor networks, location-based routing methods have inherent scalability. In other words, increase of the size of wireless networks will not result in the increase of control overhead amount of routing methods. Without solving the dead-end problem, however, location-based routing methods cannot guaranty packet delivery. Therefore, handling the dead-end problem becomes necessary. In order to handle the problem, both memorization-based [37], [60] and stateless [8], [39] recover mode algorithms have been proposed to address the dead-end problem. The memorization-based methods [37], [60] require nodes to memorize their past traffic or paths. These methods are not scalable because they are sensitive to the node queue size, changes in node activity and node mobility. In contrast, stateless

recover mode methods [8], [39] do not require a node to memorize its past traffic or paths, hence they maintain almost no state information in sensor nodes; this property makes the stateless recover mode methods [8], [39] suitable for sensor networks, which usually consist of a large number of sensor nodes.

E. Organization of the Dissertation

We focus on two related parts of a networked 3D face recognition system, namely, the core face recognition part and its wireless transportation component. This dissertation is organized as follows. In Chapter II, we review face 3D Morphable Model, and present a novel 3D facial feature extraction scheme based on the 3D Morphable Model. Chapter III is focused on a novel face recognition scheme. We describe the algorithm of the feature extraction based on generic face warping. We depict classification performance of the features generated from the scheme and compare to other features to demonstrate the its advantage. Chapter IV describes a wireless sensor network model that is used to be the wireless transportation component of the 3D face recognition system. A distributed stateless routing algorithm is provided for stationary scenarios. The loop-free and delivery-guaranty properties are proved. A distributed stateless routing algorithm for scenarios with frequent topology changes is presented. The simulation results show the advantages of these routing algorithms.

F. Contributions of the Dissertation

The main contributions of the dissertation are the following:

- A novel facial feature extraction scheme based on 3D Morphable Model to extract not only overall shape information but also details from a target face.
- A novel 3D facial extraction scheme based on the generic face warping, which

is robust to facial expressions, holes, hair and has a high recognition rate.

- A novel distributed stateless routing scheme designed for static wireless sensor networks, which is proved to be loop-free, delivery-guaranty and outperforms other routing schemes in the report literature.

- A novel distributed stateless routing specifically designed for mobile wireless sensor networks, which outperforms other routing schemes in the reported literature.

- A novel framework of face recognition system with wireless transportation component largely increases the flexibility and enlarges the surveying area, as well as the search ability.

CHAPTER II

FEATURE EXTRACTION FROM RANGE IMAGES BASED ON 3D MORPHABLE MODEL

In this chapter, a 3D feature extraction scheme [72] is introduced for 3D face recognition. Unlike previous methods that fit a 3D Morphable Model to 2D intensity images, our scheme utilizes 3D range images to extract features without requiring manually-defined initial landmark points. A linear transformation is used to achieve the mapping between the 3D model and a 3D range image, which makes the computation simple and fast. Moreover, our scheme is robust to the illumination and pose variations. In addition to features from range images, extra features can be obtained by examining optional 2D texture images. Using our scheme, we can also perform automatic eye/mouth corner localization. Experimental results show the high accuracy and robustness of our scheme.

A. Introduction

Due to tighten homeland security in the US, face recognition has attracted increasing interests recently. In a face recognition system, facial feature extraction is one of the most significant components. The goal is to automatically extract features from face images with accuracy and robustness. Currently there are two approaches: one relies on 2D texture images; another utilizes 3D range images.

The first approach is plagued by problems due to viewpoint and lighting variations, which make it difficult to extract facial features accurately without human assistance. For example, although methods based on the 3D Morphable Model [6],

[7] can handle illumination and viewpoint variations, they rely on manually-defined landmark points to fit the 3D model to 2D intensity images.

The second approach utilizes depth information of 3D range images to extract features. Since 3D range images are invariant to illumination changes, the impact of lighting variations is moot in this approach. Furthermore, 3D faces in range images can be rotated and shifted to a standard pose to overcome the problem caused by viewpoint variations. Therefore, this approach becomes increasingly attractive nowadays. Motivated by early research works [28], [52] that began about a decade ago, different feature extraction techniques [11], [18], [43], [47], [61] have been proposed by using wavelet-signature [18], curvature [61] and rigid surface matching [11], [43], [47] techniques. These techniques, however, have the problem of being sensitive to facial expression variations. While this problem was not addressed in [18], [43], [61], a partial solution of the problem was addressed in [11], [47] by using rigid surface matching.

In this chapter we propose a method based on 3D Morphable Model to extract facial features from range images. In our scheme, a range image from a 3D image acquisition system is normalized before being used as an input. Then a synthesized 3D face is generated by minimizing the range difference between the input range image and the 3D Morphable Model. After this fitting procedure, features can be obtained from the shape coefficients of the newly synthesized 3D face. In addition, an extra optimization step can be performed to extract extra features from an optional texture image, which is generated from the 3D image acquisition system together with the input range image.

Compared to the 3D Morphable Model methods in [6], [7], which use 2D texture images as inputs, our method has several advantages. First, initialization based on manually-defined landmark points is not needed. In order to fit the 3D Morphable

Model to a 2D texture face image, some key information (e.g., locations of eye and mouth corners) of the human face has to be given as initialization. Because a texture image contains a mixture of shape and color information of a human face, it is difficult to extract the shape information accurately from the texture image without human assistance. In [6], [7], this important information is extracted manually. In our method, since input range images contain only the shape information, this key information can be obtained automatically during the fitting procedure. Therefore, manual initialization is not needed in our method. Second, lighting variations, which typically lower the accuracy of feature extraction, do not present a problem in our method, because range images are robust to illumination changes. Third, a linear transformation is used to map the 3D model to an input 3D range image, which makes the computation simple and fast.

Compared to other range image-based feature extraction techniques [11], [18], [43], [47], [61], our method can overcome the difficulty caused by expression variations. This is due to the fact that the 3D Morphable Model [6], [7] contains example faces with facial expressions. This allows us to closely synthesize faces with facial expressions and extract features accurately. In addition, our method can be used to automatically label eye/mouth corners. Experimental results show the effectiveness of our method.

B. Acquisition and Normalization of Input Images

In our system, the range images, along with the corresponding texture images, are obtained by using a 3D image acquisition system from 3Q Inc., which can generate high resolution 3D surface images in less than 2 milliseconds.

The obtained 3D meshes from the 3D image acquisition system describe 3D faces

with different orientations. These 3D "raw" faces cannot be directly used as inputs to our fitting algorithm. In order to make the 3D faces have the same orientation, we rotate and shift each of them until the mean-squared-difference (MSD) between it and a standard face is minimized. After the normalization, the eyes look straight ahead and lie on a line parallel to the x-axis.

C. 3D Morphable Model

The 3D morphable face model [6], [7] is constructed based on a vector space representation of 3D faces. In this model, a synthesized 3D face can be represented by a convex combination of n shape and texture vectors $\mathbf{S}_i$ and $\mathbf{T}_i$ $(i = 1, \ldots, n)$ of real human faces (example faces). Let $\overline{\mathbf{s}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{S}_i$ and $\overline{\mathbf{t}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{T}_i$, then we can obtain covariance matrices $A_S = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{S}_i - \overline{\mathbf{s}})(\mathbf{S}_i - \overline{\mathbf{s}})^T$ and $A_T = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{T}_i - \overline{\mathbf{t}})(\mathbf{T}_i - \overline{\mathbf{t}})^T$. We further calculate the ith eigenvalue $\sigma_{R,i}^2$ and its corresponding eigenvector $\mathbf{s}_i$ of A_R. Similarly, let $\sigma_{T,i}^2$ and $\mathbf{t}_i$ be the ith eigenvalue and the corresponding eigenvector of A_T. The shape and color of a synthesized face can be represented by a shape vector $\mathbf{s}$ and a texture vector $\mathbf{t}$, respectively, with

$$\mathbf{s} = \overline{\mathbf{s}} + \sum_{i=1}^{n}\alpha_i\mathbf{s}_i, \;\; \mathbf{t} = \overline{\mathbf{t}} + \sum_{i=1}^{n}\beta_i\mathbf{t}_i, \tag{2.1}$$

where the distributions of the coefficients $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)^T$ are:

$$p(\boldsymbol{\alpha}) \propto \exp(-\frac{1}{2}\sum_{i=1}^{n}\frac{\alpha_i^2}{\sigma_{R,i}^2}), \;\; p(\boldsymbol{\beta}) \propto \exp(-\frac{1}{2}\sum_{i=1}^{n}\frac{\beta_i^2}{\sigma_{T,i}^2}). \tag{2.2}$$

Given an index i and coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, a point p of the 3D model can be defined

by its 3D position $s(i, \boldsymbol{\alpha}) = (x(i, \boldsymbol{\alpha}), y(i, \boldsymbol{\alpha}), z(i, \boldsymbol{\alpha}))$ and its color $t(i, \boldsymbol{\beta}) = (r(i, \boldsymbol{\beta}), g(i, \boldsymbol{\beta}), b(i, \boldsymbol{\beta}))$.

The 3D model is placed in the 3D space with the eyes on a line parallel to the x-axis and looking straight ahead. So the 3D Morphable Model and the 3D faces in the normalized input range images have the same head pose. As we will see later, this property simplifies the computation of our algorithm.

D. Feature Extraction from Range Images

The key step of our feature extraction method is to fit the 3D Morphable Model to an input 3D range image. After the fitting procedure, the shape coefficients of the model are optimized and become range features of the human face in the image.

1. Automatic Initialization

In the work presented in [6], [7], several manually-defined landmark points are used to initialize the feature extraction procedure. In our method, as we explained in the introduction of the chapter, automatic initialization is achieved by using range images that contain only the shape information.

2. 3D Transformation

In the 3D-2D transformation presented in [7], a perspective projection is used to map the 3D Morphable Model to a 2D image, which makes the transformation nonlinear. In our scheme, the mapping between the 3D Morphable Model and a 3D range image is a linear transformation because the perspective projection is not needed. Since the 3D model and the normalized input 3D faces have the same head pose, this linear transformation can be further simplified to scaling and shifting. Let the original

coordinates of a point p be $(x(i,\boldsymbol{\alpha}), y(i,\boldsymbol{\alpha}), z(i,\boldsymbol{\alpha}))$, which are defined by the index i and the shape coefficients $\boldsymbol{\alpha}$[1]. We use the following transformation to transfer the origin coordinates to the new coordinates $(x'(i,\boldsymbol{\alpha}), y'(i,\boldsymbol{\alpha}), z'(i,\boldsymbol{\alpha}))$:

$$\begin{cases} x'(i,\boldsymbol{\alpha}) = scale_{x,y} \cdot x(i,\boldsymbol{\alpha}) - off_x \\ y'(i,\boldsymbol{\alpha}) = scale_{x,y} \cdot y(i,\boldsymbol{\alpha}) - off_y \\ z'(i,\boldsymbol{\alpha}) = scale_z \cdot z(i,\boldsymbol{\alpha}) - off_z. \end{cases} \tag{2.3}$$

In the above transformation, the values of the scale factors $scale_{x,y}$, $scale_z$ and the offsets off_x, off_y and off_z depend on the input range image format. Note that $scale_{x,y}$ and $scale_z$ may not have the same value.

3. Cost Function

Let R represent the input range image and m be the length of the corresponding shape vector $\mathbf{s}$. When an index i $(i = 1, \ldots, m)$ of $\mathbf{s}$ and the shape coefficients $\boldsymbol{\alpha}$ are given, the described linear transformation maps the original coordinates $s(i,\boldsymbol{\alpha})$ to the new coordinates $(x'(i,\boldsymbol{\alpha}), y'(i,\boldsymbol{\alpha}),$
$z'(i,\boldsymbol{\alpha}))$. We define the cost function as the sum of the squared difference between the transformed depth $z'(i,\boldsymbol{\alpha})$ and the corresponding range image depth $R(x'(i,\boldsymbol{\alpha}),$ $y'(i,\boldsymbol{\alpha}))$, which can be represented as

$$C(\boldsymbol{\alpha}) = \sum_{i=1}^{m} (z'(i,\boldsymbol{\alpha}) - R(x'(i,\boldsymbol{\alpha}), y'(i,\boldsymbol{\alpha})))^2. \tag{2.4}$$

Note that this cost function is based on all the indices of the shape vector $\mathbf{s}$. In other words, we use $C(\boldsymbol{\alpha})$ to measure the entire range difference between the 3D model

[1]We do not have texture coefficients $\boldsymbol{\beta}$ when only range images are involved.

and the 3D face in R. This cost function can then be minimized by using standard optimization methods (e.g., Newton's method).

4. Maximum A Posteriori Estimator

Minimization of cost function $C(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ may produce unrealistic 3D faces. So we use a maximum a posteriori (MAP) estimator to modify $C(\boldsymbol{\alpha})$. Given an input image R, we try to maximize the posterior probability $p(\boldsymbol{\alpha}|R)$ with respect to the shape coefficients $\boldsymbol{\alpha}$. According to the Bayes rule,

$$p(\boldsymbol{\alpha}|R) \propto p(R|\boldsymbol{\alpha})p(\boldsymbol{\alpha}). \tag{2.5}$$

Given the shape coefficients $\boldsymbol{\alpha}$ of the model, the distribution of an input range image R is assumed to follow a normal distribution, i.e., $p(R|\boldsymbol{\alpha}) \propto \exp(-\frac{1}{2\sigma_R^2}C(\boldsymbol{\alpha}))$. The posterior probability is then maximized by minimizing

$$C'(\boldsymbol{\alpha}) = -2\log p(\boldsymbol{\alpha}|R) = \frac{1}{\sigma_R^2}C(\boldsymbol{\alpha}) + \sum_{i=1}^{n}\frac{\alpha_i^2}{\sigma_{R,i}^2}, \tag{2.6}$$

where σ_R^2 is a controllable parameter representing the relative weight of $C(\boldsymbol{\alpha})$ in $C'(\boldsymbol{\alpha})$. Thus after the modification of the MAP estimator, $C'(\boldsymbol{\alpha})$ is the cost function that we try to minimize by optimizing the shape coefficients $\boldsymbol{\alpha}$.

5. Optimization Procedure

The cost function $C(\boldsymbol{\alpha})$ is based on all the indices of the shape vector $\mathbf{s}$. In order to accelerate the minimization of $C(\boldsymbol{\alpha})$ without falling into local minima, we employ a stochastic version of Newton's method. In each iteration, eighty indices of $\mathbf{s}$ are randomly selected to compose a set $\mathcal{K}$. Based on the set $\mathcal{K}$, a new cost function $C_{\mathcal{K}}$

is defined:

$$C_{\mathcal{K}}(\boldsymbol{\alpha}) = \sum_{i \in \mathcal{K}} (z'(i, \boldsymbol{\alpha}) - R(x'(i, \boldsymbol{\alpha}), y'(i, \boldsymbol{\alpha})))^2. \tag{2.7}$$

The corresponding cost function after the modification of the MAP estimator becomes:

$$C'_{\mathcal{K}}(\boldsymbol{\alpha}) = \frac{1}{\sigma_R^2} C_{\mathcal{K}}(\boldsymbol{\alpha}) + \sum_{i=1}^{n} \frac{\alpha_i^2}{\sigma_{R,i}^2}. \tag{2.8}$$

In each iteration, the first and second derivatives of $C_{\mathcal{K}}(\boldsymbol{\alpha})$ with respect to the jth shape coefficient α_j $(j = 1, \ldots, n)$ can be calculated as

$$\begin{aligned} \frac{\partial C_{\mathcal{K}}(\boldsymbol{\alpha})}{\partial \alpha_j} \quad &= \sum_{i \in \mathcal{K}} 2(z'(i, \boldsymbol{\alpha}) - R(x'(i, \boldsymbol{\alpha}), y'(i, \boldsymbol{\alpha}))) \cdot \\ &\Big(\frac{\partial z'(i, \boldsymbol{\alpha})}{\partial \alpha_j} - \frac{\partial R(x'(i, \boldsymbol{\alpha}), y'(i, \boldsymbol{\alpha}))}{\partial x'} \cdot \frac{\partial x'(i, \boldsymbol{\alpha})}{\partial \alpha_j} \\ &- \frac{\partial R(x'(i, \boldsymbol{\alpha}), y'(i, \boldsymbol{\alpha}))}{\partial y'} \cdot \frac{\partial y'(i, \boldsymbol{\alpha})}{\partial \alpha_j}\Big), \end{aligned} \tag{2.9}$$

$$\begin{aligned} \frac{\partial^2 C_{\mathcal{K}}(\boldsymbol{\alpha})}{\partial \alpha_j^2} \quad &= \sum_{i \in \mathcal{K}} 2\Big(\frac{\partial z'(i, \boldsymbol{\alpha})}{\partial \alpha_j} \\ &- \frac{\partial R(x'(i, \boldsymbol{\alpha}), y'(i, \boldsymbol{\alpha}))}{\partial x'} \cdot \frac{\partial x'(i, \boldsymbol{\alpha})}{\partial \alpha_j} \\ &- \frac{\partial R(x'(i, \boldsymbol{\alpha}), y'(i, \boldsymbol{\alpha}))}{\partial y'} \cdot \frac{\partial y'(i, \boldsymbol{\alpha})}{\partial \alpha_j}\Big)^2 \\ &+ \sum_{i \in \mathcal{K}} 2(z'(i, \boldsymbol{\alpha}) - R(x'(i, \boldsymbol{\alpha}), y'(i, \boldsymbol{\alpha}))) \\ &\cdot \Big(\frac{\partial^2 z'(i, \boldsymbol{\alpha})}{\partial \alpha_j^2} - \frac{\partial^2 R(x'(i, \boldsymbol{\alpha}), y'(i, \boldsymbol{\alpha}))}{\partial \alpha_j^2}\Big). \end{aligned} \tag{2.10}$$

With these derivatives, we can further obtain the derivatives $\frac{\partial C'_{\mathcal{K}}(\boldsymbol{\alpha})}{\partial \alpha_j}$ and $\frac{\partial^2 C'_{\mathcal{K}}(\boldsymbol{\alpha})}{\partial \alpha_j^2}$

$(j = 1, \ldots, n)$. The shape coefficients $\boldsymbol{\alpha}$ are updated according to $\boldsymbol{\alpha} = \boldsymbol{\alpha} - \lambda H^{-1} \nabla C'_{\mathcal{K}}$, where $H^{-1} \approx diag(1/\frac{\partial^2 C'_{\mathcal{K}}(\boldsymbol{\alpha})}{\partial \alpha_j^2})$ $(j = 1, \ldots, n)$ is the inverse Hessian matrix and $\lambda \ll 1$ the learning rate.

6. Segmentation

In some subregions (e.g., eyes and mouth) of a 3D face, the range differences are subtle. Thus if we use the whole 3D morphable model to minimize the cost function $C(\boldsymbol{\alpha})$, the locations of these subregions on the resulting synthesized 3D face may not match those on the input range image. We therefore segment the 3D model into three separate subregions (e.g., eyes, nose and mouth in Fig. 4). After fitting the whole 3D model to a range image, we independently minimize $C(\boldsymbol{\alpha})$ based on these subregions. This segmentation step improves the detailed description of the subregions on the synthesized 3D face.

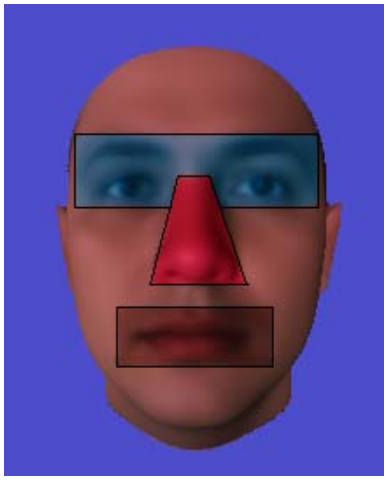


Fig. 4. Three segmented subregions on the 3D morphable model.

E. Feature Extraction from Texture Images

So far our feature extraction method only utilizes 3D range images for feature extraction. Extra features can be obtained by minimizing the color difference between the 3D model and 2D texture images. The details of the procedure are omitted here because of space limitations. Thus our feature extraction method can be considered as a 3D+2D approach that utilizes both 3D range and 2D texture images for feature extraction.

F. Experimental Results

1. Test Data Set

Fig. 5. An example of range (left) and texture (right) images.

The test data set contains both range and texture images. Each human face is represented by a pair of range and texture images as shown in Fig. 5. A range/texture image is 500×750 and each pixel has 8 bit resolution. These images are normalized before being used as inputs of our fitting algorithm. The normalized range and texture images have the following properties: 1) the tip of the nose on a range image is at the center of the image; 2) the depth of the tip of the nose has the gray level of 255; 3) zero gray level represents a plane 82 mm behind the tip of the nose.

2. Synthesis of Range and Texture Images

(a) (b) (c)

Fig. 6. Fitting results. (a) Original range image. (b) Original texture image. (c) Synthesized image.

We test our algorithm by reconstructing 3D faces from input images. For display purposes, an extra fitting procedure is used to recover the color information of an

input face. We show a fitting example in Fig. 6. The shape coefficients are optimized by fitting the 3D morphable model to the range image in Fig. 6 (a). After that, we fix the shape coefficients and optimize the color coefficients by fitting the 3D model to the texture image in Fig. 6 (b). Finally, we show the frontal view of the synthesized 3D face in Fig. 6 (c).

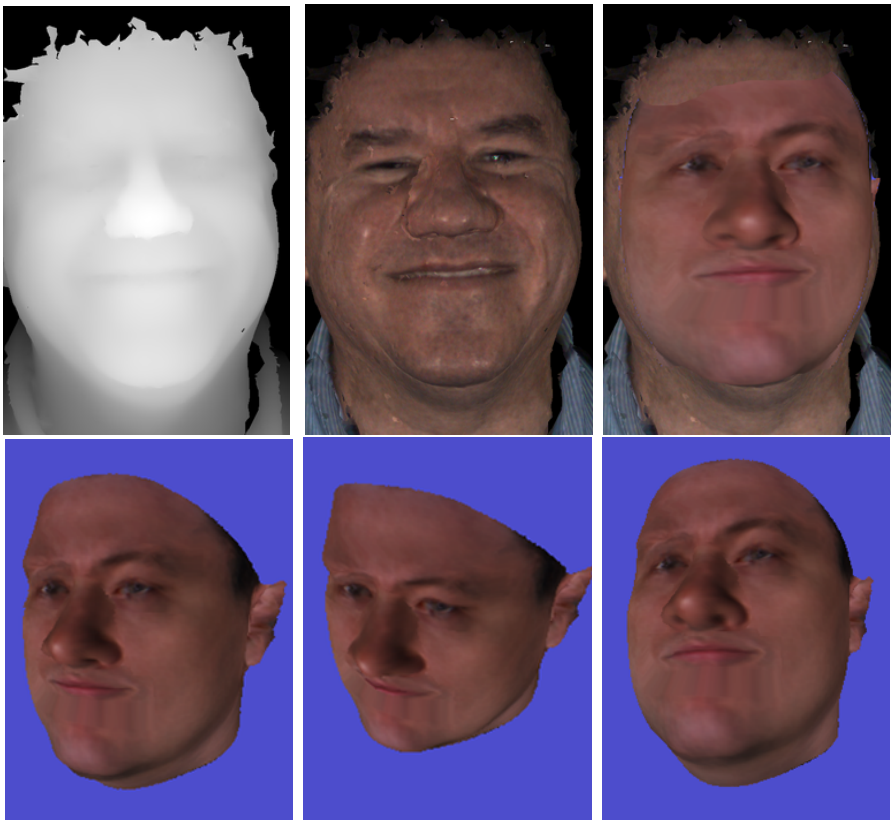


Fig. 7. Fitting results on a smiling face without perfect normalization. Top left: input range image; Top middle: input texture image; Top right: frontal view of the synthesized 3D face; Bottom row: different views of the synthesized 3D face.

Some of the faces in the data set are not well normalized and some have facial expressions, beards and mustaches. Fig. 7 shows a fitting example on one of such

faces. An extra fitting based on texture image is performed for display purposes. In this example, the face in the range and texture images has a cheerful facial expression and the right eye is slightly higher than the left eye. After the automatic face fitting procedure, the frontal view (Fig. 7 [top right]) and other views (Fig. 7 [bottom row]) of the synthesized 3D face are shown to demonstrate the effectiveness of our method.

3. Feature Localization

Fig. 8. Localization results.

To further test the performance of our algorithm, we run our proposed algorithm

on 100 range images to label the positions of the tip of the nose, eye corners and mouth corners. Some of the results are shown in Fig. 8. Again for display purposes, we show the labeling results on texture images, although our tests are performed on range images only. It is seen that our proposed algorithm produces accurate localization results.

G. Discussion

In this chapter we proposed a method to automatically extract facial features from range images using 3D Morphable Model. Like other feature extraction methods, the illumination problem is eliminated by using range images. Comparing with other feature extraction methods on range images, the proposed method is robust to the effect of facial expression, beard and mustache. The inherent linear transformation of this method makes it faster than the fitting algorithm [6], [7] of using 3D Morphable Model on texture images. Further, the manual initialization of Vetter's algorithm is no longer required in our method. A byproduct of this method is the automatical eye corner and mouth corner locator. Our experiments show the accuracy of the feature extraction and feature points locating. Despite the advantages, the proposed feature extraction scheme suffers from slow extraction speed. On the current stage, it takes more than 2 minutes to finish the feature extraction process. In the next chapter, we will propose a 3D face recognition scheme that utilizes a much faster feature extraction technique without losing the advantages of the feature extraction based on the 3D morphable model.

CHAPTER III

FACE RECOGNITION BASED ON WARPED EXAMPLE FACES*

In this chapter we describe a novel 3D face recognition scheme [73] for 3D face recognition scheme that can automatically extract features from range images, and is insensitive to holes, facial expression and hair. In our scheme, a number of carefully selected range images constitutes a set of example faces, and another range image is chosen as a "generic face". The generic face is then warped to match each of the example faces in the least mean square sense. Each such warp is specified by a vector of displacement values. In feature extraction operation, when a target face image comes in, the generic face is warped to match it. The geometric transformation used in the warping is a linear combination of the example face warping vectors. The coefficients in the linear combination are adjusted to minimize the root mean square error. After the matching process is complete, the coefficients of the composite warp are used as features for face recognition. Our technique is tested on a data set containing more than 600 range images. Experimental results in the access control scenario show the effectiveness of the extracted features.

A. Introduction

Face recognition has received broad attention in both academia and industry due to its practical relevance to homeland security. Facial appearance can uniquely identify a person, and it is a primary factor that people use to recognize each other. Although

*© [2007] IEEE. Reprinted, with permission, from "3D face recognition based on warped example faces" by L. Zou, S. Cheng, Z. Xiong, M. Lu, and K. Castleman, 2007. *IEEE Transactions on Information Forensics and Security*, to appear.

human face images can be affected by illumination, facial expression, makeup, etc., they are non-invasive in nature and easy to collect in environments where other biometrics (e.g., fingerprint, iris) require cooperation of the subjects. Therefore, face recognition is both an attractive and a challenging area for research. During the past decade, many different 2D and 3D face recognition algorithms have been presented [68].

Differentiated by the input image type, these algorithms can generally be categorized into 2D face recognition and 3D face recognition. According to the survey by Zhao *et al.* [68], 2D face recognition algorithms can be classified into feature-based, holistic and hybrid approaches. In the holistic category, algorithms use principle component analysis (PCA) [63], linear discrimination analysis (LDA) [4], independent component analysis (ICA) [3], support vector machine (SVM) [55] and neural networks [45] to analyze a face as a whole unit, without distinguishing the individual details (nose, mouth, eyes, etc.). By contrast, algorithms that extract detailed information from these regions are called feature-based approaches [22], [56], [64]. Hybrid approaches [6], [7], [34], [41], [67] consider both the overall and detailed information in a face and are regarded as the most promising method [68]. Since 2D texture pictures can be easily affected by external conditions (viewpoint, lighting, makeup, etc.), 2D face recognition algorithms have difficulty to classify human faces with high accuracy. Although face recognition methods based on 3D morphable model [6], [7], [67] have made breakthrough progress in successfully solving the problem caused by viewpoint and lighting variation, they rely on manually defined landmark points to initialize the algorithm [6], [7], [67].

3D face recognition algorithms, on the other hand, utilize the geometric structure of the face to recognize a person. Since 3D facial models are invariant to illumination, the impact of lighting variations is moot in this approach. Furthermore, 3D faces in

range images can be rotated and shifted to a standard pose to overcome the problem caused by viewpoint variations. Therefore, this approach becomes increasingly attractive nowadays. Motivated by early research works [5], [28], [52] that began about a decade ago, different 3D face recognition systems [1], [11], [14]–[19], [32], [43], [46]–[49], [51], [57], [61] have been proposed. In these schemes, principle PCA [1], [32], wavelet-signature [18], curvature [61], rigid surface matching [14], [16], [43], iterative closest point (ICP) [19], [47]–[49], [51] and optimal component analysis [46], [57] techniques are employed in order to achieve good recognition performance.

An important issue on face recognition is the problem of facial expression, which degrades the performance of most current 3D face recognition systems [9]. While most face recognition systems [18], [19], [43], [47], [49], [61] do not address the problem explicitly, several schemes have been proposed to handle it [11], [14], [16], [48], [51]. In the *multiple nose region* matching method [14], [16] presented by Chang *et al.*, different patches around the nose area in a target face are utilized to compare with those surfaces in the gallery. In their algorithm, these patches are segmented based on the surface curvatures. A target face is identified by measuring the similarity between these patches of the target face and those in the gallery. By matching the regions around the nose, which is believed to have relatively low shape variation caused by expressions, their scheme successfully handled the facial expression problem. In the *region matching* method [51] proposed by Mian *et al.*, different regions of faces in a gallery are segmented and then matched to a target image using ICP. The matching scores are then used to identify the target face. Facial expression problems in target faces are avoided by choosing matching regions that are unlikely to be affected by expression. In the *deformable model fitting* approach proposed by X.Lu *et al.* [48], a deformable model is presented to model the facial expressions and achieves good recognition rate. On the other hand, Bronstein *et al.* tried to use

a modeling technique to solve the problem [11]. In their proposed algorithm [11], generalized multidimensional scaling (GMDS) is used to model facial expressions in order to achieve the expression-invariant performance.

Two other important issues of 3D face recognition are facial hair and holes on the range images, which have negative impact on all 3D face recognition systems [9]. Facial hair in a target face changes the shape of a range image and introduces noises to the face. A hole is an area of missing data in a range image caused by sensor's failure of acquiring data from objects. The *region matching* method [51] explicitly addressed the problem and partially handled the situation by matching the nose and the forehead region only. Their approach, however, does not consider the situation when the forehead region is also partially covered by hair and holes. Although the approach [14], [16] proposed by Chang *et al.* did not state the problem explicitly, their approach also employs the same strategy of the *region matching* method [51] and can partially solve the problem.

In this chapter we propose a scheme using the combination of multiple face warps derived from 3D faces (called example faces) to extract facial features from range images. These example faces then comprise a set called "example set". In our scheme, an input 3D face image is imitated by a warped generic face using a linear combination of the face warps.

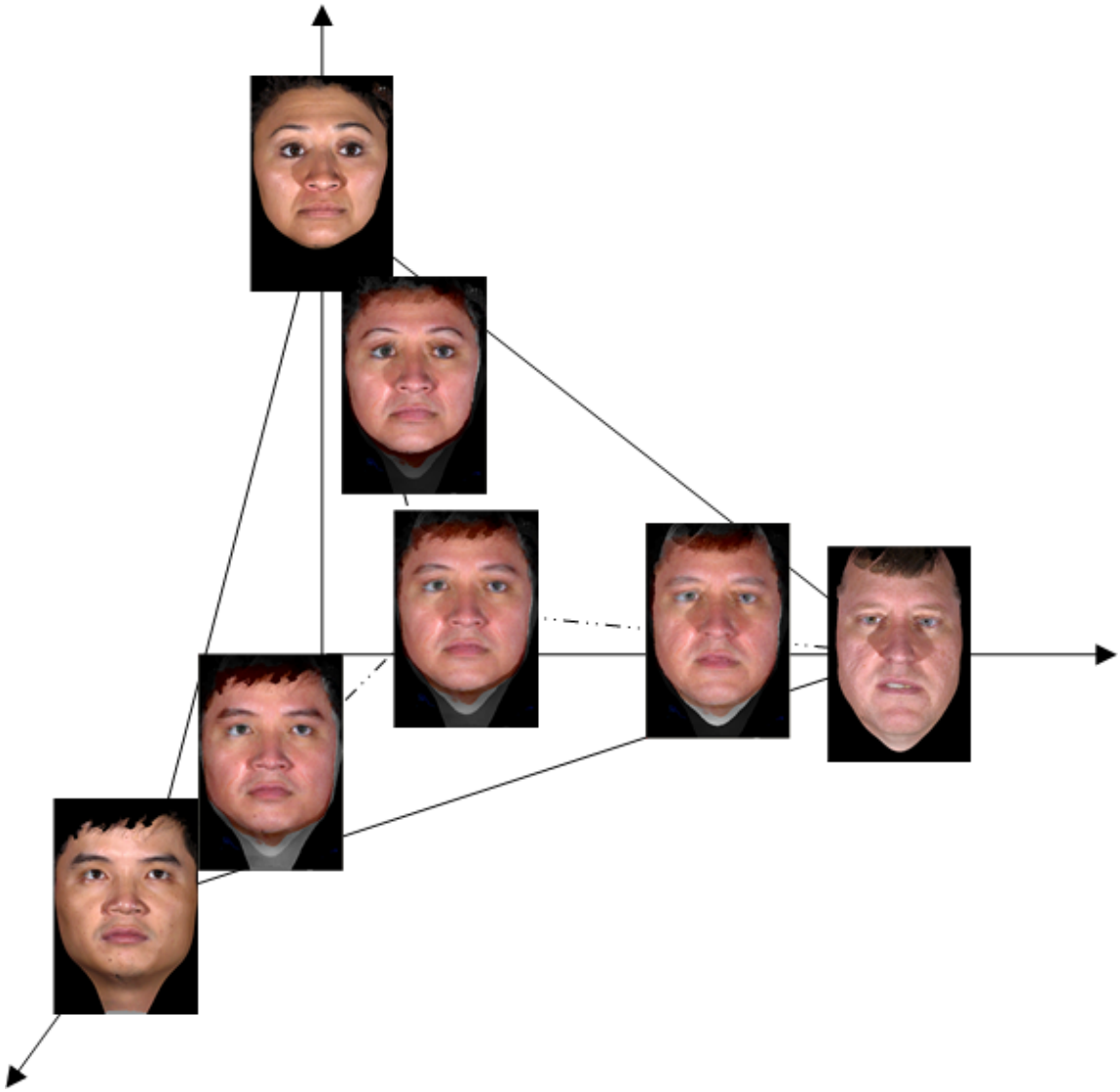


Fig. 9. Warped generic faces based on face warps derived from three example faces (for display purposes, the linear combinations of warped example faces are shown as texture images, although our scheme is based on range images).

To illustrate the idea of synthesizing 3D faces based on a linear combination of warps derived from example faces, consider the 3D feature space shown in Fig. 9. Here we have three example faces, each of which corresponds to a particular geometric transformation (shown in Fig. 10), namely, the one that makes the generic face best match that example face. Each axis represents one coefficient in a three-term weighted sum. Each example face lies at 1.0 along its corresponding axis. Further, any point in that 3D space corresponds to a particular linear combination of those three warps (shown in Fig. 11), and that in turn specifies a new face, namely, the one obtained by warping the generic face with the corresponding linear combination of warp parameters. If we restrict the coefficients to sum to unity, then the plane that passes through these three points defines the locus of all faces that can be generated by a linear combination of the three example face warp vectors. Any point on the triangle corresponds to a set of three coefficients (that sum to 1.0), and a linear combination of the three corresponding warps, using those weighting coefficients, produces a synthesized face by warping the generic face.

In feature extraction operation, the algorithm adjusts the coefficients to minimize the root mean square (RMS) error between the target face and the warped generic face. The coefficient vector corresponding to the minimum then constitutes a set of feature values that represents that face.

The feature extraction procedure of our scheme is illustrated in Fig. 12. This procedure starts with an initial coefficient vector $(w_1^1, \ldots, w_K^1)$, typically all equal to $1/K$. Based on this initial coefficient vector, the generic face is warped to generate a synthesized face S_1. Then the RMS error between S_1 and the target face T is computed. Using optimization techniques, the coefficient vector of the second iteration $(w_1^2, \ldots, w_K^2)$ can be found. This matching process continues through M iterations until the RMS error between the synthesized face S_M and the target face is small

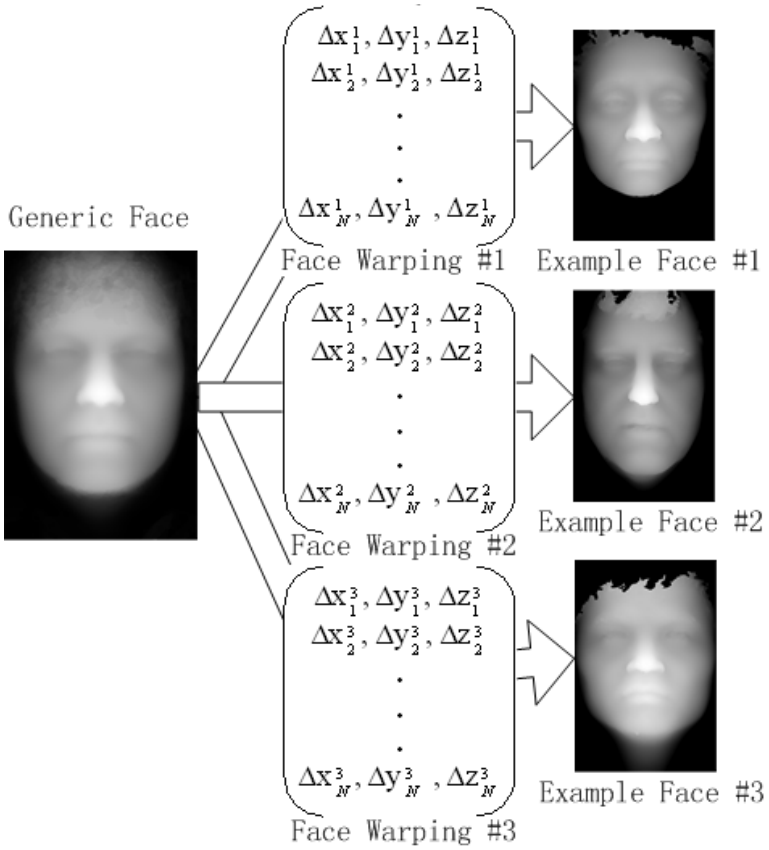


Fig. 10. Face warping from generic face to different example faces.

$$W_1\begin{pmatrix}\Delta X_1^1,\Delta Y_1^1,\Delta Z_1^1\\ \Delta X_2^1,\Delta Y_2^1,\Delta Z_2^1\\ \vdots\\ \Delta X_N^1,\Delta Y_N^1,\Delta Z_N^1\end{pmatrix}+W_2\begin{pmatrix}\Delta X_1^2,\Delta Y_1^2,\Delta Z_1^2\\ \Delta X_2^2,\Delta Y_2^2,\Delta Z_2^2\\ \vdots\\ \Delta X_N^2,\Delta Y_N^2,\Delta Z_N^2\end{pmatrix}+\cdots+W_K\begin{pmatrix}\Delta X_1^K,\Delta Y_1^K,\Delta Z_1^K\\ \Delta X_2^K,\Delta Y_2^K,\Delta Z_2^K\\ \vdots\\ \Delta X_N^K,\Delta Y_N^K,\Delta Z_N^K\end{pmatrix}=\begin{pmatrix}\Delta X_1(\mathbf{W}),\Delta Y_1(\mathbf{W}),\Delta Z_1(\mathbf{W})\\ \Delta X_2(\mathbf{W}),\Delta Y_2(\mathbf{W}),\Delta Z_2(\mathbf{W})\\ \vdots\\ \Delta X_N(\mathbf{W}),\Delta Y_N(\mathbf{W}),\Delta Z_N(\mathbf{W})\end{pmatrix}$$

Fig. 11. Linear combination of different face warpings.

enough. After that, the coefficients $(w_1^M, \ldots, w_K^M)$ are used as features to classify the target face T. In our scheme, the classification is based on Mahalanobis distance [23].

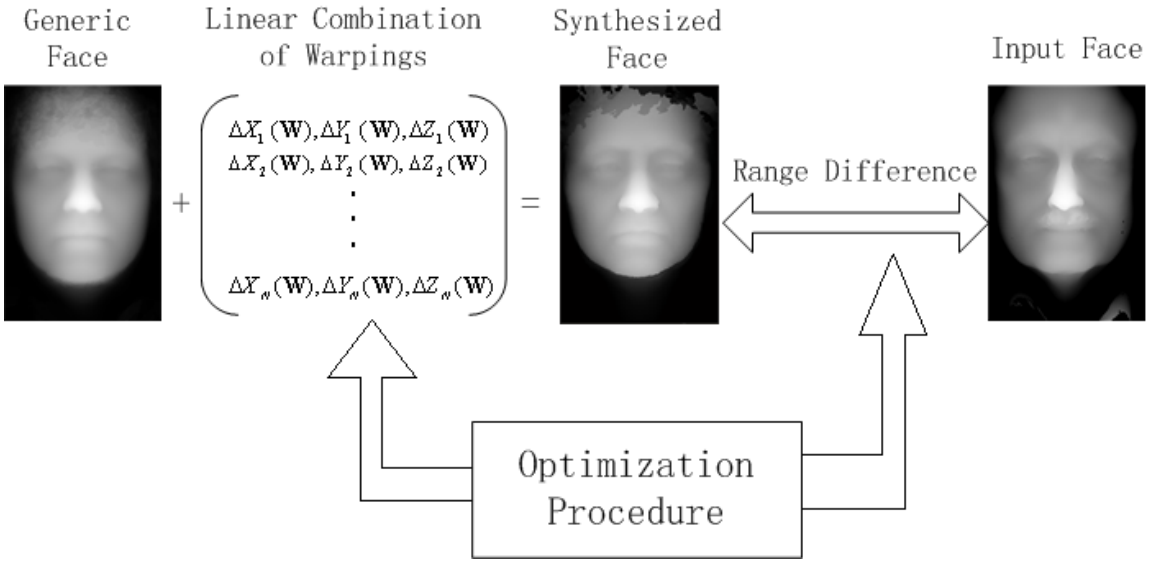


Fig. 12. The proposed feature extraction scheme.

Compared to the face recognition methods based on 3D morphable model [6], [7], [67], our scheme is different in three ways. First, our scheme utilizes 3D range images as inputs while the methods based on 3D morphable model [6], [7], [67] rely on 2D texture images. Therefore our scheme and their approaches belong to two different categories of face recognition systems and have different technical emphasis. In 2D face recognition, illumination and pose variations cause different appearances of the same humane face in 2D texture images. In order to achieve good recognition performance, it is crucial for 2D face recognition algorithms to handle the pose and illumination variations. The 3D morphable model proposed in [6], [7] and the extended approach [67] presented by L. Zhang *et al.* focus on handling this problem

and have given successful solutions. By contrast, in 3D face recognition, illumination variations do not affect the shapes of human faces on range images. Furthermore, pose variations can be solved by rotating and shifting human face shapes in range images to a standard pose. Hence pose and illumination variations are not important issues in 3D face recognition. In 3D face recognition category, holes, facial expressions and hair may change the overall shape of a 3D human face on range images, and become important technical issues. Our feature extraction scheme has the ability to extract both overall and detailed information from range images with the presence of holes, facial expression and hair. Second, our feature extraction scheme is automatic in contrast to the fitting procedure described in [6], [7], [67], which requires manual initialization. This is due to the fact that cost function defined in [6], [7], [67] is based on texture differences and has to be minimized by optimizing both shape and texture parameters of the 3D morphable model. Therefore good initial starting values of model parameters are critical and have to be discovered manually. While in our scheme, the cost function is the sum of depth differences and can be minimized by optimizing weighting coefficients only. Without involving of the texture parameters, it is feasible to achieve minimization automatically without falling into local minima. Third, compared to the methods [6], [7], [67] based on 3D morphable model, our feature extraction scheme is more efficient in computation. During the fitting procedure of 3D morphable model, in order to avoid occlusions and cast shadows, the 3D morphable model has to be rendered every 1000 iterations. Due to the large amount of vertices on 3D morphable model, this rendering procedure is inefficient in computation. By contrast, our feature extraction procedure is based on 2D warping of range images, or in a sense, 2.5D image transformation. The whole procedure does not require re-rendering of the synthesized face and thus our scheme is more efficient in computation. Note that this computation efficiency of our scheme is achieved by

sacrificing the simplicity of the image acquisition system. Compared to 2D texture images, 3D range images may require complicated image acquisition systems.

Compared to other 3D face recognition algorithms [1], [11], [14]–[19], [32], [43], [46], [47], [51], [57], [61], our method has several advantages. First, our scheme is insensitive to facial expressions. There are two techniques adopted in our scheme in order to solve the facial expressions problem: 1) matching the facial region that are insensitive to the face expressions; 2) including example faces with facial expressions in the example set. By using these techniques in our scheme, faces belonging to the same person with different facial expressions can be accurately differentiated from other people, which has been confirmed by our experimental results. Second, our scheme is robust to holes and facial expressions in range images. In our scheme, holes are adaptively avoided when comparing the differences between a target face and the synthesized face. The facial hair problem, on the other hand, is handled by: 1) including example faces with facial hair in the example set, 2) excluding the regions that can be easily affected by facial hair during the matching procedure, and 3) relying on the whole face region of a range image. By adding examples with hair in the example set, target faces with facial hair can be accurately identified. By matching the face region that is most likely to be unaffected by hair, the effect of facial hair in target faces can be further reduced. Furthermore, by relying on the whole facial region of a range image to identify a person, facial hair has limited influence on our extracted features and the recognition results. Consequently, our proposed scheme collects all useful information on the target face and successfully solves the problems caused by facial hair and holes. Finally, as a hybrid approach [68], our scheme not only consider the face region as a whole but also the details in target images. By combining the advantages of feature-based and holistic approaches [68], as proven by the experimental results, our scheme can achieve a high recognition rate.

Our scheme shares the same philosophy with the Active Appearance Models (AAM) [20]. Both schemes need manual labeling of landmark points to align images and involve image matching. However, there exist several differences. First of all, our scheme is a comprehensive solution for 3D face recognition comprising both feature extraction and classification, whereas AAM is a general modeling technique designed for studying the statistical nature of images. Second, our scheme is based on 3D range images while AAM is primarily based on 2D texture images. Third, the matching procedures in two schemes are different. In our scheme, we achieve image matching by utilizing stochastic Newton's method that updates derivative information in each iteration. In AAM, the derivatives are calculated once before the optimization starts.

To summarize, our proposed scheme can be outlined as follows

1. During design process, warp a generic model of a face to match each of K example faces, retaining the parameters of each of the K such warps in a warp vector.

2. In feature extraction procedure, warp the generic face using a geometric transformation that is a linear combination of the warp vectors that were developed for the example faces.

3. Adjust the coefficients in the weighted sum to minimize the RMS error between the input target face image and the warped generic face.

4. Take the coefficients that result in the minimum RMS error as extracted features that describe the face.

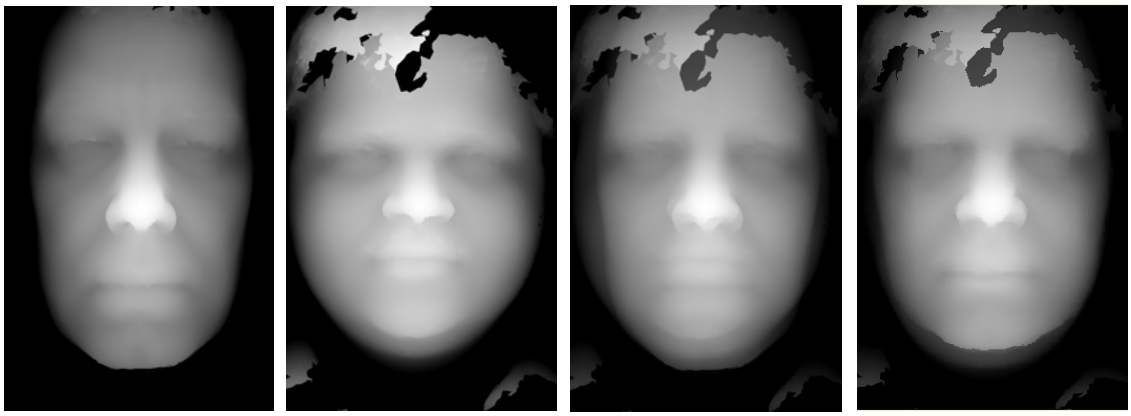
5. Use a linear classifier based on Mahalanobis distance to classify the extracted features.

B. Generic Face Warping

As the first stage of design process, the goal of generic face warping is to establish point-to-point correspondences between generic face and example faces. As illustrated previously, we can consider the warped generic face based on a combination of face warps derived from different example faces as a synthesized face. The main thrust is the generic face warping procedure that establishes the point-to-point correspondence between the generic face and example faces. In the sequel, we first describe the landmark points on the example faces and the generic face. Based on these landmark points, the generic face warping procedure that establishes the point-to-point correspondence between the generic face and example faces is introduced.

1. Necessity for Alignment

In order to generate a warped generic face by linearly combining the warps derived from example faces, we need to develop the proper face warpings between the generic face and example faces first. In other words, we need to find out the point-to-point correspondence between the generic face and the example faces. Otherwise, when different warps derived from example faces are combined together, points with different physical meanings will be added up, resulting in blurred faces. For instance, if we develop two face warpings between the generic face and two example faces (shown as example faces in Fig. 13 (a), (b)) without alignments, a combination of the face warpings will generate a blurred face shown in Fig. 13 (c). With the alignment, however, pixels with the same physical meaning are aligned together and generate an valid human face, as shown in Fig. 13 (d).



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

Fig. 13. Comparison of linear combination of two face waprings without/with alignment. (a) An example face with long chin. (b) A short example face. (c) Warped generic face without aligning the face warpings. (d) Warped generic face with aligned face warpings.

2. Landmark Points

In order to find out the point-to-point correspondences between the generic face and example faces, we first manually define seventy landmark points (e.g., eye corners, mouthe corners) on each of the K example faces and the generic face, as shown in Fig. 14. These landmark points all have distinct physical meanings and are easy to be marked manually[1]. Note that the number of landmark points can vary. In our face recognition scheme, we find seventy landmark points are enough to accurately describe the correspondences between the generic face and an example face. The x and y coordinates of these seventy landmark points are concatenated to generate

[1]Since it is easier to find out the points with distinguishable physical meanings in texture faces, the landmark points are manually marked in the corresponding texture images of the chosen example range face.

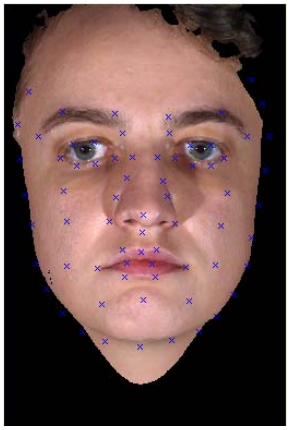


Fig. 14. Seventy landmark points on a texture face image. For display purposes, these landmark points are shown on a texture image.

position vectors $\mathbf{X}_L(i)$ and $\mathbf{Y}_L(i)$, $i = 1, \ldots, K$. As we mentioned previously in Section A, a face in the example set is chosen to be the generic face G. The position vectors of the landmark points on G are denoted as $\mathbf{X}_g$ and $\mathbf{Y}_g$. With these position vectors, the displacement vectors $\boldsymbol{\Delta}\mathbf{X}_L(i)$ and $\boldsymbol{\Delta}\mathbf{Y}_L(i)$ $(i = 1, \ldots, K)$ are defined as

$$\begin{cases} \boldsymbol{\Delta}\mathbf{X}_L(i) = \mathbf{X}_L(i) - \mathbf{X}_g \\ \boldsymbol{\Delta}\mathbf{Y}_L(i) = \mathbf{Y}_L(i) - \mathbf{Y}_g, \end{cases} \tag{3.1}$$

$i = 1, \ldots, K$. Based on these displacement vectors, we can further generate new displacement vectors

$$\begin{cases} \Delta\mathbf{X}_l(\mathbf{W}) = \sum_{i=1}^{K} w_i \cdot \Delta\mathbf{X}_L(i) \\ \Delta\mathbf{Y}_l(\mathbf{W}) = \sum_{i=1}^{K} w_i \cdot \Delta\mathbf{Y}_L(i), \end{cases} \tag{3.2}$$

where w_i $(i = 1, \ldots, K)$ are the weighting coefficients. Depending on $\mathbf{W} = (w_1, w_2, \ldots, w_K)^t$, $\Delta\mathbf{X}_l(\mathbf{W})$ and $\Delta\mathbf{Y}_l(\mathbf{W})$ actually represent a set of displacements of

the seventy landmark points on the generic face G. As we will see in the following subsection, they play an important role in aligning face warpings.

3. Face Warping

In this section, we describe the warping procedure that calculates the displacements of every point on the generic face G based on the displacement vectors $\Delta\mathbf{X}_l(\mathbf{W})$, $\Delta\mathbf{Y}_l(\mathbf{W})$. After this procedure, a point (x, y) in the generic face G will be mapped to a point $(x + \Delta x(\mathbf{W}, x, y), y + \Delta y(\mathbf{W}, x, y))$ in the warped face $S(\mathbf{W})$. The generic face and example faces used in this procedure are all rendered in range images of size 750×500.

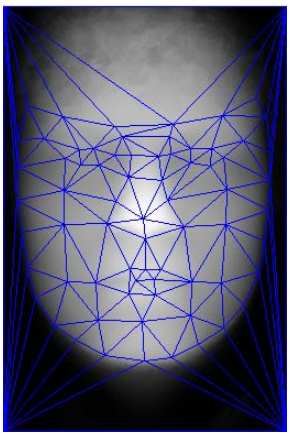


Fig. 15. A range face image divided by Delaunay triangles.

We first divide the generic face into Delaunay triangles [24] based on the seventy landmark points defined by $\mathbf{X}_g$ and $\mathbf{Y}_g$, as shown in Fig. 15. The Delaunay triangulation of a set of points in the plane is a set of triangles connecting the points satisfying an "empty circle" property: the circumcircle of each triangle does not contain any of

the points. We employ this triangulation because a fast method for such triangulation is readily available. The coordinates $\mathbf{X}_g$ and $\mathbf{Y}_g$ of the triangle vertices[2] are then mapped to $\mathbf{X}_l(\mathbf{W}) = \mathbf{X}_g + \Delta\mathbf{X}_l(\mathbf{W})$ and $\mathbf{Y}_l(\mathbf{W}) = \mathbf{Y}_g + \Delta\mathbf{Y}_l(\mathbf{W})$, respectively.

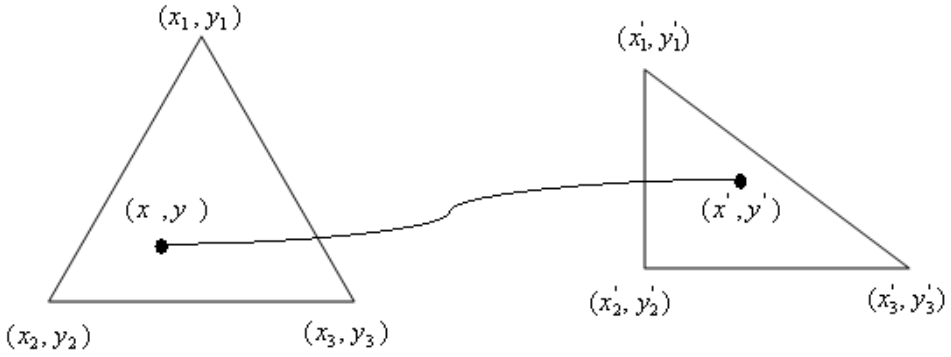


Fig. 16. Displacement calculation of pixels inside a Delaunay triangle.

We can focus on a pair of such corresponding Delaunay triangles shown in Fig. 16, in which the triangle T_1 are mapped to the triangle T_2. The vertices (x_1, y_1), (x_2, y_2) and (x_3, y_3) of T_1 are then mapped to the vertices $(x_1'(\mathbf{W}), y_1'(\mathbf{W}))$, $(x_2'(\mathbf{W}), y_2'(\mathbf{W}))$ and $(x_3'(\mathbf{W}), y_3'(\mathbf{W}))$ of T_2, respectively. Now given a point (x, y) inside T_1, we want to find out the corresponding position $(x'(\mathbf{W}), y'(\mathbf{W}))$ in T_2. If we restrict ourselves to linear mappings, we have

$$\begin{cases} x'(\mathbf{W}) = a(\mathbf{W}) \cdot x + b(\mathbf{W}) \cdot y + c(\mathbf{W}) \\ y'(\mathbf{W}) = d(\mathbf{W}) \cdot x + e(\mathbf{W}) \cdot y + f(\mathbf{W}), \end{cases} \tag{3.3}$$

where a($\mathbf{W}$)—f($\mathbf{W}$) are the coefficients that need to be determined. Because of the

[2]These vertices of Delaunay triangles are actually the seventy landmark points on the generic face.

correspondences between vertices of T_1 and T_2, we have

$$\begin{cases} x_1'(\mathbf{W}) = a(\mathbf{W}) \cdot x_1 + b(\mathbf{W}) \cdot y_1 + c(\mathbf{W}) \\ x_2'(\mathbf{W}) = a(\mathbf{W}) \cdot x_2 + b(\mathbf{W}) \cdot y_2 + c(\mathbf{W}) \\ x_3'(\mathbf{W}) = a(\mathbf{W}) \cdot x_3 + b(\mathbf{W}) \cdot y_3 + c(\mathbf{W}), \end{cases} \tag{3.4}$$

and

$$\begin{cases} y_1'(\mathbf{W}) = d(\mathbf{W}) \cdot x_1 + e(\mathbf{W}) \cdot y_1 + f(\mathbf{W}) \\ y_2'(\mathbf{W}) = d(\mathbf{W}) \cdot x_2 + e(\mathbf{W}) \cdot y_2 + f(\mathbf{W}) \\ y_3'(\mathbf{W}) = d(\mathbf{W}) \cdot x_3 + e(\mathbf{W}) \cdot y_3 + f(\mathbf{W}). \end{cases} \tag{3.5}$$

From (3.4) and (3.5), we can obtain

$$\begin{bmatrix} x_1'(\mathbf{W}) \\ x_2'(\mathbf{W}) \\ x_3'(\mathbf{W}) \end{bmatrix} = M \begin{bmatrix} a(\mathbf{W}) \\ b(\mathbf{W}) \\ c(\mathbf{W}) \end{bmatrix} \tag{3.6}$$

and

$$\begin{bmatrix} y_1'(\mathbf{W}) \\ y_2'(\mathbf{W}) \\ y_3'(\mathbf{W}) \end{bmatrix} = M \begin{bmatrix} d(\mathbf{W}) \\ e(\mathbf{W}) \\ f(\mathbf{W}) \end{bmatrix}, \tag{3.7}$$

where $M = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix}$. Note that M is invertible as long as T_1 has nonzero area, so we have

$$\begin{bmatrix} a(\mathbf{W}) \\ b(\mathbf{W}) \\ c(\mathbf{W}) \end{bmatrix} = M^{-1} \begin{bmatrix} x_1'(\mathbf{W}) \\ x_2'(\mathbf{W}) \\ x_3'(\mathbf{W}) \end{bmatrix}, \tag{3.8}$$

and

$$\begin{bmatrix} d(\mathbf{W}) \\ e(\mathbf{W}) \\ f(\mathbf{W}) \end{bmatrix} = M^{-1} \begin{bmatrix} y'_1(\mathbf{W}) \\ y'_2(\mathbf{W}) \\ y'_3(\mathbf{W}) \end{bmatrix}. \tag{3.9}$$

Let $\mathbf{X}'_t(\mathbf{W}) = (x'_1(\mathbf{W}), x'_2(\mathbf{W}), x'_3(\mathbf{W}))^t$, $\mathbf{Y}'_t(\mathbf{W}) = (y'_1(\mathbf{W}), y'_2(\mathbf{W}), y'_3(\mathbf{W}))^t$ and $\mathbf{P} = (x, y, 1)^t$, from (3.8) and (3.9), we can rewrite (3.3) as

$$\begin{cases} x'(\mathbf{W}) = \mathbf{P}^t M^{-1} \mathbf{X}'_t(\mathbf{W}) \\ y'(\mathbf{W}) = \mathbf{P}^t M^{-1} \mathbf{Y}'_t(\mathbf{W}). \end{cases} \tag{3.10}$$

Having established the mapping between two arbitrary corresponding triangles T_1 and T_2, by giving the corresponding vertices of triangles, we can map a point (x, y) in triangle T_1 to a point (x', y') in triangle T_2. The displacements of the position (x, y) can also be obtained as $\Delta x(\mathbf{W}) = x'(\mathbf{W}) - x$ and $\Delta y(\mathbf{W}) = y'(\mathbf{W}) - y$.

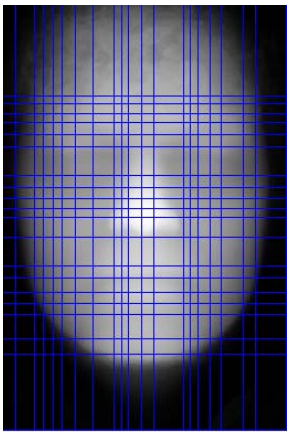


Fig. 17. A range face image divided by grids based on landmark points.

If we draw vertical and horizontal lines passing through the landmark points of

the generic face G, we can further divide G into more than 300 small grids, as shown in Fig. 17. Clearly the crossing points of these grids are either landmark points themselves or fall into different Delaunay triangles. Displacements of the crossing points that fall into Delaunay triangles can be obtained by using (3.10). When the displacements of all crossing points of the grids are determined, we can further calculate the displacements of the points inside these grids using bilinear interpolation. Suppose a particular grid has four vertices (x_i, y_i), $i = 1, 2, 3, 4$, let the corresponding displacements of the vertices be $(\Delta x_i(\mathbf{W}), \Delta y_i(\mathbf{W}))$,$i = 1, 2, 3, 4$, the displacements of a pixel with the position (x, y) inside a grid can be calculated as follows by using bilinear interpolation [13],

$$\begin{aligned}
\Delta x(\mathbf{W}, x, y) = & (\Delta x_3(\mathbf{W}) - \Delta x_1(\mathbf{W}))\frac{y - y_1}{y_4 - y_2 + 1} + (\Delta x_2(\mathbf{W}) - \Delta x_1(\mathbf{W}))\frac{x - x_1}{x_4 - x_1 + 1} \\
& + (\Delta x_1(\mathbf{W})) + \Delta x_4(\mathbf{W}) - \Delta x_2(\mathbf{W}) - \Delta x_3(\mathbf{W}))\frac{x - x_1}{x_4 - x_1 + 1}\frac{y - y_1}{y_4 - y_2 + 1} \\
& + \Delta x_1(\mathbf{W}), \\
\Delta y(\mathbf{W}, x, y) = & (\Delta y_3(\mathbf{W}) - \Delta y_1(\mathbf{W}))\frac{y - y_1}{y_4 - y_2 + 1} + (\Delta y_2(\mathbf{W}) - \Delta y_1(\mathbf{W}))\frac{x - x_1}{x_4 - x_1 + 1} \\
& + (\Delta y_1(\mathbf{W}) + \Delta y_4(\mathbf{W}) - \Delta y_2(\mathbf{W}) - \Delta y_3(\mathbf{W}))\frac{x - x_1}{x_4 - x_1 + 1}\frac{y - y_1}{y_4 - y_2 + 1} \\
& + \Delta y_1(\mathbf{W}).
\end{aligned} \tag{3.11}$$

Note that generic face and example faces in the face warping procedure are in range images with size 750×500. We found that face warping procedure can produce satisfying point-to-point correspondence between example faces even with non-rigidities existed (facial expressions, etc.). From the above warping processes, we have achieved the goal of obtaining the displacements of every point on the generic face based on the displacements of landmark points.

After the above warping procedure, the displacement of every point on the generic face G has been calculated, which clearly relies on the weighting coefficients $\mathbf{W}$. These

displacements can further be considered as displacement images $\Delta X(\mathbf{W}, x, y)$ and $\Delta Y(\mathbf{W}, x, y)$, which are indexed by a pixel's position (x, y). Let the ith component w_i of $\mathbf{W}$ be one and the rest be zero, we can obtain the displacement images of the ith example face (denoted as E_i) $\Delta X_i(x, y)$ and $\Delta Y_i(x, y)$, $i = 1, \ldots, K$. With these displacement images, the pixel correspondences between the example faces and the generic face G are established. This accomplishes our goal of seeking the point-to-point correspondences between example faces and the generic face. In the next section, we will use these face warpings derived from example faces to generate synthesized faces using linear combination.

C. Linear Combination of Face Warpings

In this section, we will introduce the second stage of the design process. During this stage, face warpings derived from example faces are linearly combined together to generate different synthesized faces based on different weighting coefficients. In addition, we describe the selection process of example faces, which is used to increase the representability of synthesized faces.

1. Synthesized Faces from Linear Combination of Face Warpings

The above described warping procedure gives us the displacement images $\Delta X_i(x, y)$ and $\Delta Y_i(x, y)$ $(i = 1, \ldots, K)$ of example faces. We can further calculate the displacement image of the ith example face E_i $(i = 1, \ldots, K)$ in the Z direction (i.e., the depth of the face images)

$$\Delta Z_i(x, y) = E_i(x + \Delta X_i(x, y), y + \Delta Y_i(x, y)) - G(x, y), \tag{3.12}$$

where x and y are the coordinates of the image. From (3.12), we can reconstruct the example face E_i $(i = 1, \ldots, K)$ as follows

$$E_i(x, y) = \Delta Z_i(x - \Delta X_i(x, y), y - \Delta Y_i(x, y)) \tag{3.13}$$

$$+G(x - \Delta X_i(x, y), y - \Delta Y_i(x, y)). \tag{3.14}$$

Moreover, we can utilize the obtained displacement images of the example images $\Delta X_i(x, y)$, $\Delta Y_i(x, y)$ and $\Delta Z_i(x, y)$ $(i = 1, \ldots, K)$ to generate new displacement images

$$\begin{cases} \Delta X(\mathbf{W}, x, y) = \sum_{i=1}^{K} w_i \cdot \Delta X_i(x, y) \\ \Delta Y(\mathbf{W}, x, y) = \sum_{i=1}^{K} w_i \cdot \Delta Y_i(x, y) \\ \Delta Z(\mathbf{W}, x, y) = \sum_{i=1}^{K} w_i \cdot \Delta Z_i(x, y), \end{cases} \tag{3.15}$$

where $\Delta X(\mathbf{W}, x, y)$, $\Delta Y(\mathbf{W}, x, y)$ and $\Delta Z(\mathbf{W}, x, y)$ can additionally be used to create a new synthesized face $S(\mathbf{W})$

$$S(\mathbf{W}, x + \Delta X(x, y), y + \Delta Y(x, y)) = G(x, y) + \Delta Z(\mathbf{W}, x, y). \tag{3.16}$$

Therefore, with the aligned face warps derived from example range faces, we are able to generate new synthesized faces using different weighting coefficients w_i, $i = 1, \ldots, K$. For instance, if we allocate each displacement image the same weight $1/K$, we can generate the average warped generic face, as shown in Fig. 18.

2. Selection of Example Faces

Our example set consists of carefully selected range faces, which are denoted as example faces in our scheme. The selection of these example faces is performed in such a way that faces with representative characteristics (for example, faces that are thin, wide, long, short, etc.) will be added. This is because the representability of the

Fig. 18. The average range face of K examples faces.

synthesized range face depends on the chosen example faces. This can be illustrated in the feature space. Projected into the feature space, the example faces can be considered as points in the feature space. The valid synthesized faces will fall into the "cloud" composed by the example faces, while the unreal synthesized faces will fall out of it, as illustrated in Fig. 19. The larger the "cloud", the wider range of valid synthesized faces will be generated. Hence we should select the extreme faces (e.g., long faces, short faces) as example faces, so that the "cloud" composed by the example faces will be as large as possible. Note that once the example faces in the example set are determined, we will not change it during the feature extraction procedure regardless of the input images. Hence the example set is irrelevant to the training set used in the classification, which will be introduced later.

In our scheme, to find out extreme faces, PCA technique is applied to analyze a number of range faces in the data set. From Section 1, we know that for a range face E, we can obtain the corresponding displacement images ΔX, ΔY and ΔZ. These

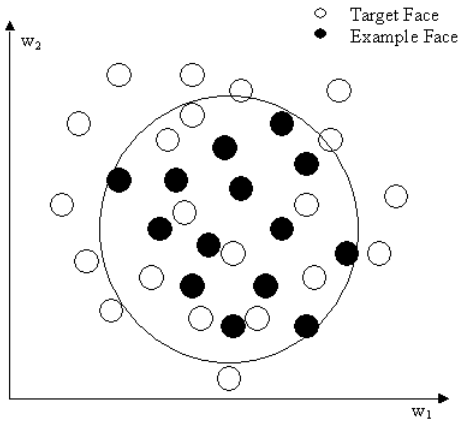


Fig. 19. Range faces in the 2D feature space.

displacement images can be concatenated as a vector $\mathbf{V}$. Let the length of vector $\mathbf{V}$ be L. The purpose of the example face selection procedure is to select K example faces out of H range images in the data set. By calculating the mean vector $\bar{\mathbf{V}}$ of these H vectors $\mathbf{V}_i$ $(i = 1, \ldots, H)$, we can obtain the data matrix $D = (\mathbf{V}_1 - \bar{\mathbf{V}}, \ldots, \mathbf{V}_H - \bar{\mathbf{V}})$. The covariance matrix of D can therefore be expressed as $P = (\frac{1}{H})DD^T$. In order to perform PCA, the eigenvectors $\mathbf{U}_i$ $(i = 1, \ldots, H)$ should be calculated from P. Note that the covariance matrix P is $L \times L$, where L can be a very large number. In order to save calculation complexity, we compute the eigenvectors $\mathbf{U}'_i$ $(i = 1, \ldots, H)$ from $P' = (\frac{1}{H})D^T D$. Let the corresponding eigenvalue of an eigenvector $\mathbf{U}'_i$ be λ_i. We have

$$D^T D\mathbf{U}'_i = \lambda_i \mathbf{U}'_i. \tag{3.17}$$

By multiplying the matrix D to the left side of the above equation, the following

equation can be obtained

$$DD^T D\mathbf{U}'_i = \lambda_i D\mathbf{U}'_i, \tag{3.18}$$

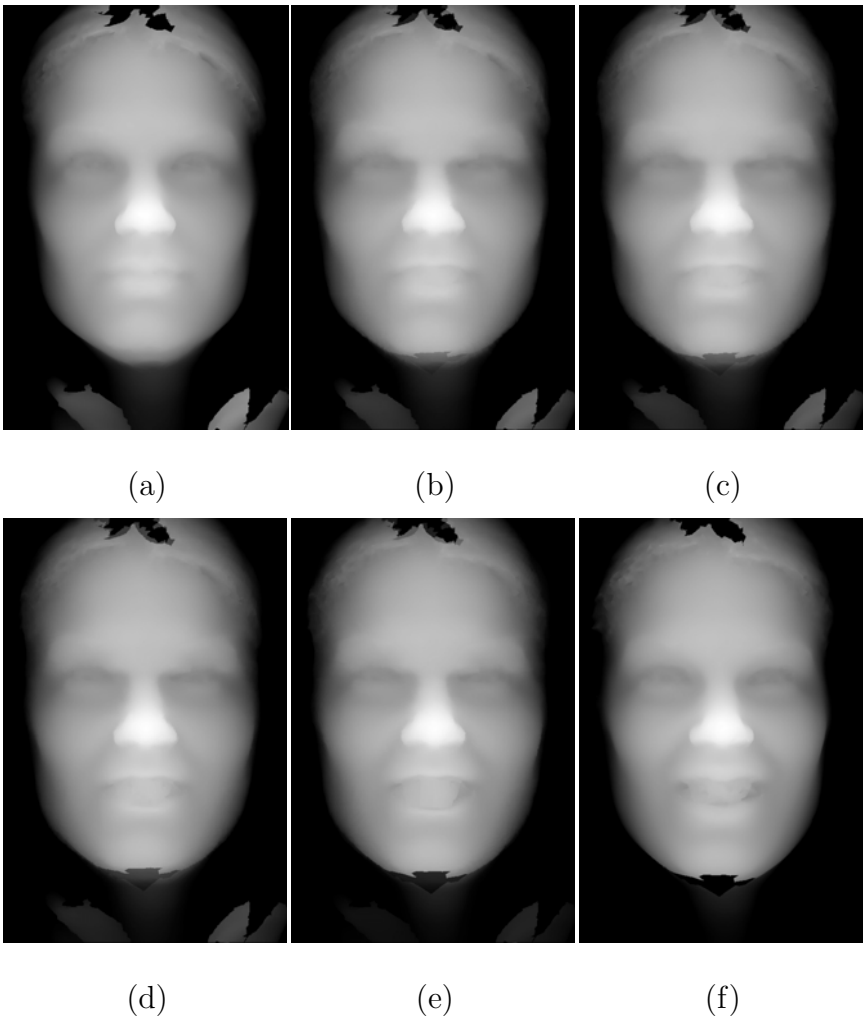
where $D\mathbf{U}'_i$ is clearly an eigenvector of the covariance matrix P. We can then sort the eigenvectors $(D\mathbf{U}'_1, \ldots, D\mathbf{U}'_H)$ according to the values of their corresponding eigenvalues, from large to small. Furthermore, since these eigenvectors may not be normal vectors, it is necessary to normalize them. After the sorting and normalization, we can obtain the eigenvectors $(\mathbf{U}_1, \ldots, \mathbf{U}_H)$ of P. Each sample vector $\mathbf{V}_i$, $(i = 1, \ldots, H)$, is then projected to an eigenvector to get its coordinate value $\mathbf{U}_i\mathbf{V}_i^T$. Since extreme faces have extreme coordinate values on these eigenvectors (maximum or minimum), based on the coordinate values of the sample vectors and the significance of the corresponding eigenvectors, we are able to choose K extreme faces from H range images in a data set.

Fig. 20. Example faces with/without facial expressions in the example set.

In order to reduce the noise in target images caused by facial expressions, we add example faces with facial expressions to the example set, as illustrated in Fig.

20. These example faces with facial expressions can be linearly combined with those example faces with neutral expressions to generate a series of synthesized faces. Depending on the weighting coefficients of the linear combination, different synthesized faces with expressions can be obtained. As shown in Fig. 21, by adding the example face with laughing expression (shown in Fig. 21 (e)) to the example with neutral expression (shown in Fig. 21 (a)), we can get a range of synthesized faces shown in Fig. 21 (b), (c), (d). This is achieved by giving different weighting coefficients to the linear combinations of face warpings derived from these two example faces. Therefore, our scheme can produce synthesized faces close to a target face with facial expression and extract reliable features. This is one of the two techniques used in our scheme in order to solve the facial expression problem. The other technique, as we described in Section 2, tries to select matching regions that tend to be unaffected by facial expressions. Note that for illustration purposes, the example faces shown in Fig. 20 are displayed as texture images, although in the example set the example faces are stored as range images.

After the design process, we can generate different synthesized faces from different linear combinations of face warpings by adjusting the weighting coefficients. In the next section, we introduce the feature extraction procedure that can automatically extracts features (weighting coefficients) from a target range image.



(a) (b) (c)

(d) (e) (f)

Fig. 21. A range of synthesized faces with expressions can be obtained from example faces. (a) An example face with neutral expression. (b)—(e) A series of synthesized faces with laughing expressions can be obtained. (f) An example face with laughing expression.

D. Feature Extraction Procedure

Given a target range image T with a novel face, the goal of the feature extraction procedure is to extract reliable facial features that can differentiate the face from others. Based on the aligned face warps derived from example faces, we design an algorithm to make the warped generic face (synthesized face) as similar as possible to the target face. In this section, we first introduce the cost function that describes the range differences between the synthesized face and a target face. Then the optimization procedure that minimizes the cost function is presented.

1. Cost Function

Suppose we have K example face E_i $(i = 1, \ldots, K)$, the generic face G and a target face T. Given a weighting vector $\mathbf{W}$, let the synthesized range face be $S(\mathbf{W})$. For a certain position (x, y) on G, the squared range difference between $S(\mathbf{W})$ and T becomes

$$\begin{aligned} C(\mathbf{W}, x, y) = & (S(\mathbf{W}, x + \Delta X(\mathbf{W}, x, y), y + \Delta Y(\mathbf{W}, x, y)) \\ & -T(x + \Delta X(\mathbf{W}, x, y), y + \Delta Y(\mathbf{W}, x, y)))^2 \\ = & (G(x, y) + \Delta Z(\mathbf{W}, x, y) \qquad (3.19) \\ & -T(x + \Delta X(\mathbf{W}, x, y), y + \Delta Y(\mathbf{W}, x, y)))^2. \qquad (3.20) \end{aligned}$$

In order to measure the similarity between $S(\mathbf{W})$ and T, we define the cost function as the sum of the squared difference

$$\begin{aligned} C(\mathbf{W}, \Omega) = & \sum_{(x,y)\in\Omega} (G(x, y) + \Delta Z(\mathbf{W}, x, y) \\ & -T(x + \Delta X(\mathbf{W}, x, y), y + \Delta Y(\mathbf{W}, x, y)))^2, \qquad (3.21) \end{aligned}$$

where Ω is a set containing selected points on the generic face. As we will see later, dynamic selection of points in Ω will allow us to avoid holes in target faces. Furthermore, by choosing the points in a region that tend not to be affected by facial hair and facial expressions, our scheme can be robust to facial hair and expressions.

2. Matching Region Selection

As just mentioned in Section 1, we should dynamically select points in Ω so that holes will be avoided in the matching procedure. To understand the adaptive hole avoiding strategy, let us consider picking a point $P = (x, y)$ the point set Ω of the generic face, as illustrated in Fig. 22. Based on the current weighting coefficients $\mathbf{W}$, this point will be compared to the point $P' = (x + \Delta X(\mathbf{W}, x, y), y + \Delta Y(\mathbf{W}, x, y))$ in the target image. In our scheme, when the gray value of P' in the target image is zero, we believe that P' has fallen into a hole. When that happens, we simply abandon the point P from Ω. From this selection strategy, the points that fall into holes (with gray level as zero) will be adaptively avoided in the matching procedure.

In order to avoid the facial hair and facial expression in a target face, we define the matching region Ψ (shown in Fig. 23) in the generic face. In choosing the matching region, the regions that below the lip and above the eyebrows are discarded to avoid the effect caused by facial hair and facial expressions. During the matching procedure, only points inside the matching region will be selected to the point set Ω. In our scheme, this matching region selection is combined with the techniques mentioned in Section A to achieve reliable feature extraction from target faces.

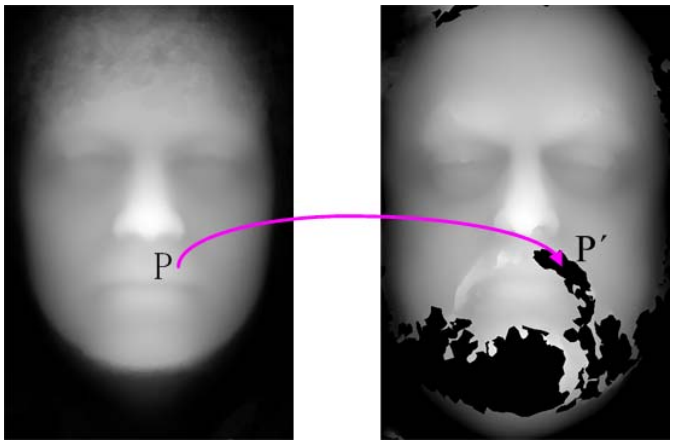


Fig. 22. The adaptive hole avoiding mechanism.

Fig. 23. The matching region on the generic face.

3. Optimization

After establishing the cost function as the measurement of the similarity between $S(\mathbf{W})$ and T, our goal is then to minimize $C(\mathbf{W}, \Omega)$ by optimizing $\mathbf{W}$. The minimization of the cost function $C(\mathbf{W}, \Omega)$ can be achieved by many optimization methods. Here we choose Newton's method [7], which is based on the calculation of derivatives of $C(\mathbf{W}, \Omega)$, because of its quick convergence. Compared to the optimization procedure in [7], our optimization is automatic.

The derivatives of $C(\mathbf{W}, \Omega)$ with respect to the ith weighting coefficient w_i are given as

$$\begin{aligned}\frac{\partial C(\mathbf{W},\Omega)}{\partial w_i} = \sum_{(x,y)\in\Omega} & 2(G(x,y) + \Delta Z(\mathbf{W},x,y) \\ & -T(x+\Delta X(\mathbf{W},x,y), y+\Delta Y(\mathbf{W},x,y))) \\ & (\frac{\partial \Delta Z(\mathbf{W},x,y)}{\partial w_i} \\ & -(\frac{\partial T(x+\Delta X(\mathbf{W},x,y), y+\Delta Y(\mathbf{W},x,y))}{\partial (x+\Delta X(\mathbf{W},x,y))} \cdot \frac{\partial \Delta X(\mathbf{W},x,y)}{\partial w_i}) \\ & -(\frac{\partial T(x+\Delta X(\mathbf{W},x,y), y+\Delta Y(\mathbf{W},x,y))}{\partial (y+\Delta Y(\mathbf{W},x,y))} \cdot \\ & \frac{\partial \Delta Y(\mathbf{W},x,y)}{\partial w_i})), \end{aligned} \tag{3.22}$$

and

$$\begin{aligned}\frac{\partial^2 C(\mathbf{W},\Omega)}{\partial w_i^2} = \sum_{(x,y)\in\Omega} & 2(\frac{\partial \Delta Z(\mathbf{W},x,y)}{\partial w_i} \\ & -(\frac{\partial T(x+\Delta X(\mathbf{W},x,y), y+\Delta Y(\mathbf{W},x,y))}{\partial (x+\Delta X(\mathbf{W},x,y))} \cdot (\frac{\partial \Delta X(\mathbf{W},x,y)}{\partial w_i})) \\ & -(\frac{\partial T(x+\Delta X(\mathbf{W},x,y), y+\Delta Y(\mathbf{W},x,y))}{\partial (y+\Delta Y(\mathbf{W},x,y))} \cdot \\ & (\frac{\partial \Delta Y(\mathbf{W},x,y)}{\partial w_i})))^2. \end{aligned} \tag{3.23}$$

Starting from all weighting coefficients equal to $1/K$, in each iteration of Newton's

method, we update the next weighting coefficients as follows

$$\mathbf{W} = \mathbf{W} - \lambda H^{-1} \nabla C, \tag{3.24}$$

where $H^{-1} \approx diag(1/\frac{\partial^2 C(\mathbf{W})}{\partial w_i^2})$ $(i = 1, \ldots, K)$ is the inverse Hessian matrix and $\lambda \ll 1$ is the learning rate.

We implement two versions of Newton's method to minimize the cost function. The first version is the standard Newton's method. We first downsample the generic face G into different subsets based on different scale factors. These subsets are then used as the set Ω in the optimization. It is obvious that proper downsampling scale factor is important to achieve the balance of quick convergence and similarity between synthesized face $S(\mathbf{W})$ and a target face T. Let us consider the extreme case of choosing all points in the generic face to form the Ω. When the minimization of the cost function is achieved, we can obtain a synthesized face closest to the target face T. However, the convergence could be slow because of the heavy computation in each iteration. On the other hand, if we downsample the generic face using a high scale factor, the size of the set Ω will be small and the computation will be fast in each iteration. But in this case, even if the cost function $C(\mathbf{W}, \Omega)$ is minimized, the actual RMS error between $S(\mathbf{W}^*)$ and T may not be able to reach the minima.

The second version is a stochastic Newton's method. In each iteration, a number of points are randomly chosen to the set Ω from the matching region on the generic face. So we should be able to determine the number of points in the set Ω in each iteration. Here we face a tradeoff of choosing proper size of Ω. On one hand, smaller size of Ω can reduce the computing time of the matching procedure in each iteration. But from (3.24), we can see that it will also cause the instability of the changing value $\lambda H^{-1} \nabla C$ in each iteration. In order to smooth the changing of the weighting

coefficients, small learning rate has to be given, resulting in overall slow convergence to the global minima even if computation in each iteration is fast. On other hand, a larger Ω can bring more stability to $H^{-1}\nabla C$. With more stability, we can give the learning rate λ a larger value, so that even if the computing time in each iteration is longer, the overall matching procedure can be faster. In our scheme, the size of Ω and learning rate λ are tested and tuned to reach the best performance.

Both versions of our proposed optimization method have been tested on more than 600 different target images, none of the optimization procedures falls into local minima, which demonstrates the robustness of the proposed optimization method.

After the feature extraction procedure, we obtain the optimized weighting coefficients $\mathbf{W}^*$, which can be used as features for classification.

E. Feature Classification Using Mahalanobis Distance

Many classifiers can be used to classify the obtained features from the feature extraction procedure. In our proposed scheme, a linear classifier based on Mahalanobis distance [23] is used, which yields good results as we will see later in Section F.

1. Feature Preprocessing

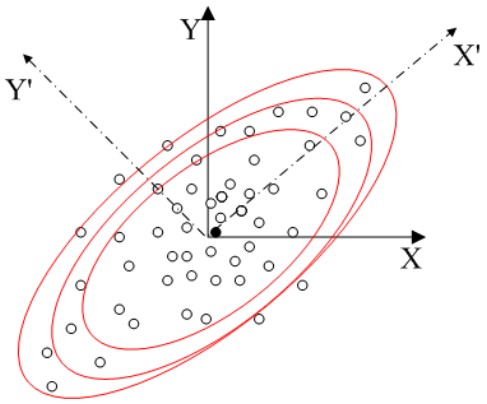


Fig. 24. Principle component analysis.

The high dimensionality of the features (weighting coefficients) can be redundant, therefore we again use the PCA to find out the dominant dimensions. Let us take a look at the samples on the XY plane illustrated in Fig. 24. Using PCA, two orthonormal vectors $\mathbf{X}'$ and $\mathbf{Y}'$ can be found. As shown in Fig. 24, these two vectors have the following properties: 1) the covariance of samples is maximized by projecting the samples to the normal vector $\mathbf{X}'$, and 2) the normal vector $\mathbf{Y}'$ is orthogonal to $\mathbf{X}'$. In addition to the two vectors $\mathbf{X}'$ and $\mathbf{Y}'$, PCA also produces the corresponding eigenvalues $\sigma_{x'}$ and $\sigma_{y'}$, respectively. These eigenvalues represent the variances of the projected samples in the directions defined by $\mathbf{X}'$ and $\mathbf{Y}'$. In our example, it is obvious that $\sigma_{x'}$ is larger than $\sigma_{y'}$. Therefore, if we discard the vector $\mathbf{Y}'$ and project all samples to the vector $\mathbf{X}'$, the dimensionality will be reduced without losing too much information. From this 2D example, we can generalize the PCA technique to higher dimensional space. As we will see later in Section 4, in our scheme, PCA

reduces more than half of the dimensionality of our extracted feature vectors to lower dimensions for analysis.

In addition to reducing the dimensionality of samples, here we use PCA as a preprocessing technique to achieve another goal. As we mention previously in Section A, our linear classifier is based on Mahalanobis distance. In order to calculate the Mahalanobis distance, the inverse of a data set's covariance matrix has be to calculated. Therefore we require the covariance matrix to be invertible. From linear algebra we learn that a square matrix is invertible if and only if its determinant is not zero. In other words, none of the eigenvalues of an invertible covariance matrix should be zero. Given a $N \times N$ covariance matrix, from PCA we can obtain a series of eigenvectors $(\mathbf{E}_1, \ldots, \mathbf{E}_N)$ sorted by their corresponding eigenvalues $(\omega_1, \ldots, \omega_N)$, from large to small. If we select the eigenvectors with large eigenvalues and discard the eigenvectors with small eigenvalues, by projecting the samples to the selected eigenvectors, we obtain a new set of samples with non-singular covariance matrix. This way, PCA can help prevent singular covariance matrices from occurring and ensure accurate calculation of the Mahalanobis distance.

2. Mahalanobis Distance

Given an input feature vector $\mathbf{x}$, assume we try to assign it to one of the two classes: ω_1 and ω_2, where ω_1 represents the class of a particular human face P and ω_2 represents all other sample range images. From the training samples of P, we can calculate the mean vector μ and the covariance matrix Σ. As shown in Fig. 25, the Mahalanobis distance from a sample vector $\mathbf{x}$ to P can be then expressed as the following

$$D_m^2(P, \mathbf{x}) = (\mathbf{x} - \mu)^t \Sigma^{-1} (\mathbf{x} - \mu). \tag{3.25}$$

Given a threshold h, we can assign a sample $\mathbf{x}$ to ω_1 if its Mahalanobis distance

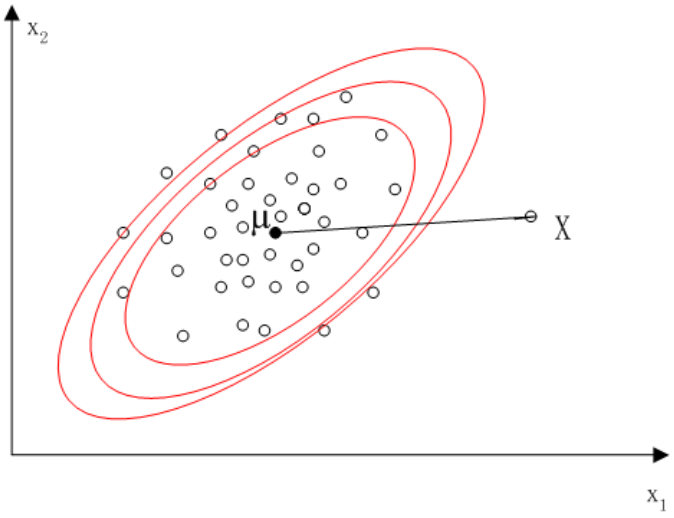


Fig. 25. Mahalanobis distance.

$D_m^2(P, \mathbf{x}) \leq h$; otherwise, $\mathbf{x}$ will be assigned to ω_2.

In the classification based on Mahalanobis distance, given the human face P, accurately estimated covariance matrix and mean are the key factors to differentiate P from other faces. Hence sufficient training samples from P are necessary to be included in the training set. In our scheme, as we will see later, for a human face P, we choose twenty samples to constitute its training set.

F. Experimental Results

1. Data Acquisition

All range images along with corresponding texture images are obtained from a 3D camera system. There are totally 650 range images in our data set containing 113 different human objects. Among these captured range images, about 100 images

Fig. 26. A range (left) image along with the corresponding texture (right) image captured by a 3D camera.

contain faces with facial hair, expressions and holes. Each range image is 750×500 in size. In addition, for a 3D face, the tip of the nose is at the center of the image; and the eyes look straight ahead and they are on a line parallel to the x-axis.

For illustration purposes, a sample range image with its corresponding texture image are given in Fig. 26. It is seen that the hair in the lady's face results in holes in the captured images. Our proposed scheme, as proven by our experiments, will be able to tolerate these artifacts.

2. Accuracy and Robustness of the Optimization

In this section, we introduce the experiments that demonstrate the accuracy and robustness of the optimization procedure in our proposed feature extraction scheme. In the experiments, we first measure the accuracy between the positions of landmark

points in synthesized face and target face. Then the distribution of range differences between the synthesized face and target faces are given. After the optimization, we

Fig. 27. Feature extraction results.

can use the weighting coefficients $\mathbf{W}^*$ to calculate the location of the seventy landmark points $\mathbf{X}_g+\Delta\mathbf{X}_l^*$, $\mathbf{Y}_g+\Delta\mathbf{Y}_l^*$ on the synthesized face. In order to evaluate the accuracy of our feature extraction scheme, we need to compare our results (the location of the landmark points on the synthesized face) with the real location of the landmark points ($\mathbf{X}_t$, $\mathbf{Y}_t$) on the target face T. We find that the landmark points obtained from our scheme are on average within the range of 5 pixels of the corresponding landmark points on the target images. We give some marking results on the tested target faces

shown in Fig. 27 to demonstrate the accuracy of our algorithm. For display purposes, the extracted landmark points are shown on texture images and connected with lines, although our scheme works solely on range images.

The accuracy and robustness of our feature extraction scheme can also be evaluated based on the cost function defined in (3.21). Let the point set Ω be the matching area Ψ defined in Section 2. Assuming there are M points in the matching area, from (3.21) we can obtain the average range difference between the target face T and the synthesized face $S(\mathbf{W})$ as

$$AD(S(\mathbf{W}), T, \Psi) = \frac{1}{M} \sum_{(x,y)\in\Psi} |G(x,y) + \Delta Z(\mathbf{W}, x, y) - T(x + \Delta X(\mathbf{W}, x, y), y + \Delta Y(\mathbf{W}, x, y))| . \quad (3.26)$$

After the matching procedure described in Section 3, based on the optimized weighting coefficients $\mathbf{W}^*$, the synthesized face $S(\mathbf{W}^*)$ can be obtained. The accuracy of the matching procedure can then be evaluated by $AD(S(\mathbf{W}), T, \Psi)$. Clearly, the smaller the average range difference $AD(\mathbf{W}^*, T, \Psi)$ is, the higher accuracy the matching procedure achieve. Based on our database that contains more than 600 range images, we test the matching procedure and give the distribution of the average range difference in Fig. 28. It is seen that after the matching procedure, about 95% resulted average range differences are below 8 range unit, and less than 1% are larger than 9 range unit, which clearly demonstrate the robustness of our optimization procedure.

3. Feature Extraction Schemes for Comparison

In order to demonstrate the effectiveness of the extracted features (weighting coefficients) generated from our proposed scheme, two other different feature extraction

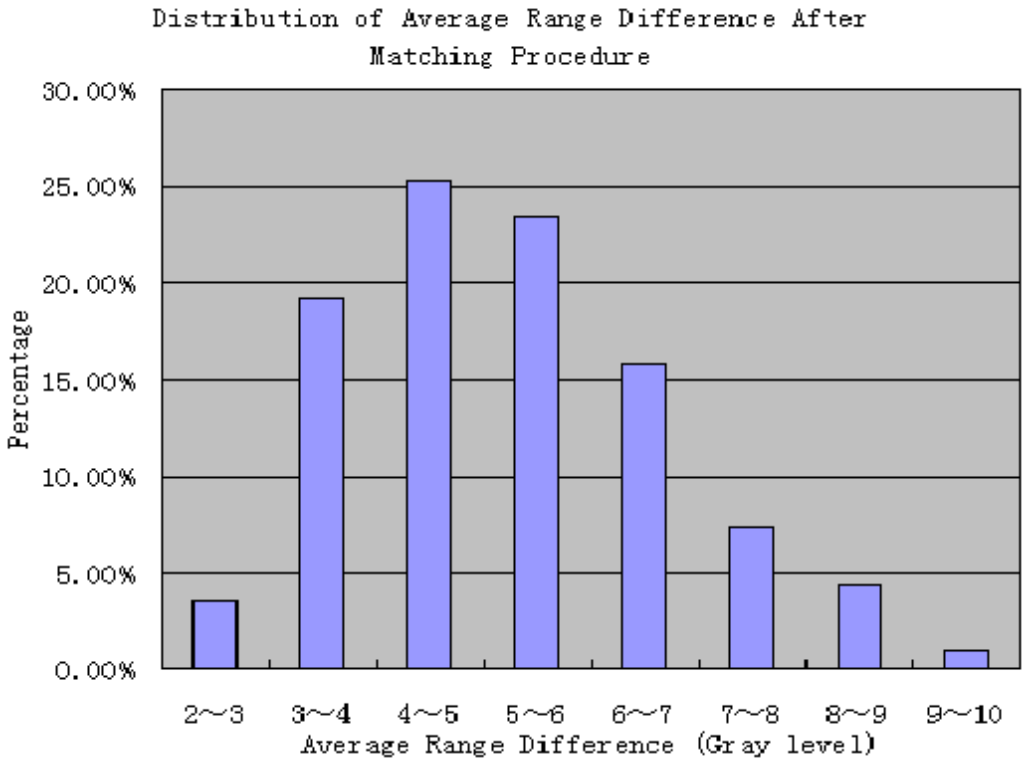


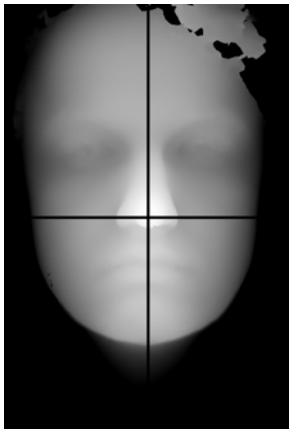
Fig. 28. Distribution of average range difference after matching procedure.

schemes are used for comparison. Like the weighting coefficients extracted from our scheme, the feature sets generated from these two schemes are also classified by the Mahalanobis-distance-based classifier to evaluate the performances.

The first chosen feature set is the face profiles, which can be generated directly from the target image T. As illustrated in Fig. 29, when an input range image has been normalized, the tip of the nose of T will be fixed to the center of the image. Therefore, we can get the vertical and horizontal profiles of T directly following the vertical and horizontal lines that divide the image across the its center. The vertical

and horizontal profiles are concatenated as a feature vector and used for comparison. The reason for choosing face profiles is because they contain some important detailed information that differentiate a person from others. In addition, profiles are relatively insensitive to facial expressions, holes and hair.

In addition to the profiles, we also use the classical eigenface method [1], [32] to extract features for comparison. On each input range image (with size of 750×500), resampling is performed to obtain a downsampled image at 75×50 resolution. All the downsampled images from range images in the database are then treated as a group and eigenvectors can be calculated from PCA. These downsampled images are then projected back to 10 eigenvectors with most significant eigenvalues to generate feature vectors. These feature vectors comprise the second feature set for comparison. The reason for choosing this type of features is that they contain the overall information of 3D human faces.



(a)

(b)

(c)

Fig. 29. Features obtained from the profiles of a target face. (a) The vertical and horizontal lines on a target face for profile extraction. (b) Horizontal profile. (c) Vertical profile.

4. Classification Performance

Our scheme is tested in the access control scenario, where a group of authorized users are allowed to enter a certain protected area, while the imposters are denied from entering. The groups of authorized users and the imposters comprised of 8 and 105 different human subjects, respectively. Each authorized user has 40 range images captured at different times, while each imposter has only a few images in the database. Totally, there are 330 range images from imposters.

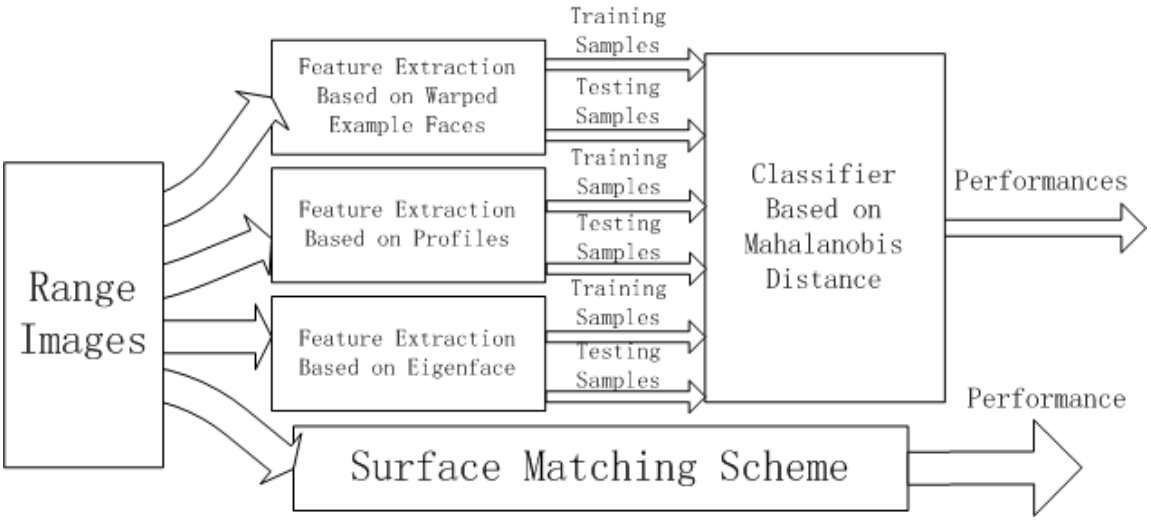


Fig. 30. Experimental settings.

The settings of the experiments are shown in Fig. 30. First, range images in the database are processed by the three different feature extraction schemes. After the feature extraction procedure, three different feature sets are obtained. In the three feature sets, a range image will have three different corresponding feature vectors (samples) respectively. A feature vector in the first feature set is comprised

of weighting coefficients extracted from our proposed scheme based on warped example faces. In the second set, the profiles of 3D faces become feature vectors, as we mentioned in Section 3. The third feature set is obtained from the eigenface method described in Section 3. A feature vector in the feature sets can be considered as a sample of the corresponding range image, and is passed to the Mahalanobis-distance-based classifier. When all the samples in a feature set have been classified, the classification performance of the feature set will be obtained. Based on the classification performances, we can evaluate the three feature sets. In addition to the three feature sets used in classification experiments, for comparison purposes, we also implement the *surface matching* method presented by X. Lu *et al.* in [49] and conduct experiments in the same access control scenario. In *surface matching* method, the Mahalanobis-distance-based classifier is not used. Instead, the surface difference metric [49] is utilized to perform classification.

In the classification experiments based on the three feature sets, samples belonging to each of the eight authorized users are divided into two halves. For each authorized user, one half samples are used for training and the other half are left for testing. The training samples from all the authorized users then constitute the training set. In the training step, for each of the eight authorized users, based on the training samples, we calculate its covariance matrix and mean, which are then used to construct the Mahalanobis-distance-based classifier. On the other hand, the samples from the imposters and the testing halves of the authorized users comprise the testing set. In the testing step, given a sample from testing set, the Mahalanobis distances between the input feature and the eight authorized subjects are calculated. Then based on a Mahalanobis distance threshold, this testing sample will be classified either as an authorized user or an imposter. By changing the Mahalanobis distance threshold, we can obtain a receiver operating characteristic (ROC) curve for each

feature set. To reduce the effect caused by statistical bias, after one classification experiment, the training halves and the testing halves belonging to the eight authorized users are exchanged respectively to perform another classification. The final classification results are the average of the two experiments.

In the experiments based on the *surface matching* method, the training set includes eight range images belonging to the eight different authorized users, respectively. In the matching procedure, given a testing image, it will be matched to all range images in the training set. The matching distance is then used as metric to make classification decision. The authorized user in the range image belonging the training set with closest matching distance to the testing range image will be considered as an identification unless the matching distance is larger than a threshold. Thus by changing the threshold, we can also obtain ROC curve of the *surface matching* method. In the surface matching procedure, given a testing image, about 300 control points around the eye area are chosen.

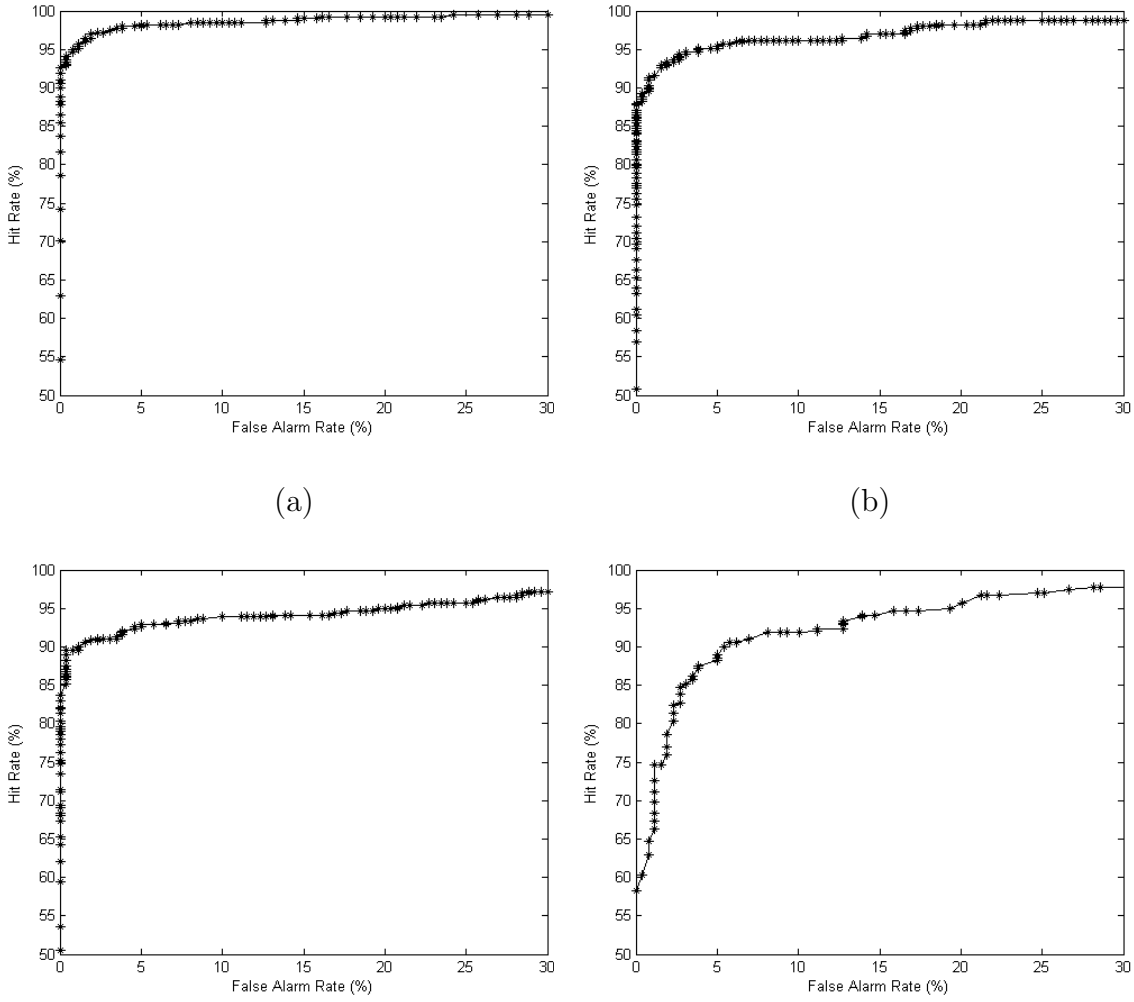


Fig. 31. ROC curves of various features. (a) ROC curve based on the weighting coefficients. (b) ROC curve based on the profiles. (c) ROC curve based on the eigenface method. (d) ROC curve based on the surface matching method.

In order to evaluate the overall performance of our extracted features (weighting coefficients) compared to the other two feature sets (features from profiles and eigenface method) using Mahalanobis-distance-based classifier and the *surface matching* method, we draw the ROC curves using hit rate versus false alarm rate. In Fig. 31 (a), we use the weighting coefficients from our extraction procedure as features; in Fig. 31 (b), the profiles become features; in Fig. 31 (c), features extracted by the eigenface method are used; in Fig. 31 (d), the *surface matching* method is used. In Fig. 31 (a), we can see that the hit rate reaches 97.2% at the false alarm rate 2%; when the false alarm rate reaches 5%, the hit rate climbs to 98.5%. Compared to the performance of weighting coefficients, the features obtained from the profiles are less efficient in differentiating a person's face from others. From Fig. 31 (b), we can see that when the false alarm rate reaches 2%, the hit rate is already 93.4%. When the false alarm rate is 5%, the hit rate reaches is 95.1%. The performance of using features from the eigenface method is the worst among the three sets of features. As shown in Fig. 31 (c), when the false alarm rate reaches 2%, the hit rate is 91.2%. At the 5% false alarm rate, the hit rate is about 92.5%. Thus, from the ROC curves, we can clearly see the advantage of our extracted features. The reason that features from profiles have better performance in classification than features from the eigenface method is because they are less sensitive to the facial expressions, holes and hair.

Compared to the performances of the three feature sets using on the Mahalanobis-distance-based classifier, the *surface matching* method has a lower recognition rate. However, it needs to address that the *surface matching* method uses completely different classification method compared to the classification based on Mahalanobis distance. Unlike the Mahalanobis-distance-based classifier that needs sufficient training data (in our experiments, twenty samples from each authorized users) to calculate covariance matrices and mean, *surface matching* method just needs one range im-

age from each of the authorized users in the training set. Therefore, the comparison in some sense is unfair. Nevertheless, the classification results do demonstrate the effectiveness of the features (weighting coefficients) extracted from our scheme.

Table I. Equal Error Rates for Different Test Schemes.

Test Schemes	EER
Weighting Coefficients	2.5%
Profiles	5.1%
Eigenface method	6.7%
Surface matching	8.6%

At each point on a ROC curve, a pair of miss rate and false alarm rate can be obtained. Considering a certain point on the ROC curve where the miss rate and the false alarm rate are equal, we can obtain the equal error rate (EER) that is defined to have the same value of both miss rate and false alarm rate. Here we list the EERs for the four tested schemes as shown in Table. I

To further evaluate the performance, we study the misclassified target faces when the system operates at the EER point on the ROC curve. For demonstration purposes, four misclassified target faces with corresponding synthesized faces and those human subjects to whom the target faces are mistakenly classified are shown in Fig. 32, respectively. From the misclassified target faces, we can see that there are two reasons that cause the misclassification. First, feature extraction procedure may not be able to generate a synthesized face similar enough to a target face. This will cause the inaccuracy of the extracted features and may result in a misclassification. Second, even with a reasonably similar synthesized face, the extracted feature vector from a target face can still be mistakenly classified when it falls within a range close to

another person's face in terms of Mahalanobis distance.

G. Summary

We have presented a novel feature extraction scheme for 3D face recognition based on warped example faces. We first introduce the face warping procedure based on landmark points defined in generic face. Based on a set of weighting coefficients, a combination of the face warps derived from example faces can be used to warp the generic face as a synthesized range face. After that, the selection of example faces is described. In order to make the synthesized face similar to a target face, Newton's method is used to minimize the range differences by optimizing the weighting coefficients. After the optimization, the optimized weighting coefficients are used as the feature of the target face. A linear classifier based on Mahalanobis distance is used to verify the identity of the target face. We tested our scheme on a database containing more than 600 range faces. Experimental results showed that the feature extracted from our scheme can highly differentiate human faces a high recognition rate. Currently, our 3D face recognition system is implemented in Matlab. On a PC with a 2.4GHz Pentium processor, the scheme takes about 19 seconds to recognize a person from its range image. We expect a significant speedup if the scheme is ported in C with optimization.

Our scheme can automatically extract features from a range image captured from 3D camera system and is insensitive to holes, facial expressions and hair. In our scheme, both detailed and the overall geometric information are utilized to recognized a person.

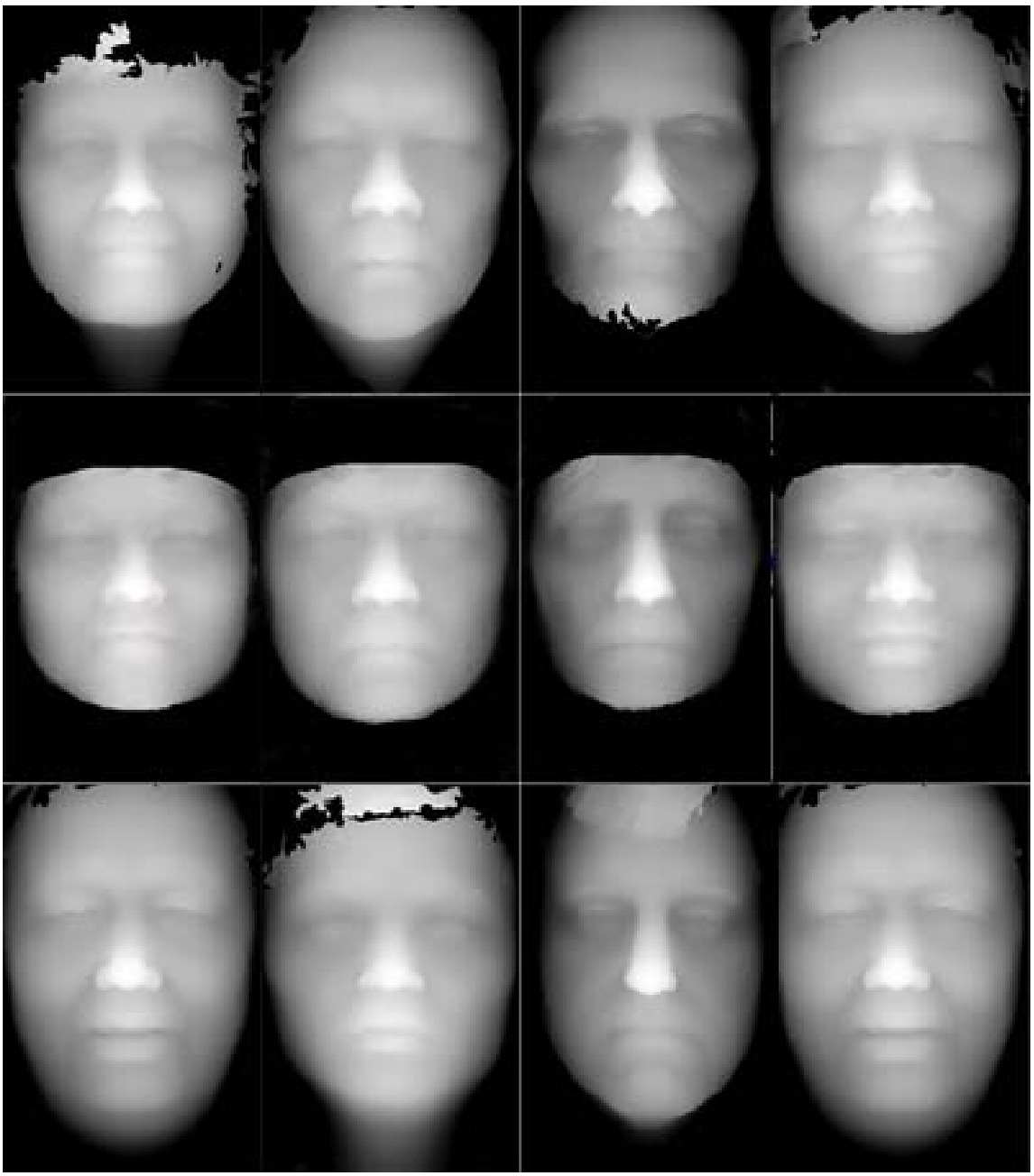


Fig. 32. Misclassified faces (top row) along with corresponding synthesized faces (middle row) and the human subjects to whom the target faces are mistakenly classified (bottom row) when the system operates at the EER point. Note that different classification results can be obtained when the system operates under different configurations.

CHAPTER IV

DISTRIBUTED ROUTING ALGORITHMS FOR THE LOCATION-BASED WIRELESS SENSOR NETWORKS*

In this chapter, we turn to the wireless transportation component of our proposed 3D face recognition system. Specifically, we will focus on the distributed routing algorithms in the location-based wireless sensor networks. Note that here sensor node can be a person, vehicle, etc., rather than simply a conventional concept of being a transducer. Here we start from the so-called dead-end problem [27], and by solving it, we propose two complete routing solutions for location-based wireless sensor networks.

The dead-end problem in greedy forwarding is an important issue of location-based routing in sensor networks. It occurs when a message falls into a local minimum using greedy forwarding. Current solutions to this problem are insufficient in either eliminating traffic/path memorization or finding satisfactory short paths. In this chapter, we propose a novel algorithm, called Partial-partition Avoiding GEographic Routing (PAGER) [74]–[76], to solve the problem. The basic idea of PAGER is to divide a sensor network graph into functional sub-graphs and provide each sensor node with message forwarding directions based on these sub-graphs. PAGER results in loop-free short paths without memorization of traffics/paths in sensor nodes. It does not require planarization of the underlying network graph. Further, the mobility

*© [2005] IEEE. Reprinted, with permission, from "A distributed algorithm for the dead end problem of location based routing in sensor networks" by L. Zou, Z. Xiong, and M. Lu, 2005. *IEEE Transactions on Vehicular Technology*, vol. 54, pp. 1509-1522.

adaptability of PAGER makes it suitable for use in mobile sensor networks with frequent topology changes. We implement the PAGER algorithm in two protocols and evaluate them in sensor networks with different parameters. Experimental results show the advantage of PAGER in the context of sensor networks.

A. Introduction

Sensor networks consist of a large number of tiny sensors with limited power and computational capability. Data collected by the sensors are usually routed to the base station (BS) in a multi-hop manner [2]. A number of routing protocols (e.g., non-location-based [2], [30], [31], [36], [38], [50], [53], [54], [62] and location-based [8], [25], [27], [29], [37], [39], [40], [58]–[60], [69], [70]) have been proposed for sensor networks. Among them, the location-based solutions [8], [25], [27], [29], [37], [39], [40], [58]–[60], [69], [70] have received more attention due to their inherent scalability and power-efficiency. In addition, by providing the location information, location service systems [33], [44] are available to support these location-based routing protocols. However, the dead-end problem is an important issue in location-based routing for sensor networks. It arises when greedy forwarding fails at a sensor node that is closer (in Euclidean distance) to the BS than all its neighbors. Left unsolved, this problem will prevent location-based routing protocols from providing paths from some sensor nodes to the BS (even if such paths might exist). In the past, both memorization-based [37], [60] and stateless [8], [39] recover mode algorithms have been proposed to address the dead-end problem. The memorization-based methods [37], [60] require nodes to memorize their past traffic or paths. These methods are not scalable because they are sensitive to the node queue size, changes in node activity and node mobility. In contrast, stateless recover mode methods [8], [39] do not require a node to memorize its past traffic or paths, hence they maintain almost no state information in sensor

nodes; this property makes the stateless recover mode methods [8], [39] suitable for sensor networks, which usually consist of a large number of sensor nodes. The greedy-face-greedy (GFG)/greedy perimeter stateless routing (GPSR) algorithm [8], [39] is currently the most widely accepted stateless recover mode algorithm in ad-hoc/sensor networks. GFG/GPSR elegantly handles the dead-end problem by routing a message along boundaries of polygons of the planarized network graphs. However, compared to the shortest-path [21] in sparse networks, this algorithm generates long paths with loops, which increase traffic burden and risk of losing data packets in large-scale sensor networks; it additionally requires the underlying network topology to be a planar graph, which increases the computational complexity. In this chapter, we propose a distributed algorithm called Partial-partition Avoiding GEographic Routing (PAGER) that divides a sensor network graph into functional sub-graphs and provides forwarding directions for each sensor node based on these sub-graphs. Distinctive features of PAGER are: 1) it does not require a node to memorize its past traffic or paths or the underlying network graph to be planar, 2) it constructs loop-free paths with lengths close to the shortest, and 3) it is adaptive to mobility, making it applicable to mobile sensor networks with frequent topology changes. We implement PAGER in routing protocols, resulting in PAGER-S and PAGER-M for stationary and mobile scenarios, respectively. The performance of PAGER-S/M is evaluated under different network topologies. Experimental results show the advantages of PAGER-S/M.

B. Sensor Network Model and the Dead-end Problem

1. Network Model

We consider the following network graph model: a number of sensor nodes are randomly deployed on an unobstructed two-dimensional sensing field with a BS. The sensor nodes are modeled by a unit graph. All sensor nodes within communication

range r of a node x are considered as neighbors of x and have bi-directional links with node x, as shown in Fig. 33. $N(x)$ denotes the set of neighbors of node x which includes all sensor nodes that fall into the circle centered at x with radius r. Depending on the applications, the BS is located either adjacent to the sensing field or inside it. It has a communication range R, which is long enough to cover all sensor nodes. Commands from the BS can be sent to sensor nodes via a broadcast channel. In this dissertation we only consider using this broadcast channel to help sensors obtain the location information of the BS. Note that for most sensor network applications, the BSs are normally with little mobility, so that the broadcasting from the BS to sensor is not necessary to be frequent even when the sensor nodes are moving with high frequency. Also, the BS in a sensor network, unlike tiny power-constrained sensor nodes, normally has sufficient power supply. With these two reasons, the cost of obtaining the location information by sensor nodes from the BS is insignificant. Therefore, we assume every sensor node knows the location of the BS, which is actually assumed by all location-based routing protocols [8], [25], [27], [29], [37], [39], [40], [58]–[60], [69], [70]. Consequently we will not consider the unidirectional links from the BS to sensor nodes when modeling the sensor networks. We formally express our sensor network model as a graph $G(V, E, BS)$, where V is the set of sensor nodes and $E \subseteq V' \times V'$ with $V' = V \cup BS$ is the set of bi-directional communication links between pairs of nodes (including the BS) when communication is possible.

2. The Dead-end Problem in Sensor Networks

Information collected at a sensor node is forwarded to the BS in a multi-hop manner with possible data fusion/aggregation [30], [31]. In location-based routing methods [8], [25], [27], [29], [37], [39], [40], [59], [60], [69], [70], a forwarding node utilizes the location information of the destination, its neighbors and itself to route a message.

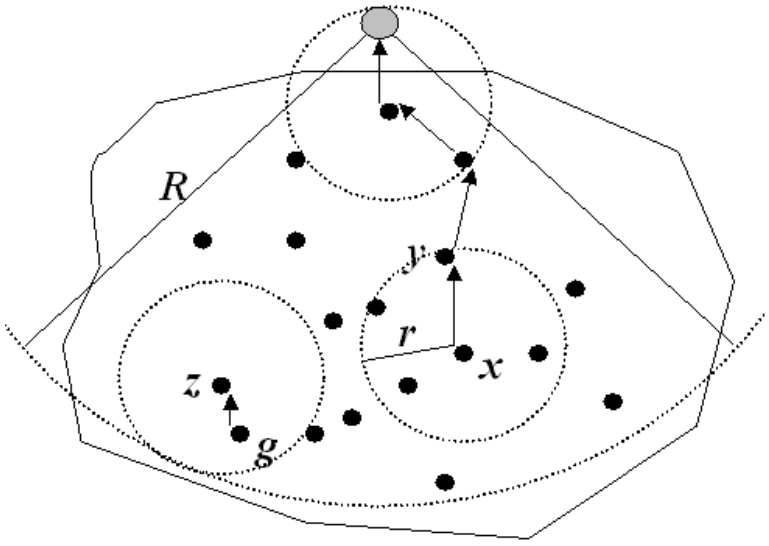


Fig. 33. The sensor network model.

The most popular forwarding method in this category is greedy forwarding, in which a node forwards the message to a neighbor that is closer (in Euclidean distance) to the destination. Greedy forwarding is shown in [39] to have path length close to the shortest in dense networks. Under greedy forwarding, node x in Fig. 33 will forward the message to its neighbor y because y is closer to the BS. The so-called dead-end problem makes greedy forwarding insufficient in sparse (or even dense) sensor networks. This problem arises when a message is forwarded to a local minimum, i.e., a node with no neighbor of closer distance to the destination. In Fig. 33, after node z receives a message from node g, it finds no closer neighbor to the BS in its communication range to forward the message (even though there exists a path to the BS). Thus node z is a local minimum. In our sensor network model, a local minimum is referred to as a concave node, which we formally define as follows.

Definition 1 (Concave Node)**:** A sensor node x in a sensor network graph $G(V, E, BS)$ is a concave node if $distance(x, BS) < distance(y, BS), \forall y \in N(x)$, where $distance(.,.)$ represents the Euclidean distance.

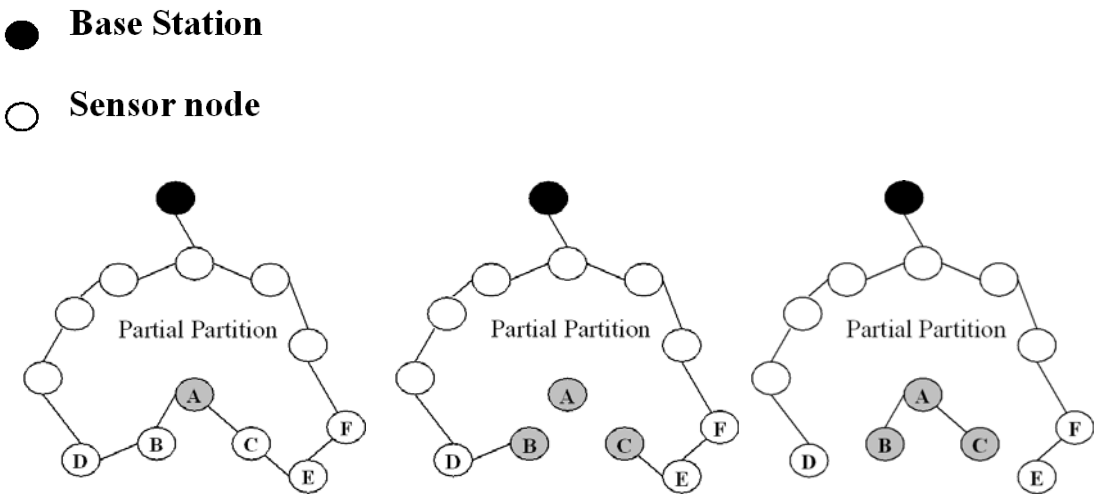
The objective of this chapter is to solve the dead-end problem in the context of sensor networks, which is different from those of [8], [37], [39], [60] that deal with ad-hoc networks with symmetric topology and no BS. We show that our proposed PAGER algorithm solves the dead-end problem due to concave nodes.

C. The PAGER Algorithm

PAGER uses two phases to solve the dead-end problem in sensor networks. The first $shadow - spread$ phase divides a connected graph into sub-graphs originated from concave nodes; the second $cost - spread$ phase establishes paths on a given sub-graph obtained in the first phase.

1. The Shadow-spread Phase

The aim of the $shadow - spread$ phase is to locate the "dangerous nodes" close to concave nodes, where a message will inevitably be forwarded to a concave node by greedy forwarding. As an example, consider the process of $shadow - spread$ on graphs shown in Fig. 34. In Fig. 34(a), node A is a concave node while nodes B and C are not. If we disconnect A from the graph (shown in Fig. 34(b)), B and C become new concave nodes. Thus we go one step further by disconnecting B and C from the original graph. As Fig. 34(c) shows, by disconnecting the sub-graph consisting of A, B and C from the original graph, we obtain a graph with no concave node. Therefore, using greedy forwarding, messages originated from any node in the new graph will be forwarded to the BS without encountering the dead-end problem.



(a) (b) (c)

Fig. 34. An example of the shadow-spread phase. (a) The original network graph contains a concave node A. (b) After disconnecting A from graph, new concave nodes B and C appear. (c) After excluding the sub-graph consisting of A, B and C, the remaining graph contains no concave node.

In Fig. 34, we mark A, B and C in the network graph as shadow nodes and the remaining sensor nodes as bright nodes. In addition, we will call the bright nodes adjacent to shadow nodes (i.e., nodes D and E) border nodes. Below we formally define shadow nodes, bright nodes and border nodes.

Definition 2 (Shadow Node)**:** A sensor node x in a sensor network graph $G(V, E, BS)$ is a *shadow node* if $distance(x, BS) < distance(y, BS)$ for any y in $N(x)$, excluding the subset of $N(x)$ that are already shadow nodes.

We denote the latter set as $Shadow(N(x))$. Note that according to Definition 2, concave nodes are a subset of shadow nodes.

Definition 3 (Bright Node)**:** A sensor node x in a sensor network graph $G(V, E, BS)$ is a *bright node* if there exists a node $y \in N(x)$ - $Shadow(N(x))$ such that $distance(x, BS)$ $> distance(y, BS)$.

Definition 4 (Border Node)**:** Given a sensor network graph $G(V, E, BS)$, a bright node x is a *border node* if $Shadow(N(x)) \neq \emptyset$.

Differentiating the nodes on the network graph based on their statuses (e.g., shadow or bright), we can thus divide the original graph into two sub-graphs. We call the sub-graph that contains all shadow nodes the *shadow area*. Similarly, the sub-graph that contains all bright nodes is called the *bright area*. Furthermore, we call the area encompassing node A in Fig. 34 a partial partition, which partially partitions the sensor network in Fig. 34 and creates the concave node A. Formal definitions of the shadow area, bright area and partial partition are given below.

Definition 5 (Shadow Area)**:** Given a sensor network graph $G(V, E, BS)$, a *shadow area* is a sub-graphs $G_s(Vs, Es)$ of $G(V, E, BS)$, where V_s is the vertex set of all

shadow nodes that are connected with each other and $E_s \subseteq V_s \times V_s$ is the edge set that consists of communication links between any pair of shadow nodes in V_s.

Definition 6 (Bright Area)**:** Given a sensor network graph $G(V, E, BS)$, a *bright area* is the sub-graph of $G(V, E, BS)$ that excludes all shadow areas.

Definition 7 (Partial Partition)**:** *Partial partition* is a region on a graph bounded by edges of sensor nodes and/or boundaries of the sensing field that encompasses at least one concave node.

In order to divide the original graph into a bright area and shadow areas, nodes should be able to exchange information of their status (shadow or bright) along with their location information. This information exchange is realized by periodically broadcasting beacon messages $beacon(status, location)$ that contain two fields: status and location. Every node on a graph should be able to decipher the location information of the BS. Based on our network model, sensor nodes obtain the BS's location from its broadcast channel. The $shadow - spread$ phase is given in Algorithm 1 for a forwarding sensor node x, where the auxiliary function $DecideShadow$ shown in Algorithm 2 is used to determine whether x is a shadow node. The location information of x (denoted as $location(x)$), the location information of the base station(denoted as $location(BS)$), the beacon message received by x (denoted as $beacon(x)$) and the neighbor list of x are the input parameters of algorithm 1 and 2.

Given a sensor network topology/graph, the $shadow - spread$ phase needs a convergence period to stabilize. Here we give its convergence time (in terms of the beacon broadcast interval B) in Proposition 1.

Proposition 1 (Convergence Time of Shadow-spread)**:** Given a sensor network graph $G(V, E, BS)$ that consists of n sensor nodes, the $shadow - spread$ phase converges in n rounds in terms of beacon broadcast interval B.

Algorithm 1 $ShadowSpread(location(x), location(BS), beacon(x))$

1: $status(x) = bright$;
2: **while** active **do**
3: **if** receive beacon from a neighbor y **then**
4: Refresh the neighbor set $N(x)$ by updating the status and location information of y;
5: **if** $DecideShadow(location(x), location(BS), N(x))$ **then**
6: $status(x) = shadow$;
7: **else**
8: $status(x) = bright$;
9: **end if**
10: **end if**
11: **if** beacon timeout occurs **then**
12: Copy status and location information of node x to beacon(status, location) and send out to all neighbors;
13: **end if**
14: **end while**

Algorithm 2 $BoolDecideShadow(location(x), location(BS), N(x))$

1: $shadow$ = true;
2: **for** each node $z \in N(x)$ **do**
3: **if** $((status(z) = bright)\&\&(distance(z) < distance(x))$ **then**
4: return false;
5: **end if**
6: **end for**
7: return $shadow$;

Proof. We know that a shadow node will not change its status to bright as long as the given graph $G(V, E, BS)$ remains static. We only need to show that shadow nodes appear continuously in the $shadow-spread$ phase. That is, the time interval between the appearances of two shadow nodes is no longer than B. By way of contradiction, assume that during the period $[t_0, t_0 + B)$, there is no new shadow node appearing in $G(V, E, BS)$; while during $[t_0 - B, t_0)$ and $[t_0 + B, t_0 + 2B)$ some shadow nodes change their status to bright. Denote the first new shadow node appears during $[t_0 + B, t_0 + 2B)$ as v, then according to Algorithm 1, there are two possibilities for v: it is a concave node or at least one of its neighbor changes status from bright to shadow. It is clear that v is not a concave node, because otherwise it would have appeared before the $[t_0, t_0 + B)$ period. So at least one of v's neighbors have changed statuses from bright to shadow during $[t_0, t_0 + B)$. This contradicts with our assumption. Consequently, the convergence time of the $shadow - spread$ phase is at most n rounds. □

We give the following two Lemmas before presenting Proposition 2 on the fatality of shadow areas.

Lemma 1 (One Direction of Greedy Forwarding)**:** Given a sensor network graph $G(V, E, BS)$, after the $shadow - spread$ phase converges, shadow nodes will not forward packets to bright nodes using greedy forwarding.

Proof. After the $shadow - spread$ phase converges, all the shadow nodes are known. Consider the shadow nodes that have at least one bright node as neighbor. If a shadow node x can forward a message to its neighboring bright node y, then y must be closer to the BS than x, contradicting the assumption that x is a shadow node. □

Lemma 2 (Two Endings of a Path)**:** Given a sensor network graph $G(V, E, BS)$, a message will end up with either a concave node or the BS using greedy forwarding.

Proof. In [60] Stojmenovic and Lin proved that following greedy forwarding, paths are loop-free. So a path constructed by greedy forwarding will end up with a certain node. In the connected sensor network graph after the $shadow - spread$ phase converges, according to the definition of concave nodes, a node other than concave node or the BS will always find a neighbor to forward a message using greedy forwarding. So the only node that can be an end of a path constructed by greedy forwarding is either the BS or a concave node. □

Proposition 2 (Fatality of Shadow Areas)**:** Given a sensor network graph $G(V, E, BS)$, after the $shadow-spread$ phase converges, a message forwarded to a shadow area will end up with a concave node using greedy forwarding; on the other hand, a messages forwarded to the bright area will end up with the BS using greedy forwarding if all shadow areas are disconnected from $G(V, E, BS)$.

Proof. From Lemma 1, we know that a message x forwarded to a shadow area cannot find a path across border nodes to the bright area using greedy forwarding. Lemma 2 stipulates that a message x will reach either a concave node or the BS using greedy forwarding. The only way for the message x forwarded to a shadow area $G_s(V_s, E_s)$ to end up with the BS is when the BS is adjacent to $G_s(V_s, E_s)$. But this violates the definition of shadow nodes. So we conclude that any message forwarded to a shadow area will end up with a concave node using greedy forwarding. On the other hand, if we disconnect all the shadow areas from $G(V, E, BS)$, we will find that there exist no concave nodes in the resulted graph. Based on Lemma 2, we can conclude that every messages originated from the resulted graph will reach the base station. □

Proposition 2 means that a shadow node will always send forwarding messages to a shadow area using greedy forwarding. For example, node B in Fig. 34(a) will always forward a message to node A. Therefore a message forwarded to a shadow node will

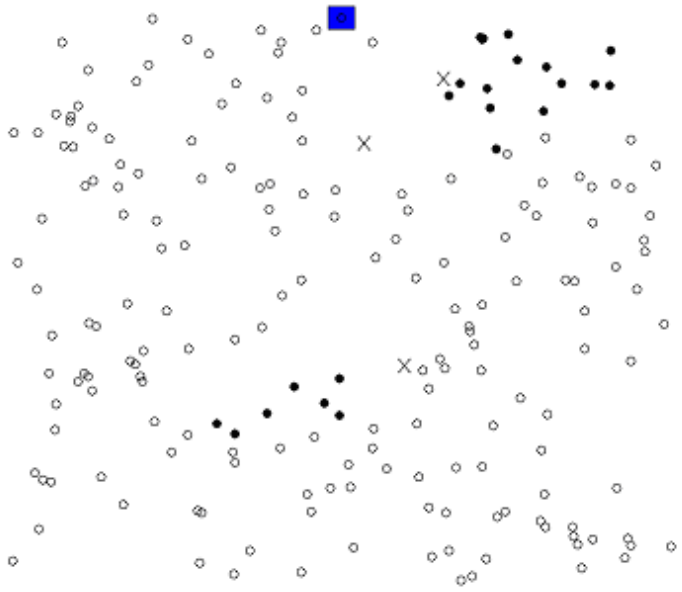


Fig. 35. Distribution of shadow/bright areas in a sensor network with 200 randomly placed sensor nodes. The BS is represented by a rectangle. Concave nodes are marked by "X". Black nodes represent shadow nodes whereas white nodes signify bright nodes.

eventually be trapped in a concave node using greedy forwarding. On the other hand, messages from bright nodes will be safely forwarded to the BS without encountering dead-ends if we disconnect all shadow areas from $G(V, E, BS)$. For example, node E will forward a message to the BS via node F after C is disconnected from the graph (see Fig. 34(c)); otherwise, all messages from E will be forwarded to C and lost in concave node A. The distribution of shadow/bright areas of a sensor network depends on its topology and the communication range of sensor nodes. In Fig. 35 we give an example distribution of *shadow/bright areas* on a 200-node sensor network. Sensor nodes are randomly deployed on a $1000m \times 1000m$ sensing field with the BS

located at $(500, 1000)$. Each node has a communication range of $113m$, which creates a network topology with average degree of 8. As shown later in our experiments, for a fixed average degree, the more sensors; the bigger shadow areas (with more shadow nodes). Similarly, for a fixed number of sensor nodes, a denser sensor network (with higher average degree) will have smaller shadow areas.

2. The Cost-spread Phase

Once the sub-graphs are obtained from first $shadow - spread$ phase, the second $cost - spread$ phase begins. The objective is to give each shadow node a forwarding direction to the nearby bright area. To illustrate how the second phase works, we show in Fig. 36 an example that starts with the same network graph as in Fig. 34. Initially each node on the graph is marked with a cost that is its Euclidian distance to the BS (see Fig. 36(a)). In the first step, every shadow node tries to avoid being surrounded by neighbors with higher costs. As Fig. 36(b) shows, node A finds that all its neighbors have higher cost. To avoid this situation, node A increases its cost to 22 to be higher than the maximum cost of its neighbors by Δ (Δ is the incremental step size, which is set to 3 in this example). In the second step, both nodes B and C find that all their neighbors have higher costs (since node A's cost is now 22). In response, they increase their costs to 25 and 26, respectively, as shown in Fig. 36(c), by following the same rule. The $cost - spread$ phase ends in the third step when node A increases its variable from 22 to 29, as shown in Fig. 36(d). Now, each of the shadow nodes A, B and C has at least one neighbor with smaller cost. If we connect each node with its lowest-cost neighbor, we establish cost gradients across the whole graph (see Fig. 36(e)). Then messages forwarded from sensor nodes to the BS will follow the paths of these gradients.

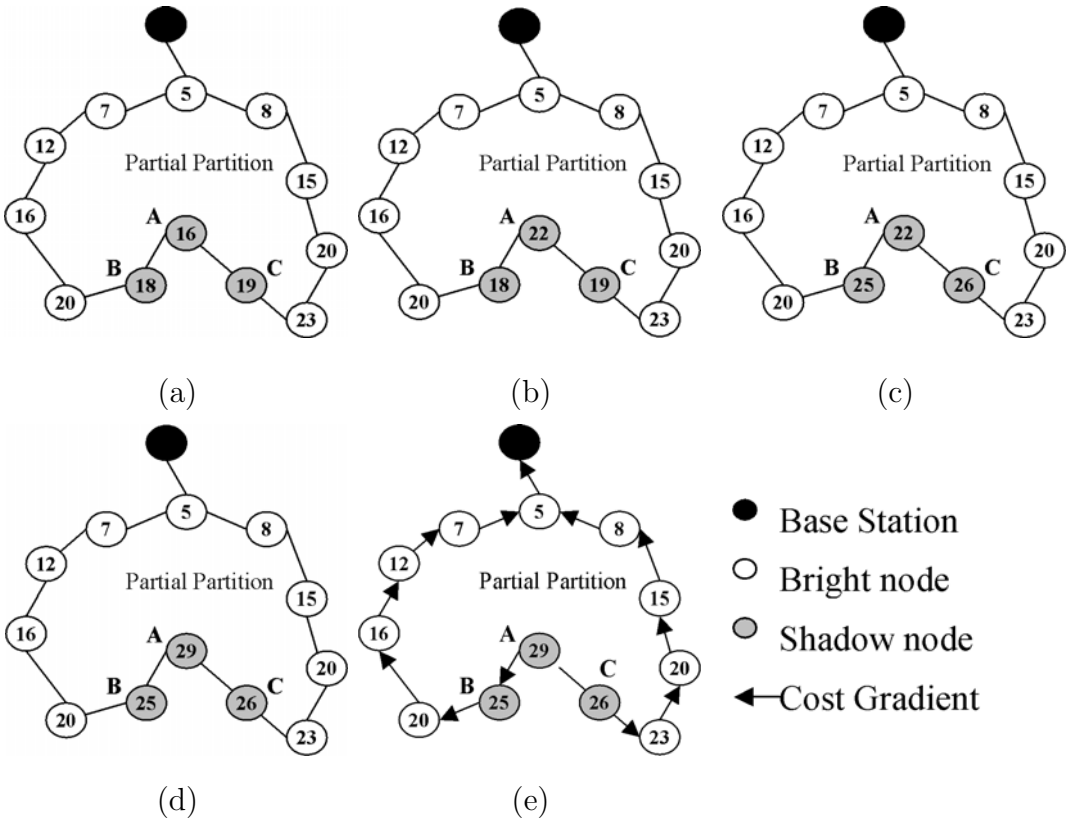


Fig. 36. An example of the cost-spread phase. (a) Before the current phase, shadow nodes (A, B, C) are identified by the shadow-spread phase. (b) Step 1: Node A increases its cost to 22=max(18,19)+Δ. (c) Step 2: Nodes B and C increase their costs to max(22,20)+Δ=25 and max(22,23)+Δ=26, respectively. (d) Step 3: Node A further increases its cost to max(25,26)+Δ=29. (e) The cost-spread phase ends when each shadow node has at least one neighbor with smaller cost. A cost gradient is established from each node to its lowest-cost neighbor.

An important parameter in the $cost-spread$ phase is the incremental step size Δ. In practice, Δ should be set to be the average Euclidean distance between neighboring sensor nodes in a sensor network graph $G(V, E, BS)$. The rationale behind this is that, if node y is the closest (in terms of number of hops) bright node to shadow node x, then the cost of node x with this choice of Δ will be approximately the desired $distance(x, y) + distance(y, BS)$. Note that the $cost - spread$ phase depends on the results of the $shadow - spread$ phase. This is because $cost - spread$ phase applies only to shadow areas instead of the whole network, while the shadow areas should be generated by the $cost - spread$ phase. In practice, however, the first phase and the second phase can be combined together. Once the status of a sensor node turns from bright to shadow, the $cost - spread$ phase can be triggered in that node immediately. In addition, cost information must be exchanged among sensor nodes to enable $cost - spread$. Similar to the way status information is handled in the $shadow - spread$ phase, we add the cost information to the periodically broadcasted beacon messages. We summarize the $cost - spread$ phase in Algorithm 3. The convergence time (again in term of the beacon broadcast interval B) of the $cost - spread$ phase depends on the size of (or the number of nodes in) the shadow areas in a given sensor network graph $G(V, E, BS)$. Before we formally present this result in Proposition 1, we introduce three lemmas under the assumption that the $cost - spread$ phase starts after the $shadow - spread$ phase converges.

Lemma 3 (Adjacent Areas)**:** Given a connected sensor network graph $G(V, E, BS)$, if shadow areas exist, they are adjacent to the bright area.

Proof. If there is no shadow area, the lemma is true. Consider the scenario with at least one shadow area. To have a shadow area not adjacent to the bright area while remaining connected to other nodes in the graph, the shadow area must be adjacent

Algorithm 3 $CostSpread(nodex)$

```
1: cost(x) = distance(x, BS);
2: while active do
3:    if receive beacon(cost) from a neighbor y then
4:       Refresh the neighbor set N(x) by updating the cost of y;
5:       if cost(x) ≤ minimum cost of neighbor set N(x) then
6:          cost(x) = maximum cost of neighbors + Δ;
7:       end if
8:    end if
9:    if beacon timeout occurs then
10:      Copy cost of node x to beacon(cost) and send out to all neighbors;
11:   end if
12: end while
```

to the BS. According to definition of shadow nodes and shadow areas, we know that this scenario is impossible. Hence the Lemma is proved. □

Lemma 4 (Convergence and Upper Bound of Cost)**:** Given a connected sensor network graph $G(V, E, BS)$ with shadow area $G_s(V_s, E_s)$, the $cost - spread$ phase converges. At convergence, shadow nodes in $G_s(V_s, E_s)$ have costs upper-bounded by $max\_distance(B_d(G_s), BS)+m*\Delta$, where $B_d(G_s)$ is the set of all border nodes in the shadow area G_s and m is the number of shadow nodes in V_s.

Proof. First, given sensor network graph $G(V, E, BS)$, from the $cost - spread$ algorithm, we know that the costs maintained in every node will not decrease.

Base on Lemma 3, we know that there exist shadow nodes that have neighbors with bright status. Let node x be a shadow node belongs to V_s that has at least one neighbor with bright status, that is, $bright(N(x)) \neq \emptyset$.

Let $max\_cost(bright(N(x)))$ be the maximum cost of the x's neighbors with bright status. Since the costs of bright nodes are not affected by the $cost - spread$ algorithm, we have:

$$max\_cost(bright(N(x))) = max\_distance(bright(N(x)), BS), \tag{4.1}$$

where $max\_distance(bright(N(x)), BS)$ represents the maximum Euclidean distance of x's neighbors with bright status to the BS. It is clear that the cost of x has an upper bound according to the $cost - spread$ algorithm:

$$cost(x) \leq max\_cost(bright(N(x))) + \Delta. \tag{4.2}$$

Combining Equations 4.1 and 4.2, we have:

$$cost(x) \leq max\_distance(bright(N(x)), BS) + \Delta, \forall x \in V_s, bright(N(x)) \neq 0. \quad (4.3)$$

Let V_s^1 be the set of all shadow nodes in V_s with at least one neighbor with bright status. Then we have $B_d(G_s) = bright(N(V_s^1))$ and Equation 4.3 can be rewritten as

$$max\_cost(V_s^1) \leq max\_distance(B_d(G_s), BS) + \Delta. \quad (4.4)$$

Similarly, let V_s^2 be the set of all shadow nodes in V_s that have at least one neighbor belonging to V_s^1; V_s^3 be the set of all shadow nodes in V_s that have at least one neighbor belonging V_s^2, etc. If we keep on dividing the shadow nodes in V_s in this manner, the process will stop in at most m sets: $V_s^1, \ldots, V_s^m$. Since

$max\_cost(V_s^1) \leq max\_distance(B_d(G_s), BS) + \Delta$,

$max\_cost(V_s^2) \leq max\_cost(V_s^1) + \Delta \leq max\_distance(B_d(G_s), BS) + 2 * \Delta$,

...

$max\_cost(V_s^m) \leq max\_distance(B_d(G_s), BS) + m * \Delta$.

This completes the proof. □

Lemma 5 (Minimum Cost Increase)**:** Given a connected sensor network graph $G(V, E, BS)$ and a shadow area $G_s(V_s, E_s)$ that contains m shadow nodes. When running the $cost - spread$ algorithm, during any interval $[t, t + B]$, the minimum cost of the shadow nodes in $G_s(V_s, E_s)$ will increase at least Δ until it is larger than $min\_distance(B_d(G_s))$. All shadow nodes will stop increasing their costs after time $t + m * B$.

Proof. Given a shadow area G_s, at the beginning of $cost - spread$ algorithm, all shadow nodes have costs equal to their Euclidean distances to the BS. Hence the minimum cost of shadow nodes in G_s is less than the minimum Euclidean distance

of the border nodes of G_s. This can be verified from the definition of shadow nodes. After running of the $cost-spread$ algorithm, let t' be the moment when the minimum cost of shadow nodes in G_s becomes larger than the minimum Euclidean distance of the border nodes of G_s to the BS. We consider any interval $[t, t+B)$ with $t+B < t'$. In G_s, let node x be a shadow node having the minimum cost at moment t and let $min\_distance(B_d(G_s))$ be the minimum Euclidean distance of the border nodes of G_s to the BS. Since $cost(x) < min\_distance(B_d(G_s))$, we conclude that at moment t, $cost(x) < min\_cost(N(x))$. According to the $cost - spread$ algorithm, x will increase its cost at least Δ during $[t, t+B)$. This proves the first part of the Lemma. Let t" be the moment when the $cost - spread$ algorithm converges in the shadow area G_s. Let t_1 be the moment that the minimum cost of shadow nodes in G_s passes the minimum Euclidean distance of border nodes of G_s to the BS. We consider the interval $[t_1, t_1+B]$ first. Let Φ be the set consisting of the border nodes with minimum Euclidean distance to the BS. Let U_s^1 be the set of all shadow nodes in G_s that have at least one neighbor in F. We pick any shadow node x in U_s^1. Since $cost(x) > min\_distance(B_d(G_s))$, x will not increase its cost any more after time t_1, because it has found a neighbor with a smaller cost. Similarly, let U_s^2 be the set of all shadow nodes in V_s that have at least one neighbor belong to U_s^1. Then after time $t_1 + B$, the minimum cost of all nodes belonging to U_s^2 will stop increasing. Similarly, let $t_2 = t_1 + B$, and let U_s^3 be the set of all shadow nodes in V_s that have at least one neighbor belong to U_s^2. After $t_2 + B$, the minimum cost of all nodes in U_s^3 will stop increasing. By keep on doing this until t", we conclude that before the $cost - spread$ algorithm converges, at least one node stop increasing its cost in every B period. Since there are m shadow nodes in G_s, this process stop at most in $t_1 + m * B$. □

Proposition 3 (Convergence Length of the Cost-spread Algorithm)**:** Given a con-

nected sensor network graph $G(V, E, BS)$, the convergence time (in term of beacon broadcast interval B) of the $cost - spread$ algorithm is linear with respect to the maximum size (number of nodes) of shadow areas in G.

Proof. Before the running of the $cost - spread$ algorithm, the minimum cost in a shadow area G_s with m nodes will be less than $max\_distance(B_d(G_s)) - m * r$, where r is the communication range of sensor nodes. We can conclude directly from Lemmas 4 and 5 that it will take $O(m)$ rounds (in term of beacon broadcast period B) for nodes in G_s to stop increasing their cost. In a sensor network, there may exist many separate shadow areas. Since the convergence processes of the $cost - spread$ algorithm in different separate shadow areas are performed in a parallel way, the convergence length of the $cost - spread$ algorithm in the whole sensor network is only associated with the maximum size (number of nodes) of these shadow areas. □

After cost gradients are established, we have the following observations from the example shown in Fig. 36: 1) A message originating from the bright area will be eventually forwarded to the destination by following the cost gradients without entering a shadow area. 2) A message originating from a shadow area will be led to a bright area and eventually reach the destination via bright nodes following the cost gradients. 3) A message originating from any node on the graph will not experience the same node twice following the cost gradients. Based on the above observations, we have Proposition 4. But before introducing it, we need Lemma 6:

Lemma 6 (Unidirectional Cost Gradients on Border)**:** Given a connected sensor network graph $G(V, E, BS)$, after the convergence of the $cost - spread$ algorithm, bright nodes will not forward massages to shadow nodes following cost gradients.

Proof. After the $cost - spread$ algorithm converges, in a shadow area G_s of the connected sensor network graph G, let us consider a shadow node x that has at

least one neighbor with bright status. The neighbors of x can be classified into $bright(N(x))$ and $shadow(N(x))$ based on their status. If following cost gradients a neighbor y with bright status can forwarded a message to x, the cost of y will be larger than that of x, i.e., $cost(y) > cost(x)$. Since $distance(x, BS) \leq cost(x)$, we get $distance(x, BS) < cost(y) = distance(y, BS)$. From the definition of shadow node, we know this is impossible. This ends the proof. □

Proposition 4 (Delivery Guarantee and Loop-free)**:** Given a connected sensor network graph $G(V, E, BS)$, after the convergence of the $cost - spread$ algorithm, following the cost gradients, all paths originating from different sensor nodes will end up with the BS and are loop-free.

Proof. We first prove that the paths constructed following the cost gradients are loop-free. For messages forwarded to any bright area, this is true since for bright nodes, forwarding messages following the cost gradients is equivalent to greedy forwarding, which has been proven to be loop-free in [60]. For messages forwarded to shadow areas, by way of contradiction, we assume that there is a loop from node x to y to z. Then we can conclude $cost(y) > cost(x)$. This is impossible because otherwise x will not forward the message to y in the first place. So there exist no loops in shadow areas and in bright areas following the cost gradients. Further, based on Lemma 6, once a message has been forwarded from a shadow area to a bright area, it will not enter any shadow area again. So we conclude that there exist no loop following the cost gradients in a given connected sensor network graph G. To prove the other part, we know from Lemma 6 that following the cost gradients, bright nodes will not forward messages to shadow nodes. According to Proposition 2, every bright node will find a path to the BS based on cost gradient (for bright nodes, forwarding a message following cost gradients equals greedy forwarding). On the other hand, after

the convergence of the $cost - spread$ algorithm, every shadow node in the shadow area will at least find a neighbor with lower cost. So for a message being forwarded to a shadow area, it will be stuck in shadow nodes. From the first part of the statement we know that the message will not experience the same node twice in a shadow area. So for a message being forwarded to a shadow area, it will be forwarded to a bright node. Once it enters a bright node, based on Proposition 2, it will reach the BS. This ends the proof. □

Proposition 4 shows that by combing the $shadow - spread$ and $cost - spread$ algorithms together, our distributed scheme has the loop-free delivery-guaranteed property. The main idea behind our scheme is to avoid messages entering "dangerous areas" and leading messages originating from these "dangerous areas" to "safe areas". According to Proposition 2, these "dangerous areas" and "safe areas" are actually shadow areas and bright areas, which are caused by $partial - partition$ defined previously. This is why we call our scheme Partial-partition Avoiding GEographic Routing (PAGER).

3. Extension of PAGER to Large-Scale Sensor Networks with Multiple BS

To manage large sensing fields, multiple BS may be necessary. In these cases, we can extend PAGER by dividing a large sensing field into multiple small cells as shown in Fig. 37. Each BS takes care of one cell and uses a broadcast channel to inform sensors in the cell of its location information and commands. In this way, shadow/bright areas and cost gradients can be established within cells corresponding to different BS. Communication between BS can be realized via a wireless or wired medium.

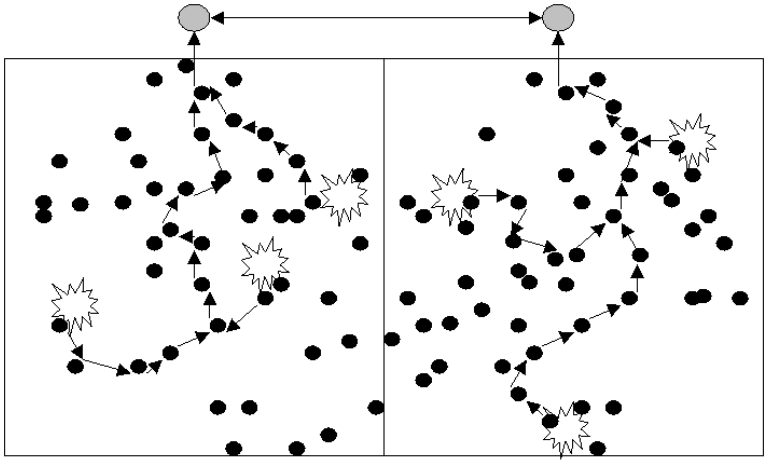


Fig. 37. Information collected by sensors is forwarded back to multiple base stations in large-scale sensor networks.

4. Impact of Passive Mode

In order to save power, sensors may randomly go from active mode to passive mode, which actually decreases the average degree of a given sensor network. As mentioned in Section 1, the number of dead-ends and the corresponding shadow areas

may increase when the average degree of the network decreases. In this case, the convergence time and path length in the PAGER algorithm will increase. However, as long as the resulting network topology composed of active sensors is connected, there exists a forwarding path for every active sensor to the BS, PAGER will converge to stabilization.

D. Mobile Sensor Networks

In some application scenarios, sensor nodes move frequently either because they have motion capacity or because of the instability of the sensing environment. These scenarios are different from the stationary sensor networks in many ways. We extend PAGER to a protocol called PAGER-M to handle these scenarios, where "M" stands for "mobility". In this section we first introduce mobility adaptability to PAGER, then explain certain features of the PAGER-M protocol, including the trade-off between the beacon broadcast interval and the utilization of path redundancy.

1. Mobility Adaptability of Shadow Areas and Cost

In mobile sensor networks that contain frequently moving sensor nodes, shadow/bright areas can appear/disappear due to sensor node mobility. Along the appearance or disappearance of shadow areas, the costs of sensor nodes also change. For the purpose of presentation, we give two examples with figures. As Fig. 38(a) shows, initially node B is within the communication range of node A although these two nodes are moving apart; node A is not a concave node since its neighbor node B has a closer distance to the BS. After a period of time, as shown in Fig. 38(b), when node B has moved out of the communication range of node A, node B is no longer a neighbor of node A. Without node B as a neighbor, node A becomes a concave node. Then the

$shadow - spread$ and $cost - spread$ processes begin. These processes finally result in shadow nodes (A, C and D) with different costs as shown in Fig. 38(b).

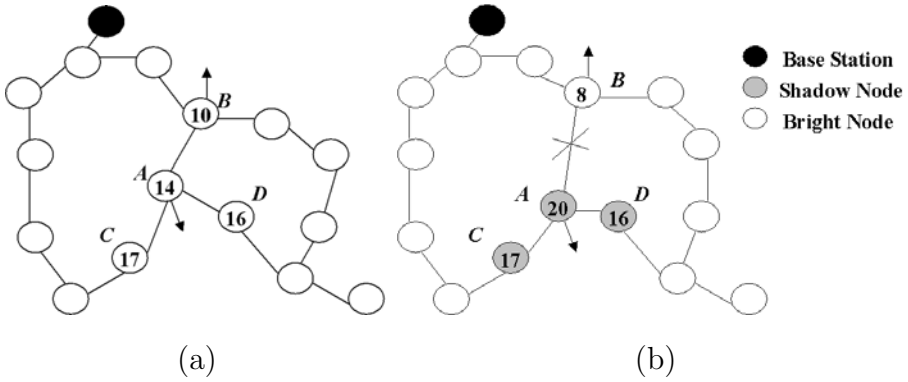


Fig. 38. Appearance of a shadow area. (a) No concave node and shadow area exist when node A and node B are within each other's communication range. (b) Link between node A and node B is broken due to movement, a new concave node and shadow area appears.

On the other hand, node mobility can also cause the disappearance of concave nodes and shadow areas. To explain this process, we consider the graph shown in Fig. 39. As Fig. 39(a) shows, initially nodes A and B are beyond each other's communication range. Node A is hence a concave node because it is the closest node to the BS within its local topology. After a period of time as Fig. 39(b) shows, when nodes A and B are moving closer, they fall into each other's communication range. Then node A finds out that it is no longer a concave node because node B has a closer distance to the BS. After that, node A first resets its cost to the Euclidean distance to the BS and changes its status to "bright". Then, it sends out beacon messages containing its new status (bright) to all its neighbors. As Fig. 39(b) shows, this finally results in a graph without a shadow node. Because shadow areas can adaptively appear/disappear with

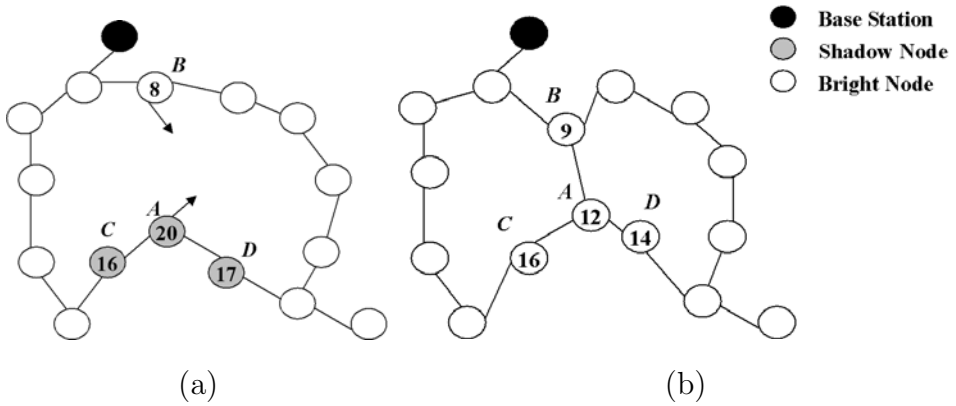


Fig. 39. Disappearance of a shadow area. (a) Node A and node B are moving approach to each other. (b) After the link established between node A and node B, node A is no longer a concave node.

the topology changes, PAGER-M has the self-reconfiguring feature.

2. Beacon Message Broadcast Interval and Arrival Time

PAGER relies on periodically broadcasted beacons to provide nodes with their neighbors' location information. The performance is greatly affected by the choice of the beacon interval B. In scenarios with high mobility, in order to have a complete and up-to-date neighbor list, the beacon broadcast interval should be small. However, the performance of PAGER in mobile sensor networks can be seriously impaired if beacon broadcasting is too frequent. This is mainly because beacon packets contend with data packets for wireless media, which will cause high data packet loss ratio due to interference. In PAGER-M, we randomize and properly prolong the beacon broadcast interval to reduce the interferences caused by beacon packets. On the other hand, the longer the beacon broadcast interval, the more out-of-date neighbor information in a

sensor node's neighbor list. This may cause a node to send out a data packet to a neighbor that has moved out of its communication range. To minimize the wrong forwarding decision caused by out-of-date neighbor information, the forwarding decision is made not only based on neighbors' costs but also the arrival time of a neighbor's beacon message. In PAGER-M, when a beacon message is received, the arrival time is recorded in the neighbor list with the beacon sender. When a sending node is choosing its neighbors to forward, both the cost and the arrival time of the neighbor are considered. When a node is forwarding messages at moment t, only those neighbors with beacon messages received at moment $t' > t - \tau$ are considered as forwarding candidates, where τ is a preset threshold. As we will show in the next section, ad hoc choices of B and τ are used to achieve the best performance of PAGER for different network scenarios.

3. Path Redundancy

With proper density and arrival time threshold τ, multiple choices of forwarding neighbors may still available for a forwarding node that follows the cost gradients. This provides redundancy for a sending node to choose an up-to-date neighbor to forward packets. We use this path redundancy to reduce forwarding failure. To clearly explain how path redundancy helps to improve delivery ratio, we introduce an example here. In the example shown in Fig. 40, node A (with a cost 28) has three forwarding candidates (nodes B, C and D). How to choose the forwarding strategy among the forwarding candidate has impact on the performance of PAGER-M. For example, since nodes B and C have smaller cost than node D, if node A forwards a packet to nodes B or C, the packet will experience 7 hops before reaching the BS. Compared to the 8-hop path length via node D, choosing node B or C may be advantageous. However, as shown in Fig. 40, since nodes B and C are moving away

from node A, packet forwarding may face transmission failure. Compared to nodes B and C, forwarding the packet to node D may be a "safer" choice although it increases the path length of the packet to the BS. In PAGER-M, the latter conservative forwarding strategy is adopted to reduce the chance of transmission failure. This strategy is implemented by choosing the neighbor with closest arrival time among available forwarding candidates (a forwarding candidate should have lower cost than the forwarding sensor node).

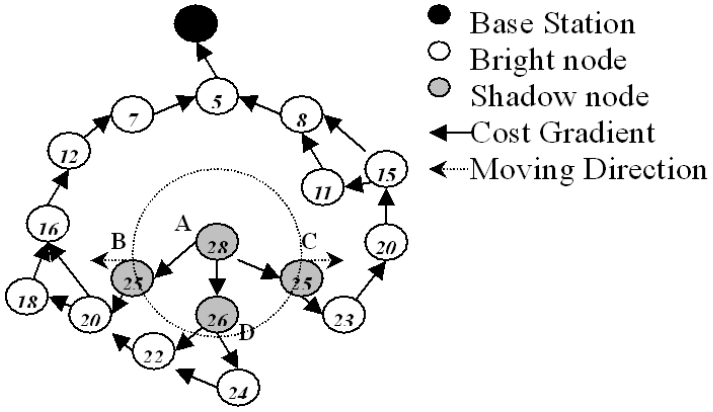


Fig. 40. Utilizing path redundancy to reduce transmission failure.

E. Stationary Sensor Networks Simulation

To test the performance of our proposed PAGER algorithm, we implement PAGER into a protocol called PAGER-S using $ns-2$ [35] for topologies with limited mobility, where S means topologies with limited mobility. In PAGER-S, two mechanisms can trigger the sending of beacon messages. First, beacon messages in PAGER-S are sent out when the timeout of beacon interval occurs. Similar to SPEED [29], in

stationary sensor network, the amount of this kind of beacon messages are reduced by prolonging the beacon broadcast interval. In the following simulation, we fix the beacon broadcast interval to $30s$. Second, to shorten the convergence time, all nodes in PAGER-S send out their beacon messages containing status and cost information after their status and costs change. We reduce collisions happening in the MAC layer by randomizing the sending time of these beacon messages. Our simulation parameters are listed in Table II. The BS is located at $(500, 1000)$. We simulate 7 CBR flows originated from 7 randomly chosen nodes across the whole networks. Each CBR flow is sent at 256 bps and uses 32-byte packets. Random topologies (15 for each degree, number of nodes) are created. All simulation results are based on the average values of these topologies.

Table II. Simulation Parameters.

Nodes	$100 \sim 500$
Workload	$256bps \times 7$
Average Degree	$6 \sim 10$
Region	$1000m \times 1000m$
Simulation Length	$100s$
MAC Layer	802.11
Routing Protocol	GPSR, PAGER-S, AODV
Propagation Model	$Two - Ray$

We first simulate the distribution of shadow/bright areas in 500-node networks. Then the convergence time of PAGER-S is obtained in sparse networks (average degree = 6) with different number of nodes ($100 \sim 500$ nodes). Finally, by comparing with GPSR [39], AODV [54] and shortest path routing, we evaluate the following

metrics of PAGER-S: 1) packets delivery ratio, 2) path length, 3) control overhead and 4) energy consumption. We use AODV and GPSR to compare PAGER-S in large-scale networks (500 nodes). The reason we choose AODV rather than DSR [38], which is used in [29], [39], is because we find that AODV performs better in large networks. GPSR is chosen because it is currently the most accepted stateless recover mode method of location-based routing.

1. Distribution of Shadow Nodes/Areas

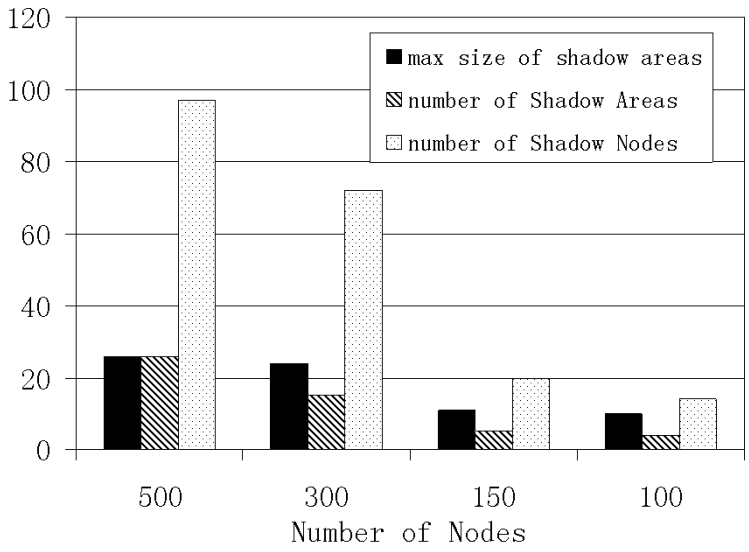


Fig. 41. Distribution of shadow nodes/areas. Average degree=7.

The performance of PAGER algorithm depends on the distribution of shadow and bright areas. We randomly create sparse network topologies (100 $\sim$ 500 nodes, average degree= 7). The average distribution of shadow nodes/areas is shown in Fig. 41. From the results we can see that 15% $\sim$ 25% nodes in these networks are shadow

nodes, which will cause at least the same percentage of packet loss rate using greedy forwarding without solving the dead-end problem. We can see that the maximum size of shadow areas is small (26 nodes) even in large-scale networks (500 nodes). Since the convergence time of the PAGER algorithm depends on the maximum size of shadow areas, as we see later, PAGER-S quickly converges to stabilization. The average number of shadow areas is small (97 shadow nodes with 26 shadow areas on average in 500-node networks). Since the routing overhead of PAGER algorithm depends on the average size of shadow areas, as we see later, the routing overhead of PAGER-S only slightly surpasses the amount of control packets of GPSR.

2. Convergence Time of PAGER-S

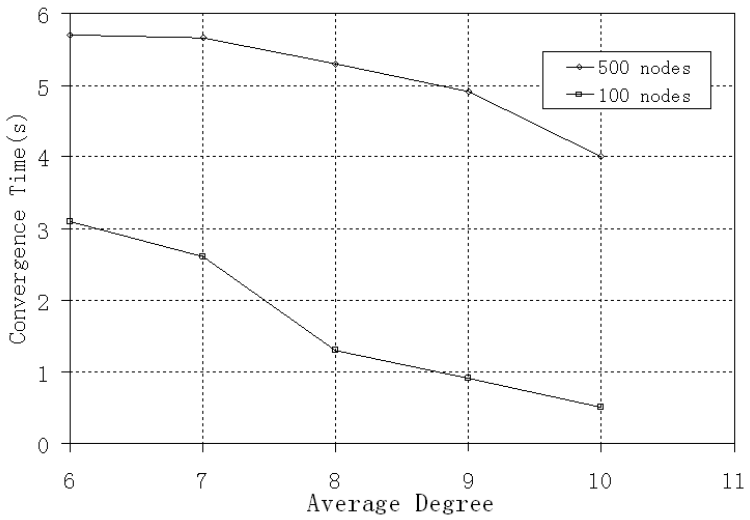


Fig. 42. Convergence time of PAGER-S.

The PAGER algorithm needs a period of time to stabilize the cost gradients. As

shown in Fig. 42, the average convergence time of PAGER-S is 5.12 $\sim$ 7.28 seconds in large-scale networks (500 nodes). In smaller networks (100 nodes), the average convergence time of PAGER-S is reduced to 0.56 $\sim$ 3.1 seconds.

3. Packet Delivery Ratio

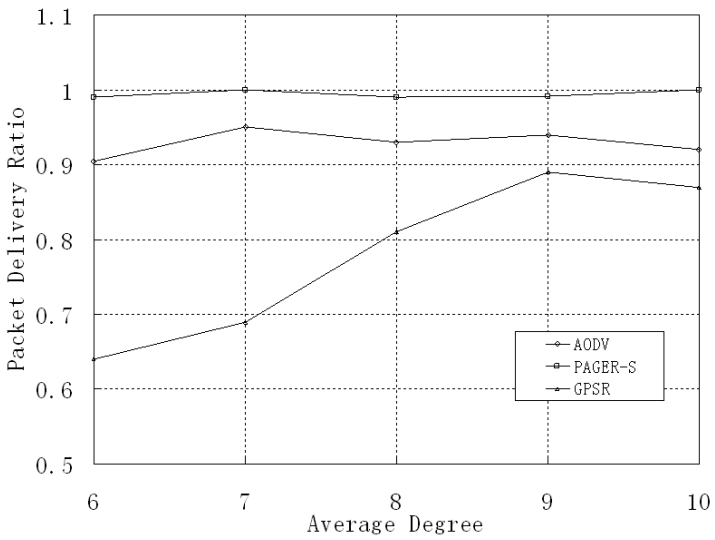


Fig. 43. Packet delivery ratio. 500 nodes.

In static sensor networks, when a packet is sent down to routing protocols, it may be lost mainly due to two reasons. First, routing protocols may cause packet loss without providing routes to destinations. Also, packets may be lost in the un-ideal MAC layer caused by packet collisions/congestions. The packet delivery ratio of PAGER-S is higher than that in AODV and GPSR in 500-node networks. As Fig. 43 shows, PAGER-S delivers about 5% more data packets to the BS than AODV even with a light traffic load (7 CBR flows from 7 nodes). This is because AODV

has to flood control packets throughout the networks to find paths to the BS, which causes congestions between data packets and control packets across the whole sensor network. We observe a low packet delivery rate ($< 82\%$) of GPSR in sparse networks (degree=$6 \sim 8$) as shown in Fig. 43. This is due to the long path length of GPSR that increases a data packet's risk of being lost due to congestion/packet collisions. Different from AODV and GPSR, PAGER-S delivers more data packets by establishing the cost gradients across the whole network before data transmissions. Further, PAGER-S maintains low control overhead by utilizing geographic information as we see later.

4. Path Length

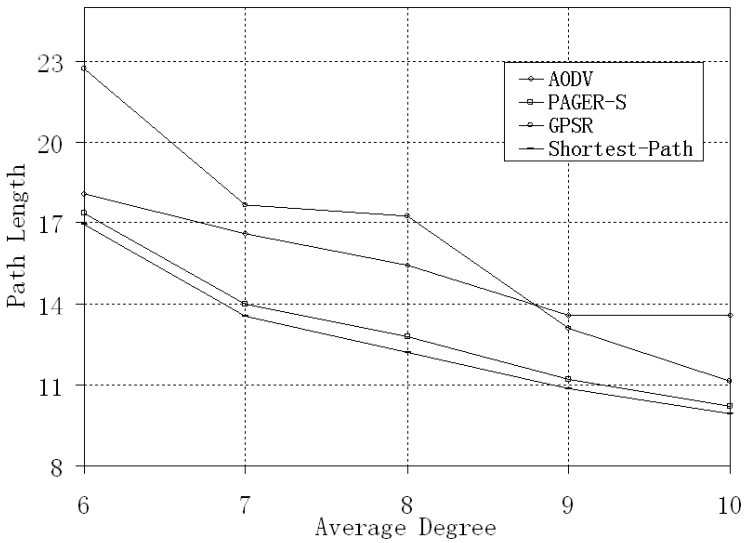


Fig. 44. Path length. 500 nodes.

We simulate 500-node topologies to compare the path length of PAGER-S,

AODV, GPSR and Shortest Path. The simulation results are shown in Fig. 44. In all cases, PAGER-S has a path length close to shortest path and is shorter than that of GPSR and AODV. In sparse networks (average degree=6), the path length of GPSR is about 6 hops longer than that of PAGER-S. PAGER-S also constructs shorter paths than AODV in all these topologies. As shown in Fig. 44, the path length of PAGER-S is 2 hops shorter than that of AODV on average in all cases.

5. Control Overhead

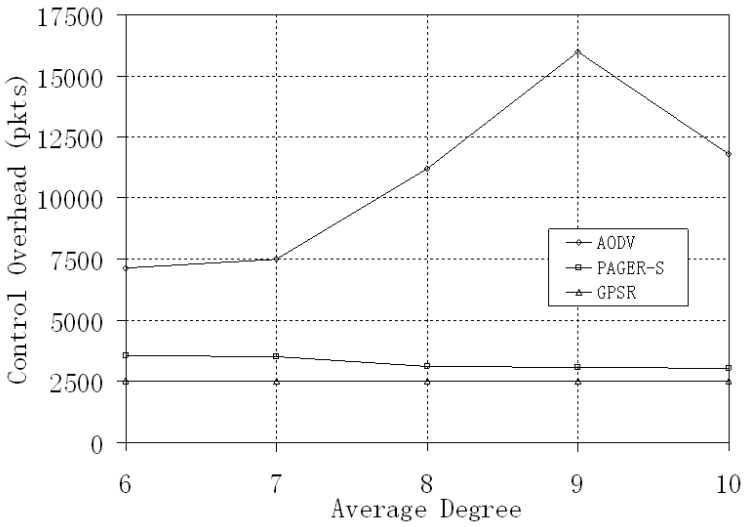


Fig. 45. Routing protocol overhead. 500 nodes.

To test the control overhead of PAGER-S, we compare PAGER-S with AODV and GPSR in sensor networks with 500 nodes. The results are shown in Fig. 45. In all cases, the number of control packets generated in AODV is over 4000 higher than that of PAGER-S. Compared to GPSR, PAGER-S produces a larger large of

control packets. This is because nodes in PAGER-S send out extra beacon messages containing status and cost information when they detect status/cost changes without waiting for the beacon timeout.

6. Energy Consumption

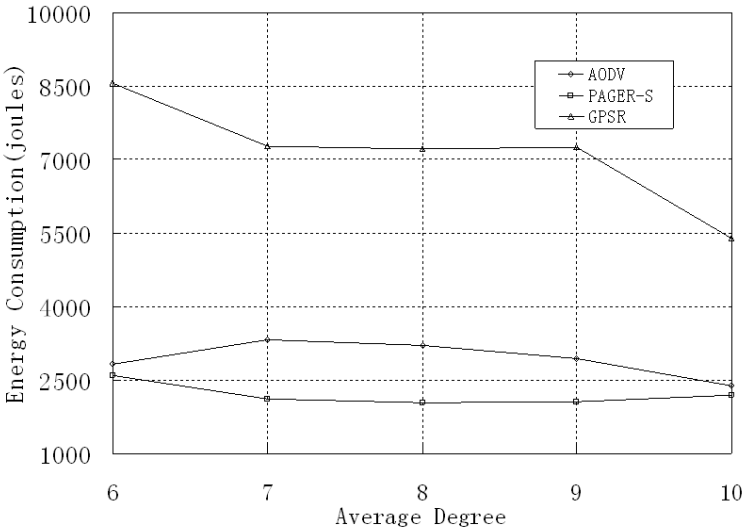


Fig. 46. Energy consumption. 500 nodes.

With low routing overhead and short path length, it is not surprising to see the energy efficiency of the PAGER algorithm. Here we give the energy consumptions of the routing protocols in Fig. 46. GPSR has the highest energy consumption because of its long detours when dead-ends are met, which happens frequently in sparse networks. Although AODV has the highest control overhead as shown previously, it has a relatively short path length, which has advantages in stationary sensor networks.

F. Mobile Sensor Networks Simulation

To test the performance of PAGER algorithm in mobile scenarios, we implement PAGER into a protocol called PAGER-M for frequently changing topologies using ns-2 [35]. In this section, we first give the simulation parameters for the mobile sensor networks simulation. Then different metrics are considered to evaluate the performances of PAGER-M. In our simulations, the following metrics are evaluated: 1) packet delivery ratio, 2) path length, 3) routing overhead and 4) energy consumption.

1. Simulation Parameters

Table III. Parameters in Mobile Sensor Networks Simulation.

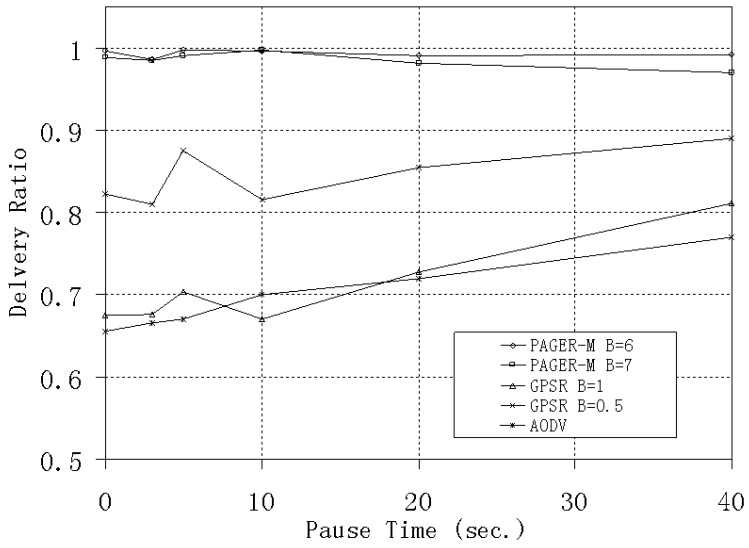
Number of Nodes	200
Average Degree	$6 \sim 10$
Workload	$256bps \times 8$
Region	$1000m \times 1000m$
Simulation Length	$100s$
Routing Protocol	GPSR, PAGER-M, AODV
MAC Layer	802.11
Propagation Model	$Two - Ray$
Mobility Model	Waypoint
Maximum Speed	$20m/s$
Communication Range	$155,126m$
Pause Time	$0 \sim 40$ seconds

Our simulation parameters are listed in Table III. The BS is located at $(500, 1000)$. We simulate 200-node sensor networks and constrain the average degree to 15 and 10 by setting the communication range to 155m and 126m, respectively. We simulate 8 CBR flows originated from 8 randomly chosen nodes across the whole networks. Each CBR flow is sent at 256 bps and uses 32-byte packets. Each sensor follows the random waypoint motion model [10] to move randomly to a direction with a random speed uniformly selected from 0 m/s to 20 m/s. The beacon interval B of PAGER-M is set to range from 3 seconds to 7 seconds according to the average degree of sensor networks. In GPSR, the beacon interval B is set to $0.5 \sim 1$ second to achieve its best delivery ratio.

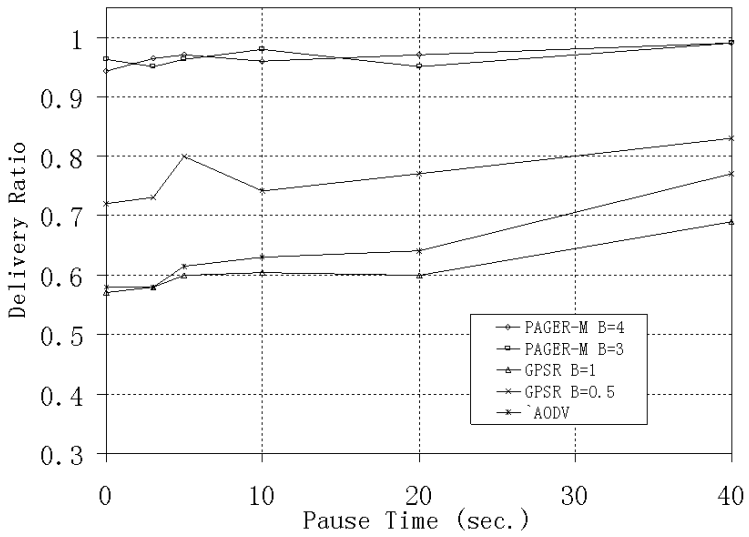
We randomly generate 5 motion patterns for each scenario (different in pause time, maximum communication ranges and number of nodes). Each motion pattern lasts 100 seconds. All experimental results are based on the average performance on these motion patterns.

2. Packet Delivery Ratio

The packet delivery ratio is presented in Fig. 47. We control the mobility of nodes by varying the pause time. In Fig. 47(a), we set the communication range of sensor nodes to 155 m, which creates network topologies with average degree of 15. In Fig. 47(b), we reduce the communication range to 126 m, which results in network topologies with average degree of 10.



(a)



(b)

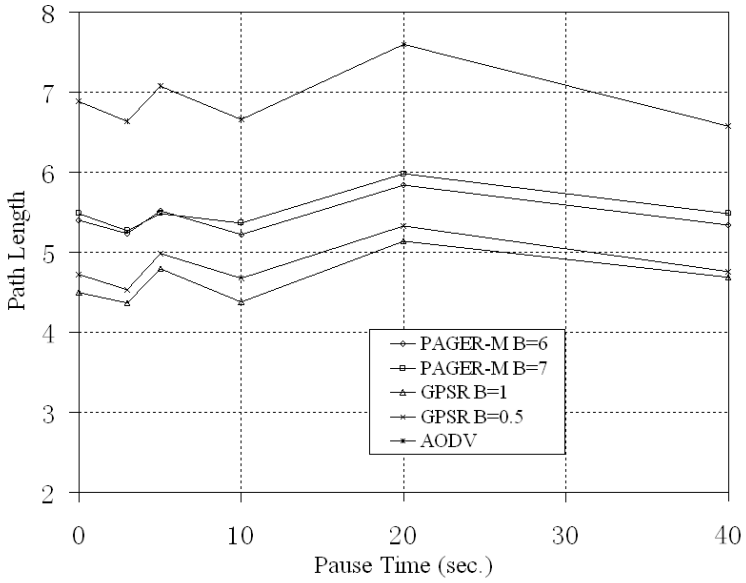
Fig. 47. Delivery ratio. (a) Communication range=155m, 200 nodes. (b) Communication range=126 m, 200 nodes.

In the communication range=155 m case, we set the beacon broadcast interval to $6 \sim 7$ seconds. As we see in Fig. 47(a), the average delivery ratio of PAGER-M is about 99.3%, significantly higher that of GPSR ($< 90\%$) and AODV ($< 80\%$). As we increase the beacon broadcast interval of GPSR from 0.5 second to 1 second, as Fig. 47(a) shows, we observe significant decreasing of delivery ratio. This is because of GPSR's dependency on planarized graphs, which has high requirement of accuracy information of a sensor's local topology. When the communication range is shortened to 126 m, we set the beacon interval $B = 3 \sim 4$ seconds in PAGER-M. As shown in Fig. 47(b), PAGER-M maintains an average delivery ratio of 96.6% while GPSR and AODV have average delivery ratios below 80% and 70%, respectively.

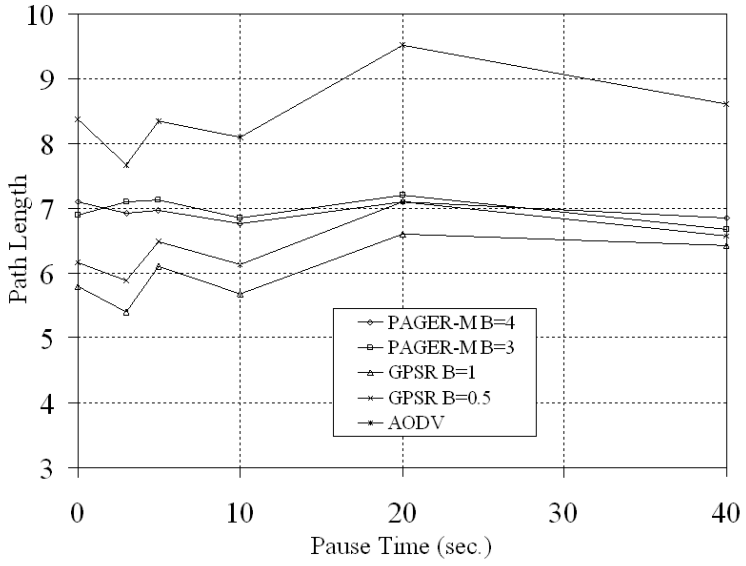
3. Path Length

The performance of average path length of the protocols is presented in Fig. 48 with different parameters. When the communication range is 155m, as shown in Fig. 48(a), PAGER-M has an average path length in between those of GPSR and AODV. This is not surprising because of the conservative choice of forwarding destinations in PAGER-M (only beacon message senders received less than 3 seconds ago are considered as forwarding candidates). This conservative strategy makes a sending node choose not the neighbor closest to the BS but the "safest" one.

When we decrease the communication range to 126m while increasing the beacon interval to $3 \sim 4$ seconds in PAGER-M, as shown in Fig. 48(b), we observe that the average path length of PAGER-M increases to about 7. Again in this case, GPSR achieves the shortest path length compared to PAGER-M and AODV.



(a)



(b)

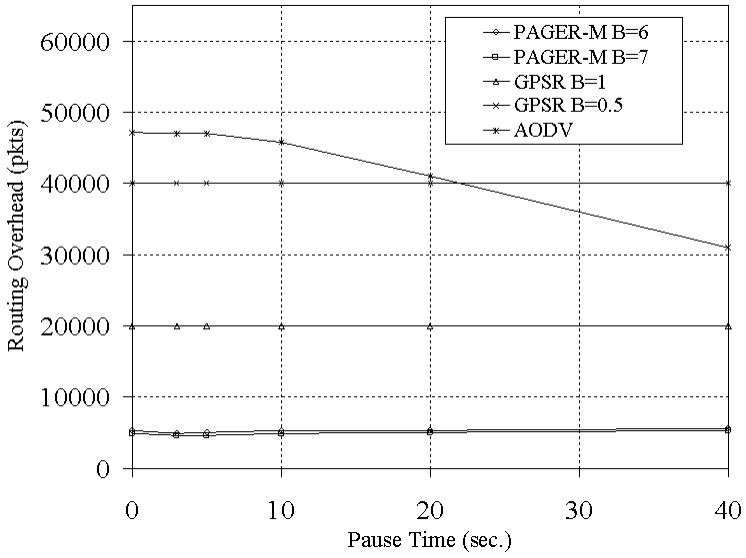
Fig. 48. Path length. (a) Communication range=155 m, 200 nodes.
(b) Communication range=126 m, 200 nodes.

4. Routing Overhead

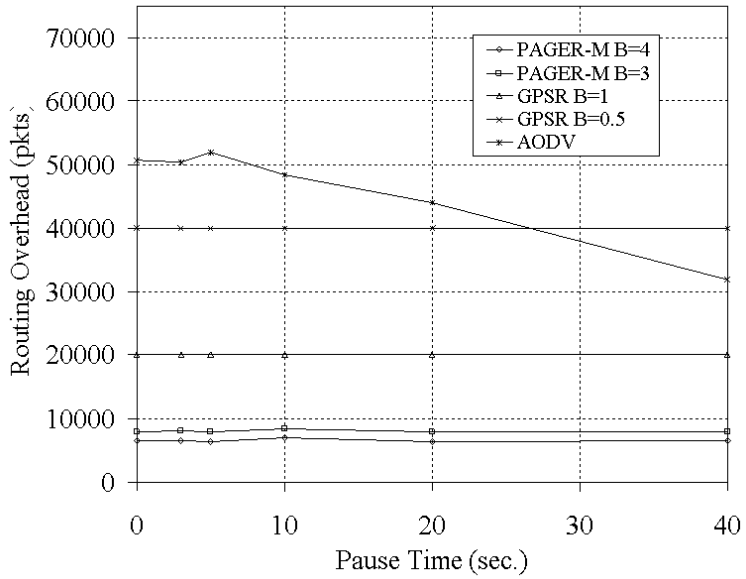
We present our experimental results in Fig. 49. In our simulations, the implicit beacon function of GPSR is disabled. When we set the communication range to 155m as shown in Fig. 49(a), the routing overhead of PAGER-M is significantly lower than those of GPSR and AODV. Again this is due to the long beacon broadcast interval of PAGER-M, which reduces a bulk of the routing overhead in PAGER-M. The routing overhead of GPSR is more than twice than that of PAGER-M because of its short beacon interval. We also observe a decrease of the routing overhead in AODV with increasing pause time. This is because with reduced mobility, AODV sends out less routing packets to repair broken links. When the communication range is reduced to 126 m as shown in Fig. 49(b), in order to maintain the high delivery ratio, PAGER-M reduces beacon interval to $3 \sim 4$ seconds, which causes the increasing of routing overhead to about 6500 and 8000, respectively. However, compared with GPSR and AODV, PAGER-M still has significantly less routing overhead.

5. Energy Consumption

Without the requirement of planariztion of underlying graphs, PAGER-M utilizes the path redundancy to prolong the broadcast interval. By increasing the broadcast interval, PAGER-M reduces not only the control overhead but also the energy consumption without experiencing much performance deterioration. We simulate 200-node networks to observe the energy consumption of PAGER-M. The performances are shown in Figs. 50(a) and 50(b). We can see from both figures that PAGER-M achieves the lowest energy consumption with prolonged broadcast interval compared to GPSR and AODV. Again, broadcast interval in GPSR is set to be 0.5/1.0 second to achieve best performance in terms of delivery ratio and path length. We also see

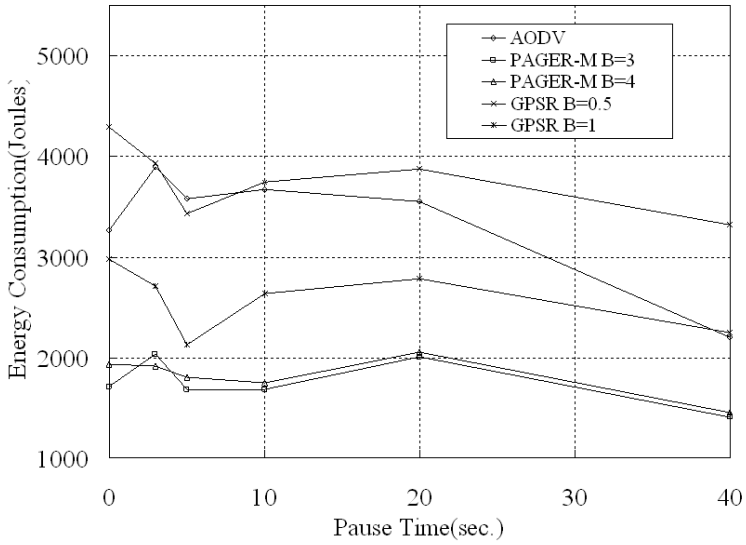


(a)

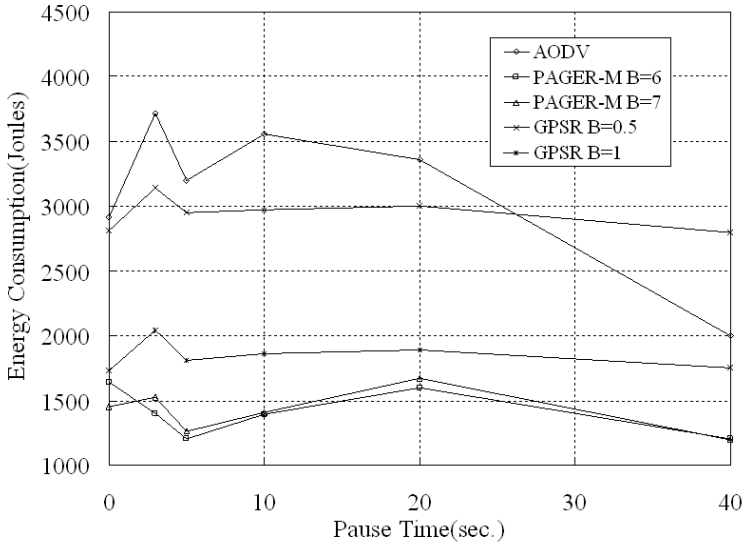


(b)

Fig. 49. Routing overhead. (a) Communication range=155m, 200 nodes. (b) Communication range=126m, 200 nodes.



(a)



(b)

Fig. 50. Energy consumption. (a) Communication range=155m, 200 nodes. (b) Communication range=126m, 200 nodes.

that the energy consumption of PAGER-M and GPSR does not change a lot with the increasing of pause time. This is due to their periodical broadcast sending.

G. Related Work

In single-path/flooding method proposed by Stojmenovic and Lin [60], when a concave node receives a message, it attempts to use flooding to send copies of messages to all its neighbors. By memorizing pass messages, the concave nodes can refuse receiving the same message again. The single-path/flooding method is proved to guarantee delivery, and the paths formed by this protocol are loop-free. Compared to their scheme, PAGER uses single path strategy and does not require nodes to memorize past traffic.

Bose *et al.* and Karp *et al.* independently propose Greedy-Face-Greedy (GFG) [8] and Greedy Perimeter Stateless Routing (GPSR) [39] to handle the dead-end problem, which are the most relevant solution to PAGER. These two schemes are two quite similar routing schemes, which route packets around faces when concave nodes receive packets. GFG/GPSR does not require nodes to memorize past path/traffic and guarantee a sensor node with a path to the BS if there exists one. The detours along faces in GFG/GPSR may produce long paths compared to the shortest path. GFG/GPSR requires planarization of network topologies, which increases computational complexity in each sensor node. In contrast, PAGER produces path lengths close to the shortest path and does not require planarization of underlying network topologies.

SPEED [29] is a location-based routing protocol specifically designed for real-time applications in sensor networks. SPEED partly handles the dead-end problem by backpressure, which however cannot guarantee a forwarding message have a path

to the BS if there exists one. Instead, our scheme provides each message a path to the BS if such a path exists.

Fang *et al.* [25] give a distributed algorithm to locate stuck nodes. Stuck nodes are actually the superset of concave nodes according to our definition. After locating stuck nodes, they presented a distributed algorithm to find out holes in sensor networks. By memorizing the shapes of holes in sensor networks, when a packet gets stuck in concave nodes, it computes the shorter side of a hole to reach the BS. This scheme does not require planarization of network topologies and can achieve shorter path lengths than GPSR. However, when holes in sensor networks are large, communication overhead and energy consumption in nodes along holes are large. Further, the shapes of holes in sensor networks have to be updated frequently in scenarios with high mobility. Compared to the works by Fang *et al.* [25], PAGER does not require memorizing shapes of holes and can adapt to mobility rapidly.

A new geographic forwarding scheme, Geographic random forwarding (GeRaF), is proposed in [69], [70]. This scheme is similar to greedy forwarding in choosing a closer neighbor to the destination in terms of Euclidean distance. GeRaF differentiates itself from greedy forwarding by relaying packets based on best-effort basis without knowing a relay a priori by the sender. GeRaF trades latency for the elimination of the broadcasting overhead, but does not offer a solution for the dead-end problem. PAGER, however, can be combined with greedy forwarding as an alternative solution for GFG/GPSR to provide a complete location-based routing scheme for sensor networks.

Faruque and Helmy propose Gradient-Based Routing in sensor networks [26], which also uses gradients to help establish routes from sources to destinations. However, the gradients they used are based on physical phenomena, like temperature, humidity, etc. PAGER, as a location-based routing protocol, establishes gradients

across the whole sensor network based on locations of the BS and the sensor nodes.

GRAB is proposed by in [66], which has the same idea of establishing a cost field across the whole sensing field as in PAGER. Compared to PAGER, GRAB has no dead-end problem since it requires broadcasting messages from the BS across the whole sensor network. As an alternative method of GPSR in sensor networks, PAGER solves the dead-end problem inherited in geographic routing by establishing the cost field based on the location information of sensor nodes and the BS. Hence, PAGER and GRAB belong to two different categories of routing protocols.

CHAPTER V

CONCLUSION

A. Summary

In this dissertation, we propose a 3D face recognition system with wireless transportation. In this system, we consider designing three important components, namely feature extraction, classification and distributed routing. In the system, 3D range images are captured by sensor with/without mobility and then transferred to the core face recognition part constituted by the feature extraction component and the classification component. The feature extraction component processes the input range images and produce features that are forwarded to the classification component that classify faces in the range images. While a successful feature extraction component should be able to produce reliable features that can effectively identify a face, the purpose of the classification component is to effectively to differentiate a face from others.

In Chapter II, we first consider extracting features from range images based on 3D Morphable Model. In our proposed scheme, the RMS error between the 3D Morphable Model and an input range image is minimized by optimizing the shape coefficients. The optimized shape coefficients are then used as features of the face in the input range image. The advantages of the scheme lie in its adaptive hole avoiding, its insensitivity of facial expression and its extraction of both detailed and overall geometric information.

In Chapter III, we presented a novel scheme for 3D face recognition based on face warping. Our scheme can automatically identify a person from a range image

captured from 3D camera and is insensitive to holes, facial expressions and hair. In our scheme, both detailed and the overall geometric information is utilized to recognized a person.

We first introduce the face warping procedure based on landmark points defined in generic face. Based on a set of weighting coefficients, a combination of the face warps derived from example faces can be used to warp the generic face as a synthesized range face. After that, the selection of example faces is described. In order to make the synthesized face similar to a target face, Newton's method is used to minimize the range differences by optimizing the weighting coefficients. After the optimization, the optimized weighting coefficients are used as the feature of the target face. A classifier based on Mahalanobis distance is used to verify the identity of the target face. We tested our scheme on a database containing more than 600 range faces. The experimental results showed that our scheme can achieve a high recognition rate.

In Chapter IV, we present a distributed routing algorithm named PAGER to solve the dead-end problem of location-based routing in sensor networks. We prove that PAGER has loop-free guarantee delivery property. We implement the algorithm in protocols named PAGER-S/M for sensor networks without/with mobility. The performance of PAGER-S/M is compared with AODV, GPSR and the shortest path with respect to delivery ratio, path length, routing overhead and energy efficiency. Experimental results confirmed the advantages of our PAGER algorithm.

B. Future Directions

Our proposed 3D face recognition system performs very well in both recognition and wireless transportation. In our 3D face recognition scheme based on generic face warping, the equal error rate reaches over 99.5%. However, the 3D face recognition

system relies on the normalized faces in range images. If the normalization of faces in range images cannot produce well normalized faces, the feature extraction procedure may produce unreliable features. And the recognition performance will largely deteriorate. Hence reliable normalization method is an interesting research direction.

Although our proposed feature extraction scheme can extract features with 20 seconds on a Pentium 4 workstation, it is always desirable for a feature extraction scheme to produce features in real time. Therefore, further reduction of feature extraction time is an exciting direction.

REFERENCES

[1] B. Achermann, X. Jiang, and H. Bunke, "Face recognition using range images," in *Proc. IEEE International Conference on Virtual Systems and Multimedia*, Geneva, Switzerland, Sept. 1997.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[3] M. Bartlett and T. Sejnowski, "Independent components of face images: A representation for face recognition," in *Proc. 4th Annual Joint Symposium on Neural Computation*, Pasadena, CA, May 1997.

[4] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[5] P. Besl and R. Jain, "Invariant surface characteristics for 3D object recognition in range images," *Computer Vision, Graphics, and Image Processing*, vol. 33, pp. 33–80, 1986.

[6] V. Blanz and T. Vetter, "Morphable model for the synthesis of 3D faces," in *Proc. the 26th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, July 1999.

[7] ——, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, 2003.

[8] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. 3rd International Workshop on*

Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, WA, Aug. 1999.

[9] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition," *Computer Vision and Image Understanding*, vol. 101, no. 1, pp. 1–15, 2006.

[10] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. 4th Annual International Conference on Mobile Computing and Networking (MobiCom 1998)*, Dallas, TX, Oct. 1998.

[11] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Three-dimensional face recognition," *International Journal of Computer Vision*, vol. 18, no. 1, pp. 5–30, 2005.

[12] J. Y. Cartoux, J. T. LaPreste, and M. Richetin, "Face authentication or recognition by profile extraction from range images," in *Proc. Workshop on Interpretation of 3D Scenes*, Austin, TX, Nov. 1989.

[13] K. R. Castleman, *Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1996.

[14] K. I. Chang, K. W. Bowyer, and P. J. Flynn, "Effects on facial expression in 3D face recognition," in *Proc. SPIE Conference Biometric Technology for Human Identification*, Orlando (Kissimmee), FL, Apr. 2005.

[15] ——, "An evaluation of multi-modal 2D+3D face biometrics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 619–624, 2005.

[16] ——, "Multiple nose matching for 3D face recognition under varying facial expression," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1–6, 2006.

[17] K. I. Chang, P. J. Flynn, and K. W. Bowyer, "Multi-modal 2D and 3D biometrics for face recognition," in *Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, Nice, France, Oct. 2003.

[18] C. Chua, F. Han, and Y. K. Ho, "3D human face recognition using point signature," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, Mar. 2000.

[19] J. Cook, V. Chandran, S. Sridharan, and C. Fookes, "Face recognition from 3D data using iterative closest point algorithm and gaussian mixture models," in *Proc. Second International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2004)*, Thessaloniki, Greece, Sept. 2004.

[20] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA: The MIT Press, 2001.

[22] I. J. Cox, J. Ghosn, and P. N. Yianilos, "Feature-based face recognition using mixture distance," in *Proc. International Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 1996.

[23] R. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY: John Wiley and Sons, 2001.

[24] H. Edelsbrunner, *Geometry and Topology for Mesh Generation*. Cambridge, England: Cambridge University Press, 2001.

[25] Q. Fang, J. Gao, and L. Guibas, "Locating and bypassing routing holes in sensor networks," in *Proc. IEEE INFOCOM'04*, Hong Kong, China, Mar. 2004.

[26] J. Faruque and A. Helmy, "RUGGED: Routing on fingerprint gradients in sensor networks," in *Proc. IEEE International Conference on Pervasive Services (ICPS)*, Beirut, Lebanon, July 2004.

[27] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," *ISI Research Report*, ISI/RR-87-180, Mar. 1987.

[28] G. Gordon, "Face recognition based on depth and curvature features," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, IL, June 1992.

[29] T. He, J. Stankovic, C. Lu, and T.Abdelzaher, "SPEED: A stateless protocol for real-time communication in sensor networks," in *Proc. International Conference on Distributed Computing Systems (ICDCS2003)*, Providence, RI, May 2003.

[30] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. Hawaii International Conference on System Sciences*, Maui, HI, Jan. 2000.

[31] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. 5th Annual International Conference on Mobile Computing and Networking (MobiCom 1999)*, Seattle, WA, Aug. 1999.

[32] C. Hesher, A. Srivastava, and G. Erlebacher, "A novel technique for face recognition using range imaging," in *Proc. International Sympanym Second International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2004)*, Thessaloniki, Greece, Sept. 2004.

[33] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[34] J. Huang, B. Heisele, and V. Blanz, "Component-based face recognition with 3D morphable models," in *Proc. Audio- and Video-Based Biometric Person Authentication (AVBPA)*, Guildford, UK, June 2003.

[35] Information Science Institute, University of Southern California, "NS2 network simulator," www.isi.edu/nsnam/ns, accessed on June 05, 2007.

[36] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 2–16, 2003.

[37] R. Jain, A. Puri, and R. Sengupta, "Geographical routing using partial information for wireless ad hoc networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 48–57, 2001.

[38] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, vol. 353, pp. 153–181, 1996.

[39] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, Aug. 2000.

[40] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proc. 11th Canadian Conference Computational Geometry*, Vancouver, Canada, Aug. 1999.

[41] A. Lanitis, C. Taylor, and T. Cootes, "Automatic face identification system using flexible appearance models," *Image and Vision Computing*, vol. 13, no. 1, pp. 393–401, 1995.

[42] J. C. Lee and E. Milios, "Matching range images of human faces," in *Proc. International Conference on Computer Vision*, Osaka, Japan, Dec. 1990.

[43] Y. Lee and J. Shim, "Curvature-based human face recognition using depth-weighted hausdorff distance," in *Proc. International Conference on Image Processing*, Singapore, Oct. 2004.

[44] J. Li, J. Jannotti, D. D. Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. 6th Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, Aug. 2000.

[45] C. Liu and H. Wechsler, "Evolutionary pursuit and its application to face recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 570–582, 2000.

[46] X. Liu, A. Srivastava, and K. Gallivan, "Optimal linear representations of images for object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 662–666, 2004.

[47] X. Lu, D. Colbry, and A. K. Jain, "Matching 2.5D scans for face recognition," in *Proc. International Conference on Pattern Recognition*, Cambridge, UK, Aug.

2004.

[48] X. Lu and A. K. Jain, "Deformation modeling for robust 3D face matching," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, NY, June 2006.

[49] X. Lu, A. K. Jain, and D. Colbry, "Matching 2.5D face scans to 3D models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 31–43, 2006.

[50] A. Manjeshwar and D. P. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," in *Proc. 15th International Parallel and Distributed Processing Symposium*, San Francisco, CA, Apr. 2001.

[51] A. S. Mian, M. Bennamoun, and R. Owens, "Region-based matching for robust 3D face recognition," in *Proc. British Machine Vision Conference (BMVC)*, Oxford, UK, Sept. 2005.

[52] T. Nagamine, T. Uemura, and I. Masuda, "3D facial image analysis for human identification," in *Proc. International Conference on Pattern Recognition*, The Hague, The Netherlands, Aug. 1992.

[53] W. Noh, K. Lim, J. Lee, , and S. An, "Time-synchronized neighbor nodes based redundant robust routing (TSN3R3) for mobile sensor networks," in *Proc. 1st European Workshop Wireless Sensor Networks (EWSN'04)*, Berlin, Germany, Jan. 2004.

[54] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, Feb. 1999.

[55] P. J. Phillips, "Support vector machines applied to face recognition," in *Proc. Conference on Advances in Neural Information Processing Systems*, Denver, CO, Dec. 1998.

[56] F. Samaria and S. Young, "HMM-based architecture for face identification," *International Journal of Image and Vision Computing*, vol. 12, no. 8, pp. 537–543, 1994.

[57] A. Srivastava, X. Liu, and C. Hesher, "Face recognition using optimal linear components of range images," *Journal of Image and Vision Computing*, vol. 24, no. 3, pp. 291–299, 2006.

[58] I. Stojmenovic, "Position-based routing in ad hoc networks," *IEEE Communications Magazine*, vol. 40, pp. 128–134, 2002.

[59] I. Stojmenovic and X. Lin, "GEDIR: Loop-free location based routing in wireless networks," in *Proc. International Conference on Parallel and Distributed Computing and Systems (IASTED)*, Cambridge, MA, Nov. 1999.

[60] ——, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1023–1032, 2001.

[61] H. T. Tanaka, M. Ikeda, and H. Chiaki, "Curvature-based face surface recognition using spherical correlation principal directions for curved object recognition," in *Proc. Third International Conference on Automated Face and Gesture Recognition*, Nara, Japan, Apr. 1998.

[62] J. Tateson and I. W. Marshall, "A novel mechanism for routing in highly mobile ad hoc sensor networks," in *Proc. 1st European Workshop Wireless Sensor*

Networks (EWSN'04), Berlin, Germany, Jan. 2004.

[63] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[64] L. Wiskott, J. Fellous, and C. Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.

[65] C. Xu, Y. Wang, T. Tan, and L. Quan, "Automatic 3D face recognition combining global geometric features with local shape variation information," in *Proc. International Conference on Pattern Recognition*, Cambridge, UK, Aug. 2004.

[66] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A robust data delivery protocol for large scale sensor networks," *ACM WINET (Wireless Networks)*, vol. 11, no. 3, pp. 285–298, 2005.

[67] L. Zhang and D. Samaras, "Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 351–363, 2006.

[68] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Survey*, vol. 35, no. 4, pp. 399–458, 2003.

[69] M. Zorzi and R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance," *IEEE Trans. Mobile Computing*, vol. 2, no. 4, pp. 349–365, 2003.

[70] ——, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *IEEE Trans. Mobile Computing*, vol. 2, no. 4, pp. 337–348, 2003.

[71] L. Zou, S. Cheng, M. Lu, Z. Xiong, and K. Castleman, "A mobile criminal identification system," in *Proc. IEEE Conference on Technologies for Homeland Security*, Woburn, MA, May 2007.

[72] L. Zou, S. Cheng, Z. Xiong, M. Lu, and K. Castleman, "Facial feature extraction from range images using a 3D morphable model," in *Proc. 32th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'07)*, Honolulu, HI, Apr. 2007.

[73] ——, "3D face recognition based on warped example faces," *IEEE Trans. Information Forensics and Security*, to appear, 2007.

[74] L. Zou, Z. Xiong, and M. Lu, "PAGER: A distributed algorithm for the dead-end problem of location-based routing in sensor networks," in *Proc. International Conference on Computer Communications and Networks*, Chicago, IL, Oct. 2004.

[75] ——, "PAGER-M: A novel location-based routing protocol for mobile sensor networks," in *Proc. First International Workshop on Broadband Wireless Services and Applications*, San Jose, CA, Oct. 2004.

[76] ——, "A distributed algorithm for the dead end problem of location based routing in sensor networks," *IEEE Trans. Vehicular Technology*, vol. 54, no. 4, pp. 1509–1522, 2005.

VITA

Le Zou has B.S. and M.S. degrees in electrical engineering from Southeast University, in 1999 and 2002, respectively. He earned his Ph.D. in Computer Engineering in August 2007 from Texas A&M University. He can be reached through Prof. Mi Lu (email: mlu@ece.tamu.edu) at the Department of Electrical and Computer Engineering, Texas A&M University.

The typist for this dissertation was Le Zou.