Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition Letters 27 (2006) 1487–1494

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Combining local features for robust nose location in 3D facial data ☆

Chenghua Xu [a], Tieniu Tan [a,*], Yunhong Wang [b], Long Quan [c]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, P.O. Box 2728, Beijing 100080, PR China
[b] School of Computer Science and Engineering, Beihang University, Beijing 100083, PR China
[c] Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Received 20 September 2004; received in revised form 10 February 2006
Available online 27 April 2006

Communicated by G. Borgefors

Abstract

Due to the wide use of human face images, it is significant to locate facial feature points. In this paper, we focus on 3D facial data and propose a novel method to solve a specific problem, i.e., locating the nose tip by one hierarchical filtering scheme combining local features. Based on the detected nose tip, we further estimate the nose ridge by a newly defined curve, the Included Angle Curve (IAC). The key features of our method are its automated implementation for detection, its ability to deal with noisy and incomplete input data, its invariance to rotation and translation, and its adaptability to different resolutions. The experimental results from different databases show the robustness and feasibility of the proposed method.

Keywords: Nose tip location; Local surface features; Local statistical features; Included angle curve; SVM

1. Introduction

With the rapid development of 3D capture system, 3D facial data are widely used in many fields, such as modelling, animation and recognition. Most applications depend on robust feature point location. In images of faces that include color, the pupils have distinctive features since they have black texture and circle-like shapes. In contrast, the nose has the most distinct features in 3D facial data while the region near the eyes becomes flat. Many other features, such as eye corners and mouth corners, are positioned depending on the baseline position of the nose.

This paper focuses on the specific task of developing a robust method to identify the location of the nose in 3D facial data. Here 3D facial data include multiple forms, such as range data, the surface model of the whole head or front of the face, and the scattered or triangulated facial point clouds. We generally call these data as 3D point clouds in the following. To the best of our knowledge, there is no existing work on detecting the nose in 3D point clouds. Many existing works requiring nose detection usually have two solutions: one is to mark feature points (including the nose) manually; the other is to detect the nose automatically. However, the existing automated methods are usually based on the assumption that the nose tip is the highest point in 3D facial data (Lee et al., 1993, 2003; Hesher et al., 2002; Beumier and Acheroy, 2000). Although this assumption can largely reduce the computational complexity of detection, it does not always hold due to the noise and rotation of the subject. Besides that, Gordon (1991) and Kim et al. (2001) used curvature information to detect the nose. Although curvature information is robust to rotation and translation, it is only suitable to

☆ Partially supported by research funds from the NSFC (Grant No. 60121302, 60332010, 60518002 and 60575003), the Outstanding Overseas Chinese Scholars Fund of CAS (Grant No. 2001-2-8), the NSFC/RGC Joint Research Scheme (Grant No. N-HKUST602/05) and the National 973 Program (Grant No. 2004CB318100).

* Corresponding author. Tel.: +8610 8261 4515; fax: +8610 6255 1993.
E-mail address: tnt@nlpr.ia.ac.cn (T. Tan).

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.02.015

clean 3D data and does not work with noisy data. In fact, data directly obtained from laser scanners usually contain many outliers and holes around the eyes, the opened mouth, the hair, the clothes and so on. It is a difficult task to remove all the noise automatically. In this paper, we attempt to develop a method that can deal with not only a well-processed 3D model but also noisy data.

To locate the nose tip in complex 3D data, we must extract the distinguishing features that make the nose tip salient from other points. To make the detection algorithm robust, the used features should also be invariant to rigid transformation. The different kinds of 3D data obtained in different ways usually have different resolutions and quality, and thus the used features should be suitable to multi-resolution. In addition, although the nose usually has less non-rigid transformation than do other facial regions, such as the mouth and the eyes, the collected data usually include not only the whole face, but also some clothing. Thus the proposed features should also be robust to non-rigid transformation. In fact, it is very difficult, perhaps even unrealistic, to find this kind of feature.

Here, we build a new scheme, i.e., a hierarchical filtering scheme that integrates two kinds of weak local features, to locate the nose tip instead of attempting to find one kind of effective feature to solve all the difficulties. After the nose tip is located, the nose ridge is marked using our proposed curve, i.e., the Included Angle Curve (IAC). This proposed method is invariant to rotation and translation, holes and outliers, and suitable to multi-resolution data. To make our method general, our proposed weak features only depend on scattered points, which are the basic component of many kinds of 3D data. An earlier shorter version of the work described in the current paper is published in (Xu et al., 2004). The current paper has the important extension in terms of the description of algorithms, experiments and the performance analysis.

The remainder of the paper is organized as follows. Section 2 describes the hierarchical filtering scheme for nose tip detection in detail. The nose ridge estimation using the Included Angle Curve is presented in Section 3. Section 4 presents experimental results from three different databases, and Section 5 concludes the paper.

2. Nose tip detection

In this section, we describe how to identify the nose tip in facial 3D point clouds. We extract local surface features and local statistical features for each point. Either of the features describes the local characteristics of facial surface. Although each of them is weak and does not identify the nose tip correctly, one hierarchical filtering scheme fusing them can be constructed to improve the precision. In the following, we first provide a definition of effective energy (EE). Then, two kinds of local features and our hierarchical filtering scheme are described in the following three subsections.

First, a definition of effective energy is provided, and both subsequent local features are extracted based on it. For each point in one point cloud, we can find its neighboring points within a given sphere with a radius r and centered at this point. As shown in Fig. 1, for point P, P_i is one neighboring point lying within the sphere, and N_P is its normalized normal. Thus, we can define the *effective energy (EE)*, d_i, for each neighboring point with the inner product of the vectors $P_i - P$ and N_P:

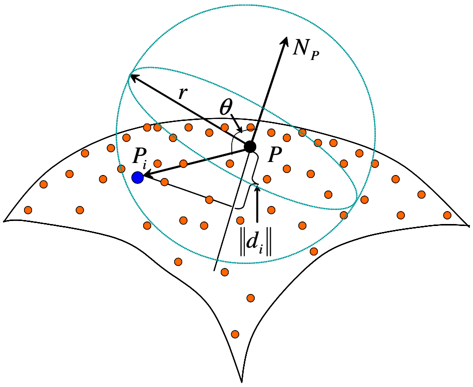

Fig. 1. The effective energy of the neighboring points.

$$d_i = (P_i - P)\cdot N_\mathrm{P} = \|P_i - P\|\cos\theta, \tag{1}$$

where θ is the angle between $P_i - P$ and N_P.

This definition requires the normal of the point, which may be missed in some data. In this kind of data, the normal of each point can be estimated using the method of principal component analysis as described in the following. It is assumed that $\Omega = \{x_1, x_2, \ldots, x_n\}$ is the set of neighboring points. Thus, its covariance matrix can be obtained as follows:

$$Q = \sum_{x_i\in\Omega}(x_i - \bar{x})(x_i - \bar{x})^\mathrm{T}, \tag{2}$$

where x_i is a column vector, representing one point's position in 3D space, and $\bar{x}$ is the mean value of all the points. Thus, Q is a 3×3 positive semi-definite matrix. We determine that λ_1, λ_2 and λ_3 ($\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3$) denote the eigenvalues of Q corresponding to three eigenvectors v_1, v_2 and v_3, respectively. Commonly, the perpendicular direction of the local area has the fewest points. So, v_3 (corresponding to the smallest eigenvalue) similarly corresponds to the normal direction. Then, we use v_3 to approximate the normal of this point.

In one facial point cloud, a number of 3D points compose the whole facial surface. Distinctly, for a point in the peak area of the facial surface, all the effective energy of its neighbors is negative; for a point in the concave area, all the effective energy is positive; for a point in the flat area, all the effective energy is approximately zero; and for a point in the other area, the effective energy is an alternative. Thus, the effective energy can coarsely distinguish some shapes.

To locate the nose tip reliably, we consider two distinguishing properties of the nose tip in the following. The first is that the nose tip is the highest local point; the second is that the nose tip has a special shape, like a peaked cap.

These two properties correspond to Rules 1 and 2 as described in Sections 2.1 and 2.2, respectively. Section 2.3 gives our hierarchical filtering scheme combining two rules to identify the nose tip position.

2.1. Local surface features

This section describes one kind of local feature, i.e., the local surface feature. Rule 1, based on the local surface features, is constructed to select the candidates for the nose tip.

Some existing work utilizes the assumption that the nose tip is the highest point among the facial data. This is correct when the subject is nicely toward the capturing equipment during collection. If the captured subject makes some poses, such as rotation, the nose tip is not the highest point. Moreover, influenced by hair and clothes, this assumption does not always hold. However, the nose tip is still the local highest point in a certain direction even under rotation poses. To use this observation, one key problem is how to determine this "certain direction". In fact, we can use the normal of the point to approximate this direction. Further, we can use the effective energy to describe this property of the local highest point. Thus we obtain Rule 1:

Rule 1. To a point, P, in the point cloud, $NB(P)$ is its neighboring set and $EE(P)$ is the corresponding EE set. Then, P is the nose tip candidate only if all the components in $EE(P)$ are negative.

Mathematically, the neighboring set of point P is $NB(P) = \{p_1, p_2, \ldots, p_n\}$, and the corresponding EE set is $EE(P) = \{d_1, d_2, \ldots, d_n\}$, where d_i is defined by Eq. (1). Then, P is a nose tip candidate only when $\forall d_i < 0$.

This rule is easy to understand. If P lies near the nose tip, that is, lying in the convex area of the facial surface, θ in Eq. (1) is always larger than 90°. Thus, d_i is always negative. The points around the nose tip satisfy this requirement. But this condition is very weak and many points lying in other areas, such as the cheeks and chin, also meet this condition. However, this simple rule is very helpful to reduce the search space effectively.

2.2. Local statistical features

Now, we consider another kind of feature: the local statistical features. Intuitively, the shape of the nose is different from other areas on the facial surface, and it is like a cap. However, it is very difficult to describe it quantitatively. Here, we describe the cap-like shape with two statistical features: the mean and variance. In the following, we will describe them in detail.

For each point, P, we can obtain its neighboring set, $NB(P)$, and corresponding EE set, $EE(P)$

$$NB(P) = \{p_1, p_2, \ldots, p_n\}, \tag{3}$$

$$EE(P) = \{d_1, d_2, \ldots, d_n\}. \tag{4}$$

Then, the mean and variance of the EE set are calculated; that is,

$$\mu = \frac{1}{n}\sum_{i=1}^{n} d_i, \tag{5}$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (d_i - \mu)^2, \tag{6}$$

where n is the number of neighbors and d_i is the effective energy of the ith neighbor.

Each point may be characterized by one feature vector of two components: the mean and variance. Thus, each point is transformed into a 2D space parameterized by the mean and variance. Commonly, the nose tip is the most distinct protuberance on the facial surface. The mean of its neighbors' EE is negatively smaller and the variance is positively larger than that of any other facial area. Thus, the point around the nose tip distributes within a certain scope in a mean–variance domain. Due to the similar shapes of human noses, the points near the nose tips of different people also congregate tightly. Thus, it is expected that we can find a classifier to identify the nose tip.

We can use the classic method of Support Vector Machine (SVM) (Burges, 1998; Joachims, 1999) to identity the boundary between the nose tip and other points in the mean–variance space. All the points are classified into two categories: nose tip points and non-nose tip points. To learn this boundary, one training set is required. In the training set, each sample is described with one vector, (x_i, y_i), $i = 1, 2, \ldots, l$, where $x_i \in R^2$ is the feature vector (μ_i, σ_i^2), and $y_i \in \{-1, 1\}$ denotes two classes.

The non-linear optimal boundary can be considered as

$$w \cdot \Phi(x) - b = 0, \tag{7}$$

where $\Phi(x)$ is one non-linear transformation and $w = \sum_i a_i y_i \Phi(x_i)$. We can find this optimal boundary to distinguish these two classes by minimizing the following energy function:

$$Q(a) = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i,j=1}^{l} a_i a_j y_i y_j K(x_i \cdot x_j), \tag{8}$$

where $K(x_i \cdot x_j) = \Phi(x_i) \cdot \Phi(x_j)$ is the kernel of the SVM classifier. For a testing vector, x, we can use the following discriminant for classifying:

$$f(x) = sgn(w \cdot \Phi(x) - b) = sgn\left(\sum_{i=1}^{l} a_i^* y_i K(x_i, x) + b^*\right), \tag{9}$$

where a^* is obtained by minimizing Eq. (8), and b^* is calculated by using any sample (x_i, y_i) according to Eq. (7). l is the number of support vectors. In our work, $K(x_i \cdot x_j) = \exp\left(-\frac{|x_i - x_j|^2}{\sigma^2}\right)$ and $\sigma^2 = 14$.

We thus obtain another rule:

Rule 2. To a point, P, in the point cloud, $NB(P)$ is its neighboring set, $EE(P)$ is the corresponding EE set, and u and σ^2 are the mean and variance of $EE(P)$. Then, P is the nose tip candidate only if $f(x) > 0$ in Eq. (9).

It is noted that this rule is still weak. Due to the complexity of the scanned surface, especially the crinkles of the clothing, some points in other areas could also be viewed as the nose tip (see Fig. 2). Fortunately, they are commonly too sparse and can be ignored.

This kind of feature characterizes the local statistical property. Thus, it is robust to data noise and multi-resolution. The feature consists of a scalar quantity, and thus it is also robust to rotation and translation.

2.3. The hierarchical filtering scheme

By combining the above two rules, we develop a hierarchical filtering scheme to locate the nose tip robustly. The whole implementation process includes three phases as illustrated in Fig. 3.

In Phase 1, Rule 1 is used to select nose tip candidates. Since this rule is very weak, some points are wrongly classified, i.e., some points in non-nose tip areas are classified as nose tip candidates and some near the nose tip are classified as non-nose tip candidates. The results are illustrated in the upper-left image of Fig. 3. Even so, this rule can largely reduce the searching region. In the next phase, we further select the nose tip candidates.

In Phase 1, the largest computational load focuses on finding the neighboring points for one given point. Usually, we use a local sphere centered at the given point to enclose the neighboring points. If the facial data consist of n points, the computational cost will be $O(n^2)$ distance calculations for all points. We can use an alternative method to accelerate this process in a triangulated point cloud, where there is a table recording the index of the triangle vertices. Thus, we can construct an indexed table beforehand, in which each row records one point's neighboring points connected with one edge. To generate this table requires only $O(m)$ comparison calculations, where m is the number of triangles. After this table is generated, one point's neighbors can be substituted by the points recorded in the corresponding row in the indexed table.

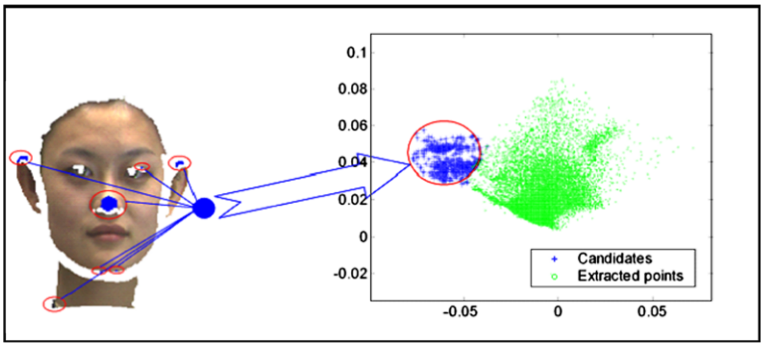

Fig. 2. The filtering result using Rule 2. The blue points in the left image are nose tip candidates by Rule 2 and the right image shows the projection of all the points in the mean–variance space. (For interpretation of the references in colour in this figure caption, the reader is referred to the web version of this article.)

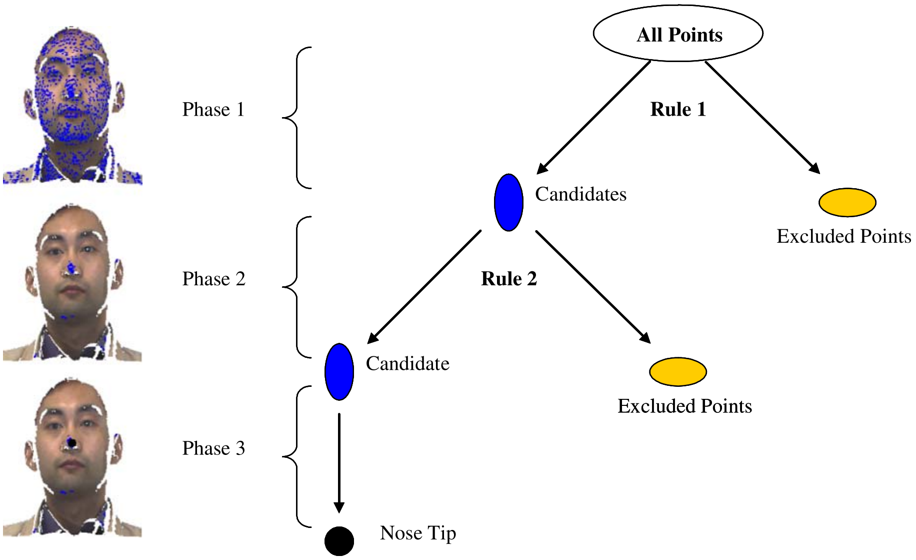

Fig. 3. The process for detecting the nose tip. The left three images show the results of the three phases. The blue points are nose tip candidates and the black point in the bottom image is the detected nose tip. (For interpretation of the references in colour in this figure caption, the reader is referred to the web version of this article.)

In Phase 2, we only consider the obtained candidates in Phase 1. We can find each candidate's neighboring points within a given sphere. The statistical feature vector, $x = (\mu, \sigma^2)$, is calculated using Eqs. (5) and (6). Then, we can further select the candidates of the nose tip by Rule 2. In this phase, we may not use the above accelerated processing since the number of selected candidates is much fewer than that of the original data.

In fact, Rule 2 is still weak, and we cannot strictly identify the nose tip. However, there are much more candidates near the nose tip as illustrated in Fig. 2. In Phase 3, we can consider that the position having the most dense candidates is the nose tip. For each candidate, Q_i, we can count the number of the candidates within a local sphere centered at Q_i. Then, three points with the highest number of candidates are selected, and the mean of their positions is considered as the nose tip.

In Phase 1, the neighboring area can be small in order to lower the computational cost; and in Phase 2, the neighboring area should be larger to conquer the influence of noise. This proposed method for nose tip location only deals with scalar features so that it is immune to geometric transformation. Also, the features are the statistics of the local points that provide a robust way to conquer noise and different resolutions.

3. Nose ridge detection

After identifying the nose tip position, we use the Included Angle Curve (IAC) to determine the nose ridge. It makes sense to estimate the head pose.

First, the Included Angle Curve (IAC) is described in the following. For the detected nose tip, P, we place two spheres with radius R and $r (R > r)$, centered at P as shown in Fig. 4. The intersection of two spheres and the facial surface generates a torus in 3D space. We define a reference plane, G, by the normal vector, N, of the nose tip and a reference vector, C, orthogonal to N. A plane, Q, rotates along the axis, N, beginning from G with a fixed step, $\Delta\theta$. Thus, the rotated plane, Q, divides the torus into $360/\Delta\theta$ partitions. We obtain the barycenter, B_i, of ith partition by averaging the points within it and then calculate the included angle between $B_i - P$ and N. We can use one curve to describe the variation of these included angles, which is called the Included Angle Curve (IAC).

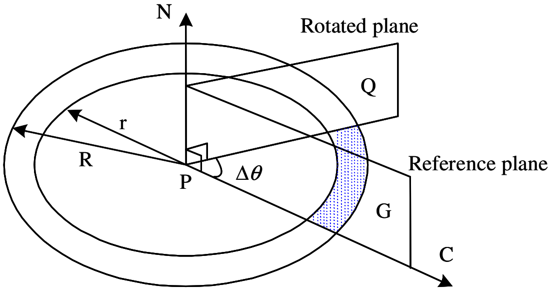

Fig. 4. Partition of the torus.

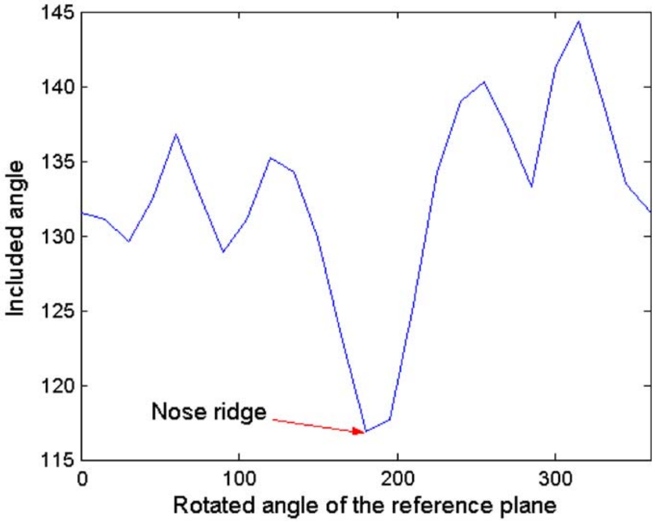

Fig. 5. The Included Angle Curve.

Since P is the local highest point according to our hierarchical filtering scheme, all its included angles are larger than 90°. The area near the nose ridge has the smallest included angle as shown in Fig. 5, which can be used to determine the approximate direction of the nose ridge.

This algorithm is similar to point signatures (Chua and Jarvis, 1997). The main difference is that our method does not acquire a plane fit to the local points. It largely reduces the computational cost. In our algorithm, the radius R and r are the principle parameter for a successful implementation. Appropriate values can avoid the influence of the nosewing and make the nose ridge stand out. They may be different in different databases. In a database in which the data have the real size, the results are satisfactory when $R = 25$ mm and $r = 20$ mm. In a database in which the data are zoomed in or out, the results are satisfactory when r is set to the distance between two inner eye corners and R is 0.25 times larger than r. The data in a database usually have the same scale. Thus, R and r should be preset manually. $\Delta\theta$ is another important parameter. The smaller it is, the smoother the IAC is, and the precision of estimating the nose ridge is higher. However, if $\Delta\theta$ is too small, some partitions will include no points. In this paper, we set $\Delta\theta = 15°$.

In addition, the original point cloud from the laser scanner usually contains holes, which result in no points in some partitions. During implementation, we can ignore these partitions since the nose ridge usually has enough data due to its distinct protrusion.

4. Experimental results

4.1. Databases

We use three different databases to test our proposed algorithm. The first database (3D Pose and Expression Face Models, 3DPEF) is collected using a Minolta VIVID

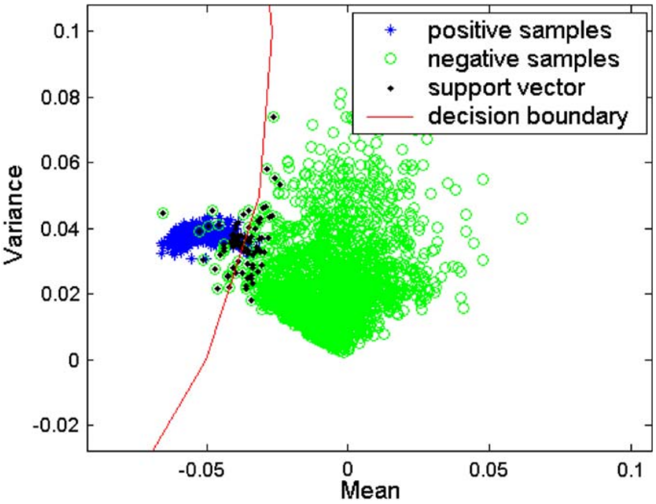

Fig. 6. The training result of SVM.

900 working on Fast Mode. This data set contains 300 3D point clouds from 30 persons (six are women), and each person has 10 point clouds, including five poses (front, tilting up or down, and rotating left or right) and five expressions (smile, laugh, anger, surprise and eyes closed). This set is used to train SVM classifiers and test the robustness of our method. The second database (MPI)[1] is from MPI in Saarbrücken. Unfortunately, we obtained only five 3D head models from them. These models are used to test the performance with high-resolution data. The third data set (PART) is from the 3D Part Database[2] of the IRIS Laboratory at the University of Tennessee, Knoxville. It is used to test the performance under different resolutions. In advance, all the point clouds are triangularized and the normals of all the points are also calculated.

4.2. Training for SVM

During the training process, we select 20 samples from two persons (one man and one woman) from 3DPEF as the training set. Due to the limitation of the capturing equipment and also the variations in pose and expression, the data contain some holes and outliers. To ensure that the positive samples in the training set can describe the correct shape of the nose tip, we first fill the holes and remove the outliers manually. We select the points around the nose tip in the 20 training point clouds and record their means and variances as positive samples. The negative samples can be arbitrarily obtained from any other area from the original point clouds. Finally, we obtain 325 positive samples and 3178 negative samples for training the SVM classifier. By using the algorithm in Section 2.2, we can determine one optimal boundary. Fig. 6 shows the training results.

From Fig. 6, we can see that the positive samples congregate tightly. This means that human noses have a similar statistical shape. The negative samples are distributed in a large space. We also see that the decision boundary does not strictly distinguish the two classes, but it can effectively select the candidates of the nose tip.

4.3. Nose tip location

During the testing process, we first locate the nose tip in the 3DPEF database. The total number of tested samples in 3DPEF is 280 of 28 persons, excluding the training samples (20 samples of two persons) for SVM. Under the complex condition of pose and expression variations, the unwanted influence of clothes, and holes and outliers, only two samples fail and the correct detection rate is up to 99.3%. Fig. 7 shows some detected results.

The first row in Fig. 7 shows pose variations; the second shows expression variations. In Fig. 7(c) and (d), the nose tip is not the highest point due to up or down motion. Our method can detect the nose tip successfully. In Fig. 7(b), (c) and (g) the sampled data contain some clothing. In Fig. 7(g), there is much noise generated due to long hair. Fortunately, our method can overcome these challenges effectively. Intuitively, the jaw and the protruding throat in men have similar statistical features to the nose. Our proposed scheme can distinguish the nose tip from these features effectively.

The samples that are falsely detected are affected by clothing. When the clothing is deformed into a certain shape, it may have similar statistical features to the nose, which induces incorrect detection. Fig. 7(h) shows one example. Here, the neckline also collects more candidates, which creates a puzzle on the nose tip position.

The MPI set[1] includes high quality 3D data, which have dense points (more than 70,000 vertices, 150,000 triangles) and smooth surfaces. Our proposed method can correctly locate the nose tip and nose ridge as shown in Fig. 8. However, due to too many local highest points, it is difficult to reduce the number of candidates, thus resulting in a large computational load. In fact, we can first reduce the number of vertices in the model by mesh optimization (Hoppe et al., 1993) and then locate the nose.

Finally, we also test our algorithm with the PART data set,[2] which provides three complete 3D head models of a status as shown in Fig. 9. In these models, the search space is the whole head surface, and the models lie horizontally in three-dimensional space. The nose tip can be located successfully in different resolutions. Unfortunately, in Fig. 9a, the points are too sparse and the nose ridge is not marked correctly.

4.4. Discussion

Nose detection is usually a preliminary phase of a larger task. It requires a reasonably low computational cost. The computational cost of our scheme for detecting the nose tip

[1] http://faces.kyb.tuebingen.mpg.de
[2] http://iristown.engr.utk.edu/%7Epage/database

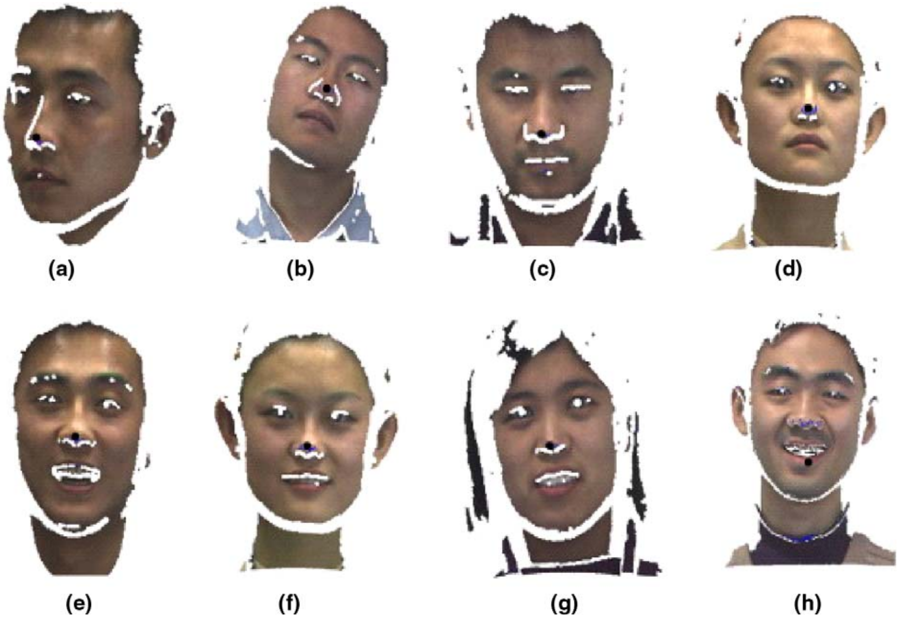

Fig. 7. Results from 3DPEF. The black point is the detected nose tip.

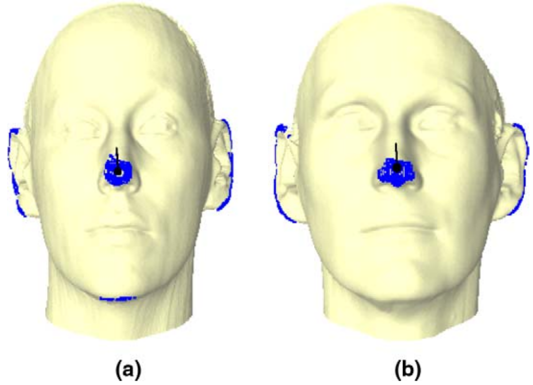

Fig. 8. Results from the MPI set. The black line connecting to the black point is the detected nose ridge.

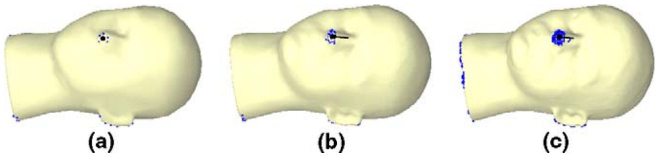

Fig. 9. Results from the PART set. (a) 502 vertices; (b) 2502 vertices; (c) 40,071 vertices.

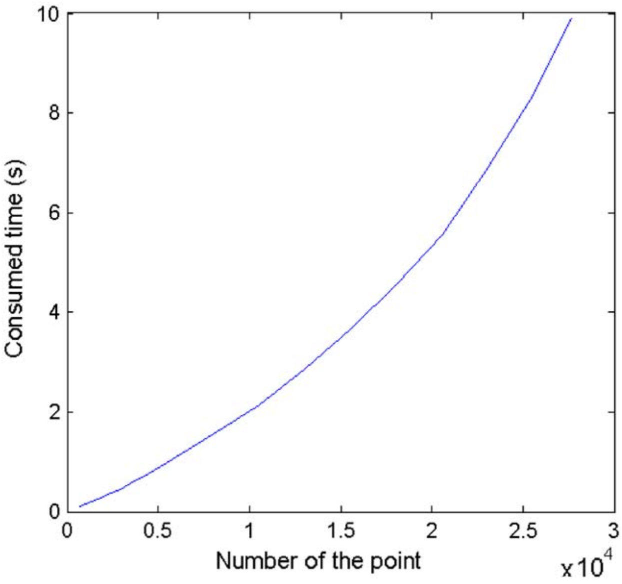

Fig. 10. Consumed time at different resolutions.

strongly depends on the number of points. Fig. 10 shows the consumed time for nose tip detection in one sample with different numbers of points in 3DPEF set. We can see that the consumed time increases quickly with an increase in the number of points. Fortunately, three thousand points are usually enough to describe facial details. It takes only 0.4 s to locate the nose tip in this kind of data. This experiment was executed on a PC with a PIV 1.3 GHz processor, 128M RAM and a Nvidia Getforce2 MX 100/200 display card.

We usually consider the nose tip as a region, not an exact point. In our scheme, the area containing the densest candidates in the final step shows the scope of the nose tip. In order to represent it simply, we still give one point to indicate the nose tip. It should be noted that the detected nose tips in the different point clouds are not exactly in the same physical position. However, it is enough to initialize the clouds for subsequent work, such as registration or locating other facial features.

Both local features based on the effective energy are independent of the rotation and transformation. Thus, our algorithm for nose tip location is robust to pose variations. Moreover, using statistical features (the mean and variance in this paper) is an efficient way to overcome resolution variations and noise. Due to the weak classification, neither of these features exactly represents the characteristics of the nose tip. So, it is necessary to build a hierarchical filtering scheme to make the final choice.

5. Conclusions

In this paper, we have proposed a novel hierarchical filtering scheme combining two local features to locate the nose tip in 3D facial data. Then, the Included Angle Curve (IAC) is defined to estimate the nose ridge. The proposed method is robust to noisy and incomplete input data, invariant to rotation and translation and suitable to multiple resolutions. The experimental results with different databases have shown its excellent performance. In the future, we will locate other features based on the detected nose, and it will also be necessary to consider other statistical features for detection.

Acknowledgements

The authors gratefully thank Mr. Z.P. Jiao for helping us to collect 3DPEF data and Prof. H.H. Buelthoff for providing the MPI head models. We also thank Dr. J.L. Cui for discussion on SVM.

References

Beumier, C., Acheroy, M., 2000. Automatic 3D Face Authentication, vol. 18. IVC, pp. 315–321.

Burges, C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining Knowledge Discovery 2 (2), 121–167.

Chua, C.S., Jarvis, R., 1997. Point Signatures: A New Representation for 3D Object Recognition. IJCV, 25(1), 63–85.

Gordon, G.G., 1991. Face recognition based on depth maps and surface curvature. In: Proc. SPIE Geometric Methods in Computer Vision, vol. 1570.

Hesher, C., Srivastava, A., Erlebacher, G., 2002. A novel technique for face recognition using range imaging. Internat. Multiconference Comput. Sci.

Hoppe, H., Derose, T., Duchamp, T., Mcdonald, J., Stuetzle, W., 1993. Mesh optimization. In: ACM SIGGRAPH Proceeding, pp. 19–26.

Joachims, T., 1999. Making Large-Scale SVM Learning Practical. Advances in Kernel Methods—Support Vector Learning. MIT Press (Chapter 11).

Kim, T.K., Kee, S.C., Kim, S.R., 2001. Real-time normalization and feature extraction of 3D face data using curvature characteristics. In: IEEE Internat. Workshop on Robot and Human Interactive Communication, pp. 74–79.

Lee, Y.C., Terzopoulos, D., Waters, K., 1993. Constructing physics-based facial models of individuals. Proc. Graphics Interf., 1–8.

Lee, Y., Park, K., Shim, J., Yi, T., 2003. 3D face recognition using statistical multiple features for the local depth information. In: Proc. 16th Internat. Conf. Vision Interf.

Xu, C., Wang, Y., Tan, T., Quan, L., 2004. Robust nose detection in 3D facial data using local characteristics. In: Proc. ICIP'04 1995–1998.