Pattern Recognition 44 (2011) 544–558

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A training-free nose tip detection method from face range images

Xiaoming Peng [a,b,*], Mohammed Bennamoun [b], Ajmal S. Mian [b]

[a] School of Automation Engineering, University of Electronic Science and Technology of China, No. 4, Section 2, North Jianshe Road, Chengdu, Sichuan 610054, China
[b] School of Computer Science and Software Engineering, The University of Western Australia, M002, 35 Stirling Highway, Crawley, WA 6009, Australia

ARTICLE INFO

Article history:
Received 22 June 2010
Received in revised form
13 August 2010
Accepted 21 September 2010

Keywords:
Nose tip detection
Range images
Biometrics

ABSTRACT

Nose tip detection in range images is a specific facial feature detection problem that is highly important for 3D face recognition. In this paper, we propose a nose tip detection method that has the following three characteristics. First, it does not require training and does not rely on any particular model. Second, it can deal with both frontal and non-frontal poses. Finally, it is quite fast, requiring only seconds to process an image of 100–200 pixels (in both x and y dimensions) with a MATLAB implementation. A complexity analysis shows that most of the computations involved in the proposed algorithm are simple. Thus, if implemented in hardware (such as a GPU implementation), the proposed method should be able to work in real time. We tested the proposed method extensively on synthetic image data rendered by a 3D head model and real data using FRGC v2.0 data set. Experimental results show that the proposed method is robust to many scenarios that are encountered in common face recognition applications (e.g., surveillance). A high detection rate of 99.43% was obtained on FRGC v2.0 data set. Furthermore, the proposed method can be used to coarsely estimate the roll, yaw, and pitch angles of the face pose.

1. Introduction

Face recognition is an active research area in biometrics with much of the work being performed in the 2D domain. With the rapid development of advanced range imaging devices, face recognition based on range data has lately received growing attention from researchers and industries. Compared with 2D face recognition, 3D range-based face recognition that uses range images is less sensitive to illumination, pose variations, facial expressions, and makeup [1]. In 3D face recognition methods, the nose region plays a very important role in face normalization [1,7], pose correction [2–4], and nose region based matching [5–7]. As a result, nose localization has become an important (if not necessary in some cases) preliminary step in 3D face recognition tasks. More often, nose localization is achieved through the detection of distinctive facial features, e.g., the nose tip, nose ridge, and eye corners in face scans. Below, we provide a list of the most relevant references, analyze their techniques and discuss their limitations. We will finally summarize the main characteristics of these techniques.

Mian et al. [1] propose a nose tip detection method on horizontally sliced image contours. For each slice contour a triangle of maximum altitude is found using a circle whose center travels along the slice contour. The vertices of the triangle are all on the slice contour with one vertex coinciding with the center of the circle and the other two vertices being the two intersections of the circle with the slice contour. The nose tip candidate for this slice contour is regarded as the center of the circle. Random Sample Consensus (RANSAC) is used to further remove outliers. Of the remaining nose tip candidates, the one that has the maximum confidence is taken as the nose tip. Since this method relies on the triangle altitude for the determination of the nose tip, it is likely to fail in some poses, e.g., pure profile facial views. Koudelka et al. [6] observe that the nose and eyes are radially symmetric and can be detected using the radial symmetry transform [29]. However, their observation is only valid when the face is frontal and almost upright. Colbry et al. [19] propose to use two different strategies for nose tip detection under frontal and non-frontal poses. Under a frontal pose, they take the nose tip as the point of the face scan that is closest to the viewer. Under non-frontal poses, they use up to six criteria to find the nose tip. They, however, do not provide a mechanism that distinguishes between a frontal face from a non-frontal one. In addition, they did not justify the use of their six criteria. The detection rates for profile views under neutral expressions and smiles are only 82.7% and 75.0%, respectively.

Some researchers use curvature analysis, including the mean (H) and Gaussian (K) curvatures classification (HK Classification) [4,10,15,21] and the principal curvatures[1] [4,16,17] to help confine the search scope of particular facial features. In HK

* Corresponding author at: School of Automation Engineering, University of Electronic Science and Technology of China, No. 4, Section 2, North Jianshe Road, Chengdu, Sichuan 610054, China. Tel.: +86 28 61830304.
E-mail address: pengxm@uestc.edu.cn (X. Peng).

[1] The mean curvature H and Gaussian curvature K used in HK Classification are closely related to the principal curvatures λ_1 and λ_2 by the equation $\lambda^2+2H\lambda+K=0$.

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.09.015

Classification based methods, an input image is first divided into up to eight different regions according to the signs of the mean curvature H and Gaussian curvature K, and the search of particular facial features is confined in thresholded curvature maps. Colombo et al. [10] search for the nose region in the positive regions of the thresholded mean curvature map and the eyes in the thresholded HK-classified map considering only the elliptical concave regions. Similarly, Chang et al. [5] set two thresholds each for the two curvature maps to search for the nose tip, the two eyes, and the nose ridge. Segundo et al. [15] first use HK Classification to find the regions in which the nose tip and base reside, and then utilize the so-called "profile curve" and "median curve" to refine the nose tip and corners. Szeptycki et al. [21] first use HK Classification to coarsely locate the nose tip and inner corners of the eyes, and then refine their locations by aligning the input image to a face model with manually labeled fiducials on it. Analogous to HK Classification, principal curvatures can also be used to divide an input range image into up to eight different regions according to the signs of the principal curvatures. When searching for the nose tip Malassiotis and Strintzis [4] consider only those candidate points whose principal curvature values are relatively large. Similarly, Sun and Yin [16,17] use principal curvatures to locate the eyes and the nose tip. There are two pitfalls with curvature analysis based methods. First, the computation of the H and K curvatures and the principal curvatures involves estimating the second order derivatives of a range image, which is error-prone because of the noise in the image. Surface approximation methods can only partly alleviate this problem [25]. Second, these methods are not pose invariant. For example, in frontal or near frontal face images, the nose tip usually does locate in a "peak" region in the curvature maps, but this is not the case for non-frontal face images.

In [2,19], the authors try to detect a triplet formed by the so-called "core points", namely, the inner and outer corners of the two eyes, the nose tip, the chin tip, and the corners and middle of the mouth. To this end they first use shape index [14] to divide a test face scan into different regions and from these regions they select three vertices that can form a triplet. For example, the inner eye corner candidates are selected from the region where the shape index value is close to zero, the nose tip candidates are supposed to be located in the region where the shape index value is between 0.7 and 1. Obviously, numerous possible sets of triplets can exist and many of them are invalid. Finding a correct set of triplet could be time-consuming—although the authors use a relaxation labeling strategy to avoid an exhaustive search, the time for this step was not reported in their papers.

Xu et al. [9] propose a hierarchical filtering scheme for nose tip detection. They first use the "effective energy" defined in their paper to filter out points in non-convex areas. Only those points whose effective energy sets contain only negative elements can pass this phase. However, the effective energy condition is very weak and many points in other areas like the cheeks and chin also meet this condition. To distinguish nose tips from these points they then train a support vector machine (SVM) using the means and variances of the effective energy sets as input to further filter out other non-nose tip points. Although a high detection rate of 99.3% on a small dataset containing 280 test face scans under frontal or near frontal poses was reported in their paper, their SVM-training based method is problematic when applied to various poses because the effective energy set is not pose invariant—as a matter of fact, even for a same nose tip, the means and variances of the various effective energy sets corresponding to substantially different views (e.g., a frontal view and a profile view) are dramatically different. Chew et al. [18] also use the effective energy condition to select nose tip candidates, but circumvent the SVM-training stage by trying to find the mouth-and-eyes regions in the input image, which is not always easy. The nose tip detection rates were moderate as reported in their paper—93% and 68%—for frontal and non-frontal faces, respectively. In Pears et al.'s method [7] a nose tip candidate has to pass a four-level filtering scheme at the 3D vertex level in order to be identified. Their method is computationally expensive[2] and cannot deal with pure profile facial views.

Slater et al. [24] propose to represent a point using a feature vector containing integral invariants of multiple scales, and then adopt the quadratic discriminant analysis to classify the feature vectors of nose tips and non-nose tips. They reported a 98.08% successful detection rate on 4007 images of FRGC v2.0 data set. However, the integral invariants in their paper are only invariant to the Euclidean group but not invariant to various poses; thus this method will suffer the same problem in the training process as that suffered by Xu et al.'s method [9]. The same problem is encountered by Wang et al.'s method [8] which uses the point signature representation [13] combined with the Principal Component Analysis (PCA) to detect four facial features. However, the point signature representation is also not pose-invariant. Conde et al. [11] first select three candidate areas for the nose tip and inner corners of the eyes using the mean curvature and clustering, then use the spin image representation [12] in conjunction with three SVMs (each for one of the three facial features) to choose from these three candidate areas three feature points corresponding to the nose tip and inner corners of the eyes. Since the spin image representation is only invariant to rigid transformations but not invariant to various poses, this method works well only in frontal or near-frontal conditions.

Breitenstein et al. [3] propose to represent each pixel in a range image using single signatures. A single signature is a set of yaw (the rotation around the vertical axis) and pitch (the rotation around the horizontal axis) orientations and can be represented by a sparse Boolean matrix. Single signatures can form into aggregated signatures that can be used to generate a set of nose candidates and head pose hypotheses. For a pixel to be a nose candidate its aggregated signature must have more local directional maxima than a provided threshhold and the center of the set of the local directional maxima has to be part of the pixel's single signature. A nose tip and the associated pose is validated by finding the best match between the input range image and a set of reference pose range images whose orientations are close to that of the pose associated with the nose tip candidate. The reference pose range images are rendered by rotating a 3D head model under many poses. The 3D head model was generated from the mean of an eigenvalue decomposition of laser scans of 97 male adults and 41 female adults. However, in some practical applications a 3D head model such as this one may not always be available. More importantly, we argue that a particular model generated from a quite limited number of training data is unlikely to accurately represent any input face. For this reason, the method described in [20] that locates nose by finding the best matches to two particular nose templates in an input image is likely to suffer from the similar problem. Whitmarsh et al. [23] propose to detect facial landmarks by registering a flexible facial model (CANDIDE-3 model) to the scan data. However, in order to find a reasonable starting point for the registration, it requires an

[2] As reported in [7] (Section 5.5) the average time needed to identify a nose tip on a range image whose size is approximately one fourth (in two directions) of the size of the FRGC data is the sum of the following processing times: (i) normals and DLP (distance to local plane) descriptors (10 mm radius neighborhood): 4.8 s; (ii) RBF (radial basis function) model fitting: 12.1 s; (iii) SSR (spherically sampled RBF) values (128 spherical samples): 40.7 s; (iv) SSR value local maxima 0.0003 s; and (v) SSR histogram generation (4096 spherical samples) and comparison: 6.5 s. The total of these five items is approximately 64.1 s. The times were obtained from a PC with the following specification: AMD Athlon 64×2 Dual core 4200+2.20 GHz, 4 Gb RAM, running Windows XP and MATLAB R2006a.

input face scan which is oriented upwards and a nose tip that corresponds to the smallest z-value—a requirement that is not always met in practice.

In contrast to the majority of the literature on 3D facial feature detection, there are only few reports on doing the job on 2D face profiles.[3] Lu and Jain [26] assume that the nose tip has the largest z value (called "directional maximum" in their paper) if projected onto the corrected pose direction. Starting with this assumption, they rotate a face scan along the y-axis with a given step size, recording the point on the face scan that receives the most number of directional maxima during the process. This point is regarded as the nose tip. In the case that multiple nose tip candidates exist, they choose the one with the smallest distance-from-feature-space (DFFS) metric as the nose tip. The drawback of their method is that it does not take the pitch (the rotation around the x-axis) into consideration. Faltemier et al. [27] propose to first get 37 profiles by rotating a face scan around the y-axis from the range of [0°, 180°] in a step size of 5°. Then, for each profile they search for the nose tip by translating two nose profile models along the profile to search for the points that best match the nose tips of the two models, respectively. If the distance between the two searched points is within some threshold, the nose tip of the profile is regarded as the midpoint of the two points; otherwise, the nose tip is taken as one of the two points. Finally, for each quadruple of four consecutive profiles the authors sum the four matching errors of each nose tip and the quadruple with the smallest sum of matching errors is kept. The nose tip that has the smallest matching error in this quadruple is regarded as the detected nose tip. This translation-based model matching is obviously scale sensitive. In addition, the authors provide no details of the two nose profile models other than that they were "taken from the nose of an image with a 0° pitch" and "from the nose of an image with a 45° pitch." In addition, just as in the case of 3D model, the generalization of their model is limited. We will use an example to justify this in the appendix.

From the above, we can conclude that there are three main pertinent characteristics to current nose tip detection methods:

(1) *Firstly, most existing methods [2–5,7,9–12,15–24] are 3D-based*: They rely on the direct detection of facial features from range images. These methods are largely not pose invariant, relatively slow, and sometimes need to estimate the second order derivatives of a range image when computing the H and K curvatures and the principal curvatures, which is error-prone because of the noise in the image.
(2) *Secondly, many existing methods are training based [7–11,22,24,28] or model based [3,20,21,23,27]*: Training is laborious, and only works well for limited poses, as has been reported in [30]. Furthermore, many training-based methods work well on frontal poses and are not suitable for profile views. Model-based methods are likely to be biased towards a particular model's limited generality.
(3) *Finally, most existing methods [1,4–6,8–11,15–17,20–24,28] deal mainly with frontal or near-frontal poses*:

In this paper we propose a nose detection method that has the following characteristics. First, it does not require any training nor rely on any particular model. Thus, compared with training-based and model-based methods, it has the advantage of being exempt from laborious training and pose limitations associated with training. It inherently exhibits a better generalization. Second, it does not require the distinction between frontal and non-frontal poses. As demonstrated in Section 3, it is robust to many scenarios that are encountered in common face recognition applications. Finally, it is relatively fast. It takes seconds for the proposed method to detect a nose tip detection on an image of 100–200 pixels (in both x and y dimensions) with a MATLAB implementation.

The rest of the paper is organized as follows. In Section 2, we describe the proposed method in detail. Section 3 presents the experimental results of the proposed method. Finally, in Section 4 we provide the conclusions and discussions.

2. Method description

The proposed method works on 2D face profiles. In the following we describe in order: (1) the generation of 2D Left-and-Right-Most face profiles, (2) the detection of nose tip candidates, and (3) the identification of the nose tip.

The input of the method is a face range image or a uniformly sampled point cloud of a face. Without loss of generality, we set up the coordinate system for the input image as shown in Fig. 1. The directions of the rotations in Fig. 1(a) are as follows: The pitch is the rotation about the x-axis, the yaw is the rotation about the y-axis, and the roll is the rotation about the z-axis. The origin "o" of the coordinate system lies on the xy-plane of the face scan, positioned at the midpoint of the bottom row Fig. 1(b). We further assume that the smallest z-value of the face points on the scan is zero.[4] Let R and C denote the number of rows and columns of the face scan, respectively.

2.1. Generation of 2D face profiles

Assume that there are N face points in an input face scan. Rotated around the y-axis by an angle of β, a face point (x_i,y_i,z_i) $(i=1,2,\ldots,N)$ in the original input face scan takes the new 3D coordinates $(x_i^\beta,y_i^\beta,z_i^\beta)$ as

$$\begin{pmatrix} x_i^\beta \\ y_i^\beta \\ z_i^\beta \end{pmatrix} = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}. \quad (1)$$

Note that x_i and y_i take integer values and that rotations around the y-axis do not change the y-coordinates, i.e., $y_i^\beta = y_i$.

Rotating the input face scan within the range of $[-90°, 90°]$ in a step size of 3°, we get 61 new point sets, each consisting of the rotated valid points of the original input face scan by a particular angle in the range $[-90°, 90°]$. We empirically chose a step size of 3° to compromise between accuracy and efficiency. We then divide the 61 new point sets into two groups and we call one group the Min Group and the other the Max Group. The Min Group consists of 30 point sets corresponding to the rotations in the range of $[-90°, -3°]$ and the Max Group contains the remaining 31 point sets corresponding to the rotations in the range of $[0°, 90°]$. For a point set that corresponds to rotation β_j $(j=1,2,\ldots,30)$ in the Min Group its 2D left most face profile S_j is a 2D curve on the xy-plane:

$$S_j = \{(x_k^j,y_k^j)_{k=1,2,\ldots,N_j} \mid y_k^j \in \{0,1,\ldots,R-1\},$$
$$x_k^j = \min(\{x_i^{\beta_j} \mid (x_i^{\beta_j},y_i^{\beta_j}) = y_k^j,z_i^{\beta_j})_{i\in\{1,2,\ldots,N\}}\})\}, \quad (2)$$

where N_j is the total of 2D points in S_j. It is obvious that S_j is the *left most* boundary of the region formed by projecting the points

[3] The word "profile" has different meanings in this paper. A "profile view" of a face is used to distinguish from a frontal view of the face; while the "2D face profiles" are the 2D contours of a face projected onto a 2D plane.

[4] This can be obtained by subtracting the smallest z-value of all the face points from the z-values of all the face points.

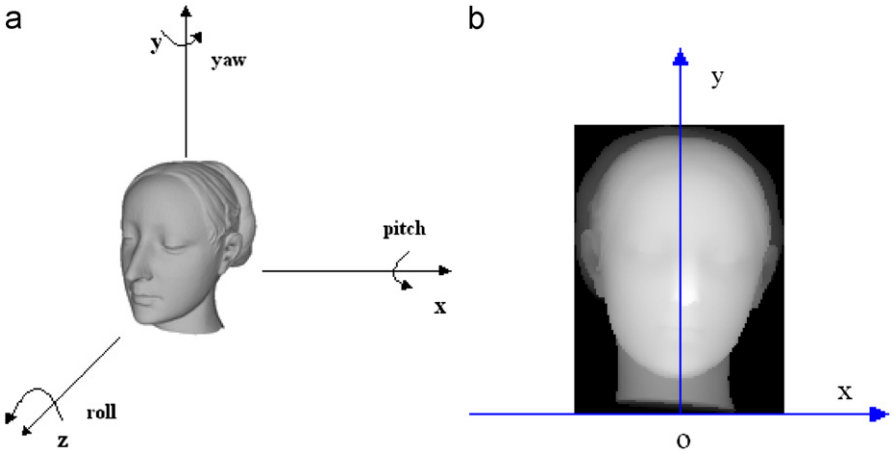

Fig. 1. The coordinate system of an input range image: (a) 3D coordinate system. Larger z-value points are closer to the viewer and (b) the x–o–y plane.

$(x_i^{\beta_j}, y_i^{\beta_j}, z_i^{\beta_j})_{i=1,2,\ldots,N}$ onto the xy-plane. We round x_k^j to its nearest integer value in the case that x_k^j is non-integer.

Similarly, we define the 2D right most face profile S_j ($j=31,32,\ldots,61$) for a point set in the Max Group as

$$S_j = \{(x_k^j, y_k^j)_{k=1,2,\ldots,N_j} \mid y_k^j \in \{0,1,\ldots,R-1\},$$
$$x_k^j = \max(\{x_i^{\beta_j} \mid (x_i^{\beta_j}, y_i^{\beta_j} = y_k^j, z_i^{\beta_j})_{i \in \{1,2,\ldots,N\}}\})\}. \quad (3)$$

S_j is the *right most* boundary of the region formed by projecting the points $(x_i^{\beta_j}, y_i^{\beta_j}, z_i^{\beta_j})_{i=1,2,\ldots,N}$ onto the xy-plane.

Examples of 2D face profiles can be seen in Fig. 3 (1st column).

2.2. Detection of nose tip candidates

After the generation of 61 2D Left-and-Right-Most face profiles S_j ($j=1,2,\ldots,61$), the next step is to detect possible nose tip candidates. We illustrate this step on a Max Group right most face profile in Fig. 2. The nose tip detection on a Min Group left most face profile is similar. We move the center of a circle of radius r along the face profile.[5] As can be seen in Fig. 2, at different positions on the face profile the areas of the circle enveloped "inside" the face profile (shown in blue) are different. At protruding positions, such as the nose tip, this area is relatively small; at sunken positions, such as the nose ridge, this area is relatively large; at flat positions, this area is between the above two extremes. We use a simple metric to describe these differences.

We uniformly sample the circle at 180 points along its perimeter. Let n^+ and n^- to denote, respectively, the numbers of the 180 points inside and outside the face profile when the circle's center moves to a particular point on the face profile. The difference of n^+ and n^- is used as a metric to estimate the difference between the areas of the circle inside and outside the face profile at this particular point:

$$D = n^+ - n^-. \quad (4)$$

For a point on the face profile to be a possible nose tip candidate, we require that $D \leq -50$. This threshold is experimentally obtained and works very well in many cases. The metric is similar to that used in [7] for estimating the SSR (spherically sampled RBF) values. However, in [7] the computation is carried out on a sphere surface.

[5] We uniformly resampled all input face scans used in the experiments of this paper to a resolution of 1 mm/pixel. Through this paper we use $r=10$ pixels which works very well in our experiments.

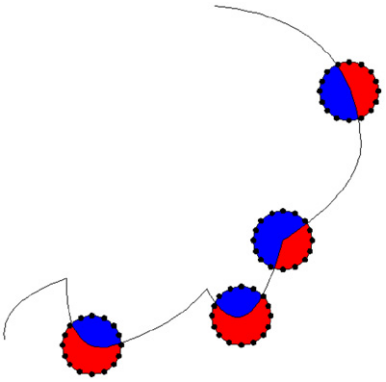

Fig. 2. Nose tip detection on a Max Group right most face profile. The center of the circle is moved along the Max Group face profile. At different positions on the face profile the areas of the circle enveloped "inside" the face profile (shown in blue) are different. Smaller areas are likely to correspond to the nose tip (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Examples of possible nose tip candidates (shown by red crosses) on 2D Left-and-Right-Most face profiles can be seen in Fig. 3 (1st column). One can see three interesting facts from Fig. 3. First, besides the real nose tip candidates some outliers are also detected, such as the points on the chin. Second, for some poses, like the pure profile views in the second and third rows, the real nose tip candidates exist on only one group (either the Min Group or the Max Group) of face profiles. Third, the real nose tip candidates cluster into narrow regions with small dimensions in the y direction, which is most obviously seen in the first and last rows. The reason for this fact is that rotations around the y-axis do not change the y-coordinates, thus the real nose tip candidates should have similar y-coordinates.

In view of this fact, we use a simple method to locate the y-coordinates of nose tip candidate clusters after the detection of possible nose tip candidates on all the 2D profiles as follows:

(1) Initialize a histogram with each entry of the histogram corresponding to each y-coordinate value of the 2D profiles.
(2) For each point (x_k^j, y_k^j) [$k=1,2,\ldots,N_j$, where N_j is the total of points in S_j. See Eqs. (2) and (3)] at where a possible nose tip candidate is detected [using Eq. (4) and the -50 threshold], increase the entry corresponding to y_k^j by one.

Examples of the distribution histograms of possible nose tip candidates at the vertical direction (y-axis) of four face scans are provided in Fig. 3 (the second column). It can be seen from the

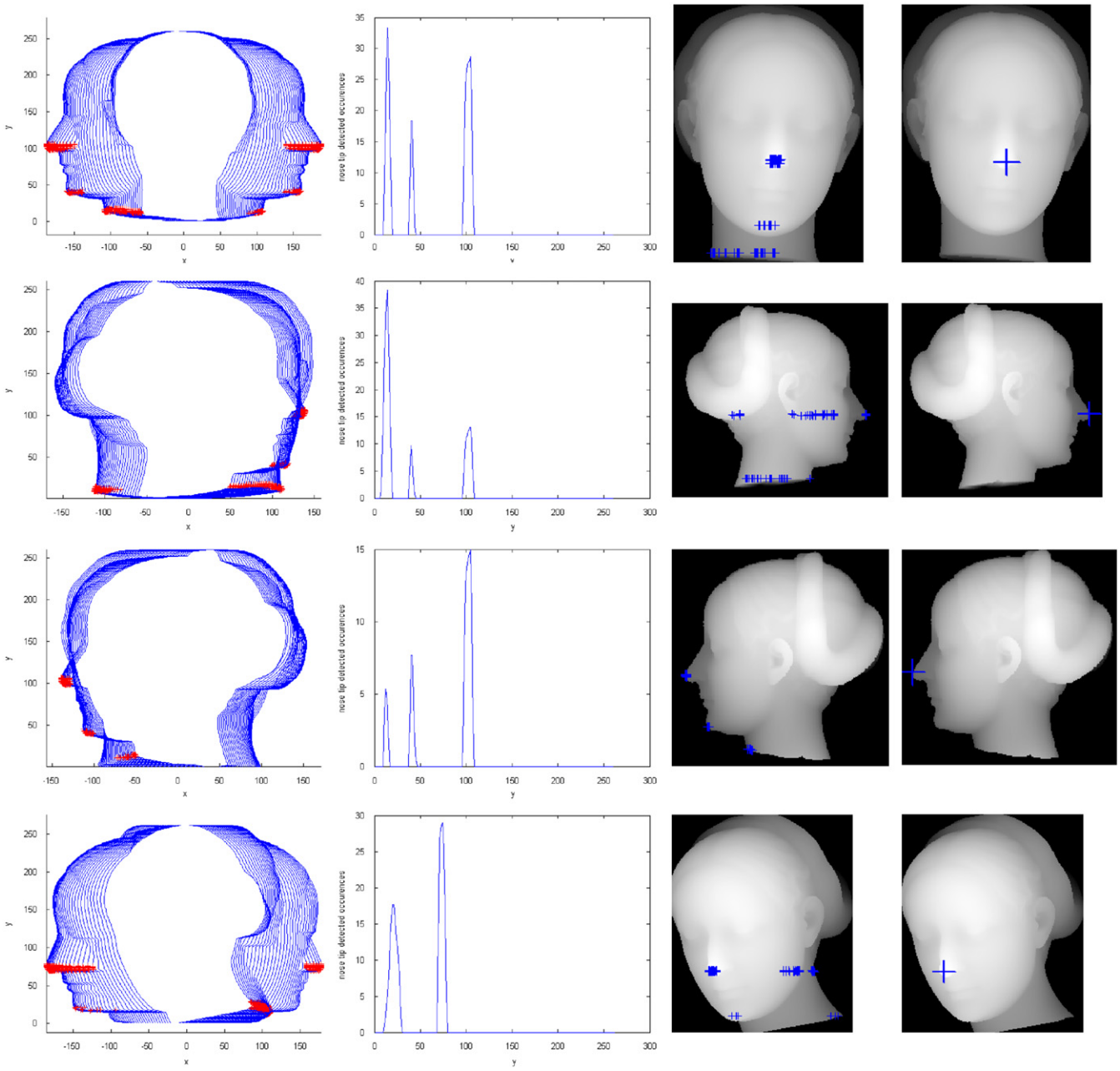

Fig. 3. Nose tip detection process. The 1st column: 2D Left-and-Right-Most face profiles and possible nose tip candidates (shown by red crosses). The 2nd column: the distribution histograms of possible nose tip candidates at the vertical direction (y-axis) of individual face scans. The 3rd column: re-projections of cardinal nose tip candidates onto individual face scans. The 4th column: detected nose tips after identification (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

figure that the nose tip candidate clusters correspond to spikes in the histograms. Also, the highest spike may not correspond to the real nose tip candidates cluster. Since the nose tip is always the point that is most detected in its local neighborhood, thus in each nose tip candidates cluster we only consider those points whose y-coordinates correspond to the peak value of the related spike. We call these points the "*cardinal points*" in this paper. In the third column of Fig. 3, we provide examples of cardinal points re-projected onto individual face scans.[6] We abandon all short spikes whose heights are either less than 30% of the highest spike or less than five to avoid false detections due to noise. Our next step is to identify from these many points the nose tip.

2.3. Identification of the nose tip

To identify the nose tip, we calculate two metrics, the *cardinal point fitness* and the *spike fitness* described below.

We observed that in a large number of cases the nose shape in a face profile can be fitted using a triangle. Based on this observation we first compute the degree of similarity between a triangle and a face profile segment corresponding to a nose shape candidate. An example is provided in Fig. 4 in which the point S is

[6] We use a lookup table for this purpose. When generating the 2D face profiles, we construct a lookup table and store in it the 3D coordinates corresponding to the (projected) 2D points of the face profiles. In this way, when a nose tip candidate is detected on a face profile, its position on the face scan is easily retrieved.

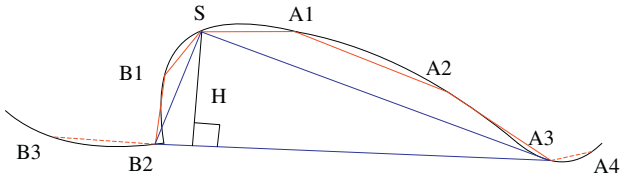

Fig. 4. The fitness of the cardinal point S is calculated as $2-S_p/S_t$, where S_p is the area of the polygon (B2, B1, S, A1, A2, and A3) and S_t is the area of the triangle (B2, S, and A3).

a cardinal point.[7] Starting from S we fit the face profile segment around it using connected straight line segments. For a fit to be valid we require that the connected straight line segments must form a maximum convex polygon. For example, the polygon (B2, B1, S, A1, A2, and A3) is valid and adding more points, such as B3 and A4, will violate the requirement. In our implementation we use a line segment to approximate a curve segment if all the points in the curve segment are within distance d of the line segment. We empirically found that optimal values for d are two or three pixels. Larger values are likely to lead to an over-fitting of the nose shape candidate. After the polygon fitting operation, we also extract a triangle by connecting S with the two polygon vertices that are the farthest from it on the two sides of the polygon, B2 and A3. Let S_p and S_t denote the areas of the polygon and the triangle, respectively, the fitness of the cardinal point S is calculated as

$$CF_S = 2-\frac{S_p}{S_t}. \tag{5}$$

This metric has a positive polarity and is equal to or less than one. Obviously, a large CF_s value indicates that the nose shape candidate with S as the nose tip is similar to a triangle.

Although a larger value of CF_s means that the face profile segment is closer to a triangle in shape, it does not necessarily mean that the segment represents a nose. For example, assume in Fig. 4 that S is the nose tip, A3 is the nose ridge, and B2 is the nose bottom, then we must have $|\overline{A3S}| > |\overline{B2S}|$ in order for the segment to correspond to a possible nose shape. In view of this fact we use the detected nose tip, nose ridge and nose bottom to form a nose triangle, and introduce the definition of a *valid cardinal point*. A valid cardinal point satisfies all three heuristic conditions below (called the "*nose triangle conditions*" in this paper):

(1) The distance of the nose tip to the nose ridge must be larger than that of the nose tip to the nose bottom. In our implementation we require that the former is at least 1.2 times larger than the latter.
(2) The distance of the nose tip to the nose ridge must be within some range. In our implementation the range is 20–60 pixels. This heuristic helps to eliminate too large or too small triangle-like shapes which are not likely to be nose shapes.
(3) The altitude H perpendicular to the nose tip-nose bottom side of the triangle must be larger than 5 pixels. This heuristic helps to eliminate too flat triangle-like shapes which are not likely to be nose shapes.

The spike fitness metric calculation is more straightforward. It is given by Eq. (6) and the parameters of Eq. (6) are explained below. Assume that there are M spikes to be considered, then the fitness of the mth ($m=1,2,\ldots,M$) spike is

$$SF_m = c_m \sum_s CF_s - D_m, \tag{6}$$

[7] Note that here we do not distinguish whether a cardinal point S is 2D or 3D, because as described in footnote 6, a 2D point on a face profile has a 3D counterpart on the face scan and can be indexed using a lookup table.

where c_m is the ratio of the height of this spike to that of the highest spike, CF_s is the fitness of a valid cardinal point associated with this spike, and D_m is the maximum dimension of the cluster formed by all the *valid* cardinal points that contribute to the summation in Eq. (6).

We multiply the summation in Eq. (6) by c_m to penalize short spikes that are likely to be generated by noise. The D_m term is introduced based on the fact that the real nose tip cluster occupies a relatively small region on the dome-like nose tip area. We observed in many experiments that very small D_m values are usually caused by spur-like noise in the face scan, while very large D_m values are usually generated by points on the edges of cylinder-like objects (examples can be seen from the third column of the first and second rows in Fig. 3, near the end of the neck). In this paper we use the following function to describe D_m:

$$D_m = \begin{cases} 200e^{(-0.6D_m)} & \text{if } D_m \le 5, \\ D_m & \text{otherwise} \end{cases}. \tag{7}$$

We use an exponent term to penalize small D_m values. However, better strategies may apply.

The spike that has the largest fitness value is regarded as corresponding to the real nose tip candidates cluster. The valid cardinal point in this cluster with the highest altitude perpendicular to the nose tip-nose bottom side of the nose shape triangle is taken as the nose tip.

2.4. Algorithm complexity analysis

Now let us analyze the complexity of the proposed nose tip detection method. Assume that we have a total of N face points and generate K ($K=61$ in this paper) face profiles. Computations involved in the proposed algorithm are as follows:

1. Generation of 2D face profiles

This step involves the following two operations:

(1) *Rotate the face points around the y-axis to generate point sets in the Min Group and the Max Group*: The 3D coordinates $(x_i^\beta, y_i^\beta, z_i^\beta)$ obtained by rotating a given face point (x_i, y_i, z_i) ($i=1,2,\ldots,N$) around the y-axis by angle β is described in Eq. (1), which includes four multiplications and two additions (note that the y-coordinate is unchanged). Thus the complexity for rotating all the N face points is $O(N)$. To rotate the face points K times, the complexity is $O(KN)$.
(2) *Generate 2D face profile curves*: This part can be divided into two parts. First, we can divide the N face points of a point set in the Min Group or the Max Group into R (note that R is the number of rows in the face scan) subsets, according to the y-coordinates of the face points. The complexity of this operation is $O(N)$. Since rotations around the y-axis do not change the y-coordinates, this operation is once and for good, and the division results apply to all the K point sets in the Min Group and the Max Group. Assume that the number of face points in subset r ($r=1,2,\ldots,R$) is N_r. Then, according to Eqs. (2) and (3), to generate a 2D face profile curve one needs to compute the maximum or minimum of the x-coordinate values of all the points in each subset r ($r=1,2,\ldots,R$). The complexity of the "max" or "min" operation on an array of size N_r is $O(N_r)$, with the basic operation being a comparison between two real numbers. Since $\sum_{r=1}^{R} N_r = N$, the complexity of generating a 2D face profile curve is approximately $O(N)$. To generate K face profile curves, the complexity is $O(KN)$.

2. Detection of nose tip candidates

This step involves the following two operations.

(1) *Detect possible nose tip candidates using a circle on the 2D face profile curves*: For each 2D face profile curve, this operation consists in shifting the center of a circle along the curve and at each point of the curve counting how many of the 180 sample points, which are uniformly positioned on the perimeter of the circle, are "inside" and "outside" of the curve. To align the center of the circle with a given point on the curve, one needs to apply two additions to each of the 180 sample points. To judge wether a sample point on the circle is "inside" or "outside" of the 2D face profile curve, one only needs to directly compare the x-coordinate of the sample point with that of the curve point who shares the same y-coordinate with the sample point. Thus, the complexity of this operation is $O(180R)$, where R is the total of points constituting the 2D face profile curve. One then needs to use Eq. (4) and the -50 threshold to judge wether or not a given curve point is a possible nose tip candidate. The complexity of this operation is $O(R)$, which is negligible compared with $O(180R)$. Thus, to detect all the possible nose tip candidates on all the K face profile curves, the complexity is $O(180KR)$.

(2) *Find the cardinal points*: In this operation, the entries of a histogram are increased dependently on the positions of the possible nose tip candidates previously detected on the 2D face profile curves. Assuming that we have N_p possible nose tip candidates, the complexity of this operation is $O(N_p)$. When the histogram has been established, one needs to traverse the histogram to locate the entries that have local peak values. For this purpose, one needs to compare each entry of the histogram with its two neighboring entries. The complexity of this operation is $O(R)$.

3. Identification of the nose tip

This step involves the following two operations:

(1) *Calculate the cardinal point fitness metric*(*using* Eq (5)): This operation involves fitting a face profile curve using connected straight line segments around a given cardinal point on the face profile curve, and calculating the area of the maximum convex polygon formed by the connected line segments and the area of a specific triangle contained in the polygon. Obviously, the major computational load for this operation lies in the straight line fitting of the face profile curve. Recall that we use a line segment to approximate a curve segment if all the points in the curve segment are within distance d of the line segment (see Section 2.3). Let us assume that we successfully fit J curve segments around a given cardinal point using line segments and each curve segment contains N_j points ($j=1,2,\ldots J$), such that $Q=\sum_{j=1}^{J} N_j$ is the total of the points in all the J curve segments. It is easy to show that the complexity of checking all the N_J points contained in curve segment j ($j=1,2,\ldots J$) for straight line fitting is $O(N_j^2/2)$, with the basic operation being calculating the distance of a point on the curve segment to a straight line. When a straight line $\mathbf{L}$ is given by two points $P_0(x_0,y_0)$ and $P_1(x_1,y_1)$, the distance $d(\mathbf{L},P)$ of a point $P(x,y)$ to $\mathbf{L}$ is calculated as $d(\mathbf{L},P)=((y_0-y_1)x+(x_1-x_0)y+(x_0y_1-x_1y_0))/\sqrt{(x_1-x_0)^2+(y_0-y_1)^2}$. Thus, the complexity for the straight line fitting process is

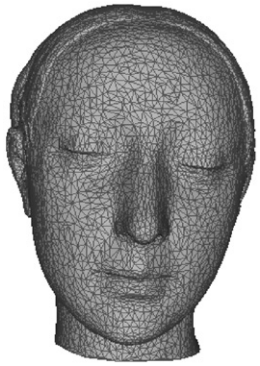

Fig. 5. The 3D head model used to render synthetic image data to test our nose tip detection method. It has 13,426 vertices and 22,984 triangular faces.

Table 1
Algorithm complexity analysis.

Step	Complexity	Basic operation	Remark
Generation of 2D face profiles			
Rotate the face points around the y-axis to generate point sets in the Min Group and the Max Group	$O(KN)$	4 multiplications and 2 additions	N is the total of face points and K is the total of 2D face profiles
Generate 2D face profile curves	$O(KN)$	A comparison between two real numbers	
Detection of nose tip candidates			
Detect possible nose tip candidates using a circle on the 2D face profile curves	$O(180KR)$	2 additions and one comparison between two real numbers	R is the number of rows in the face scan
Find the cardinal points	$O(N_p)+O(R)$	Increase a counter by one	N_p is the total of possible nose tip candidates detected.
		Two comparisons between two real numbers	
Identification of the nose tip			
Calculate the cardinal point fitness metric	$O(CQ_{max}^2/2)$	Calculating the distance of a point to a straight line	C is the total of cardinal points, and Q_{max} is the maximum possible number of facial curve points that can be involved in the straight-line-fitting around any given cardinal point. Note that the estimation of this complexity is excessive, and $O(CQ_{max}^2/2) \ll O(CR^2/2)$.
Calculate the spike fitness metric	$O(M)$	A few summations, one multiplication, and one exponential term calculation	M is the total of spikes

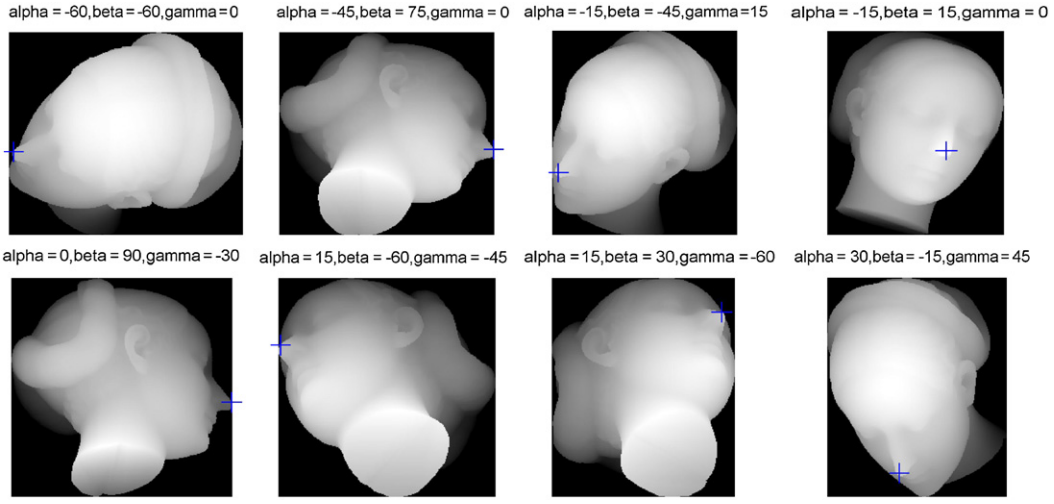

Fig. 6. Some synthetic images with detected nose tips plotted with blue crosses on them.

Table 2
Detection rates corresponding to the various roll angles in the $\times 60^\circ$ range.

Roll angle (deg)	−60	−45	−30	−15	0	15	30	45	60
Detection rate (%)	82.9	89.7	93.16	95.73	97.4	97.4	94.9	93.1	89.7

$\sum_{j=1}^{J} O(N_j^2/2)$. Since $\sum_{j=1}^{J} N_j^2 < \left(\sum_{j=1}^{J} N_j\right)^2 = Q^2$, we can safely use $O(Q^2/2)$ to approximate $\sum_{j=1}^{J} O(N_j^2/2)$. In fact, $\sum_{j=1}^{J} O(N_j^2/2)$ is almost always much smaller than $O(Q^2/2)$, especially in the case that the values of the N_j's are nearly the same. Now if we have C ($C \ll N$) cardinal points and Q_{max} is the maximum value of the associated Q's, the complexity for implementing line-fitting for all these cardinal points will be $O(CQ_{max}^2/2) \ll O(CR^2/2)$. After this step, we proceed to find the valid cardinal points among these cardinal points according to the "nose triangle conditions" (see Section 2.3). The complexity for this step is $O(C)$, with the basic operation being checking the three nose triangle conditions. This complexity is negligible compared with $O(CQ_{max}^2/2)$.

(2) *Calculate the* spike fitness *metric*: The spike fitness metric is calculated for all the M spikes using Eq. (6). The complexity for this step is $O(M)$, with the basic operation including a few summations, one multiplication, and one exponential term calculation (using Eq. (7)).

It can be seen from above that most of the computations involved in the proposed algorithm are simple. Thus, if implemented in hardware (such as the GPU implementation of Breitenstein et al.'s method [3]), the proposed method should be able to work in real time. We also summarize the above discussions in Table 1.

2.5. Pose estimation

A byproduct of the proposed method is that it can be used to coarsely estimate the roll, yaw, and pitch angles of the face pose. Assume that in Section 2.3 the nose tip is detected at face profile S_j ($j=1,2,\ldots,61$), which corresponds to an angle of β in the range $[-90^\circ, 90^\circ]$ (see Section 2.1), then the yaw rotation is estimated by

$$Yaw = \begin{cases} -(90^\circ+\beta) & \text{if } \beta < 0, \\ 90-\beta & \text{if } \beta \geq 0. \end{cases} \quad (8)$$

The pitch rotation can be estimated by computing the angle between the y-axis of the profile S_j and the nose ridge-nose bottom line (the line connecting A3 and B2 in Fig. 4).[8]

Assume that the positions of the detected nose ridge and nose bottom on the range image plane is (x_r,y_r) and (x_b,y_b), respectively. Let the angle between the vector $[x_r-y_b,y_r-y_b]$ and the y-axis of the range image plane be α, then the roll rotation can be estimated using

$$Roll = \begin{cases} -\alpha & \text{if } x_r > x_b, \\ \alpha & \text{otherwise}. \end{cases} \quad (9)$$

Table 3
77 failure cases for the synthetic data and reasons.

Number of failure cases	Reason
45	Nose triangle conditions are not satisfied
24	Nose tip is the lowest or highest point on a face scan
5	The nose tip spike is too short
3	D_m is too small

3. Experimental results

In this section, we present the experimental results of our method on synthetic image data and FRGC v2.0 data set. The

[8] To the best of our knowledge, the definition of the zero pitch of a face is rather subjective. We noticed in many cases that when the nose ridge-nose tip line is parallel to the y-axis, the face is approximately upright. Therefore, we define the pitch angle as the angle between the nose ridge-nose bottom line and the y-axis.

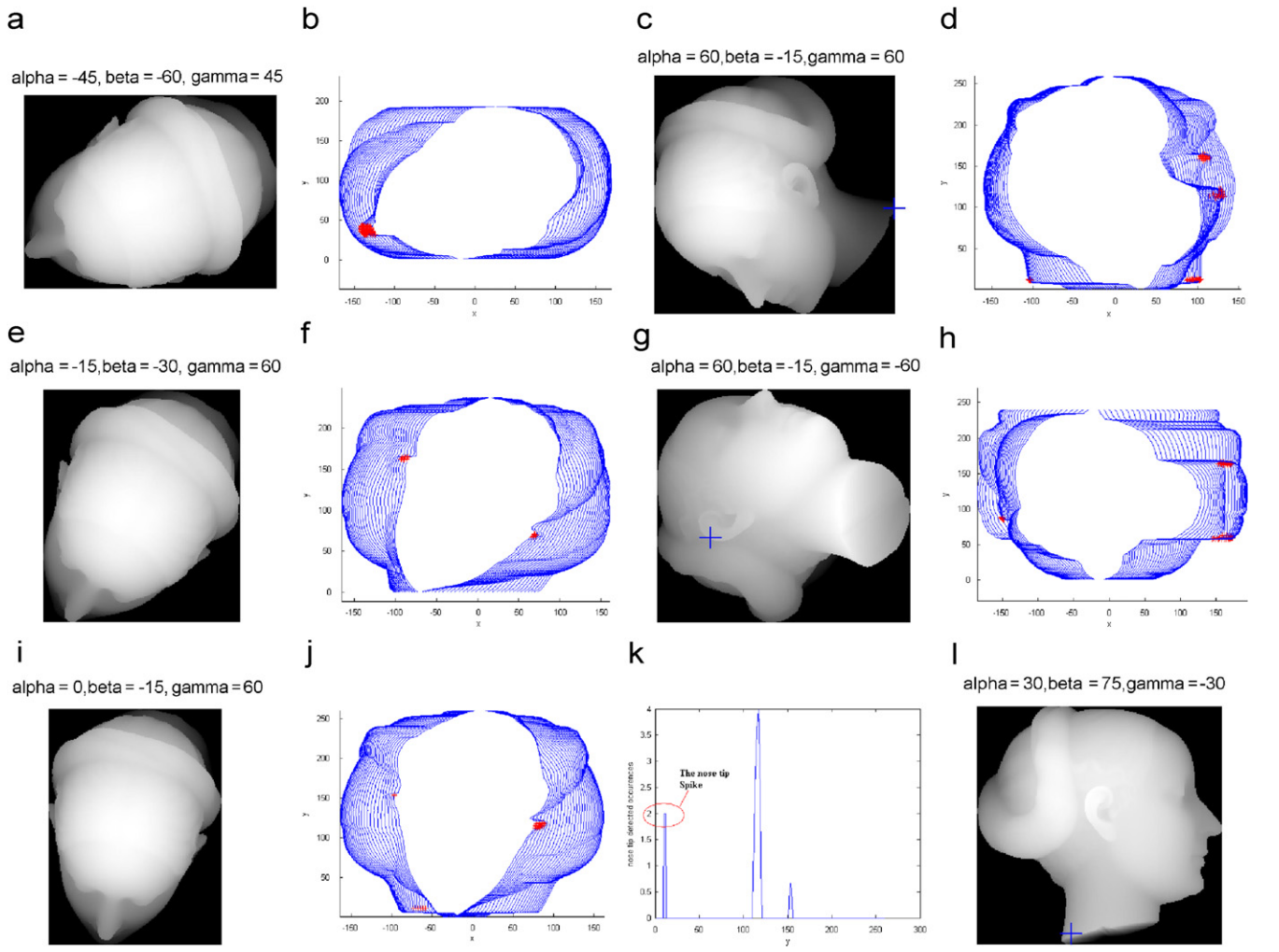

Fig. 7. Some unsuccessful examples. (a)–(d) Two examples that violate the nose triangle conditions. (e)–(h) Two examples in which the nose tip is the lowest or highest point on a face scan. (i)–(k) An example in which the failure is due to the real nose tip spike that is too short. (l) An example when the value of D_m is too small.

method was implemented in MATLAB R2007a, and the experiments were done on a Lenovo laptop (Intel core 2 Duo 2.10 GHz CPU, 1 GB RAM, Windows XP professional operating system).

3.1. Experimental results on synthetic image data

The purpose of using synthetic image data to test the proposed method is to evaluate the capability of the nose tip detection method in the handling of poses variations. To this end, we used a 3D head model to render synthetic image data under various poses. The particular 3D head model used for this purpose is a 3D clip from the "Laurana50k.ply" data that comes with a free software package, MeshLab, which is publicly available.[9] The mesh model representation is shown in Fig. 5.

We manually adjusted the pose of the 3D head model such that the face was approximately in the frontal and upright position. Then we developed a z-buffer like algorithm to render the synthetic image data under different combinations of roll (denoted by alpha), yaw (denoted by beta), and pitch (denoted by gamma).[10] In this paper, the ranges of alpha, beta, and gamma are $\times 60^\circ$, $\pm 90^\circ$, and $\times 60^\circ$, respectively, with a step size of 15° in each range. The total of synthetic images is $9 \times 13 \times 9 = 1053$. Some synthetic images with detected nose tips plotted with blue crosses on them are provided in Fig. 6. The images range from 196 to 268 pixels in two dimensions, and the average time run by the proposed method was 5.71 s.

[9] http://meshlab.sourceforge.net/.

[10] Given three basic rotation matrices in three dimensions:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix},$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix},$$

and

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

in this paper the product $R_x(\gamma)R_y(\beta)R_z(\alpha)$ represents a rotation whose roll, yaw, and pitch are α, β, and γ, respectively.

Of all the 1053 poses, nose tips were successfully detected in 976 poses (an overall detection rate of 92.68%). Table 2 provides the detection rates corresponding to the various roll angles in the $\times 60^\circ$ range.

It can be seen from Table 2 that as the absolute roll angle value increases, the performance of the proposed method decreases. The reason for this performance degradation may be explained by the fact that the number of extreme poses increases in the wake of increased roll angle values. We analyzed the 77 failure cases and summarize the failure reasons in Table 3.

It can be seen from Table 3 that the most failure cases are due to the violation of nose triangle conditions. Two examples are provided in Fig. 7(a)–(d). In both examples the distance of the (detected) nose tip to the (detected) nose ridge is shorter than that of the (detected) nose tip to the (detected) nose bottom. The second largest majority of failure cases are attributed to the reason that the nose tip is the lowest or highest point on a face scan. Under these two circumstances, the proposed method is unable to detect the real nose tip candidates Fig. 7(e)–(h). In a few number of cases the failure is due to the reason that the real nose tip spike is too short. An example is provided in Fig. 7(i)–(k). As can be seen in Fig. 7(j), most angles at the nose tip are too obtuse such that only very few nose tip candidates were detected. This phenomenon is caused by the large pitch angle. In three rare cases, the failure is related to values of D_m that are too small. An example is provided in Fig. 7(l), in which $D_m=2.92$. It is recognized as a spur noise and therefore undergoes severe punishment by the exponent term in Eq. (7). We believe that this is because the nose tip is too "sharp" at that particular pose, when the flat nose flank is parallel to the face scan's *xy*-plane.

We also estimated the roll, yaw and pitch angles and compared the results with the ground truth values. The results are summarized in Table 4.

It can be seen in Table 4 that the yaw estimation is much more accurate than the roll and pitch estimation. The reason can be explained as follows. Recall that in Section 2.3 the nose tip is regarded as the valid cardinal point with the maximum altitude perpendicular to the nose tip-nose bottom side of the nose shape triangle. However, this maximum is mostly reached when a face profile (from which the nose tip is detected) is generated by rotating the 3D data around the *y*-axis to some exact angle β. Thus as the proposed method can choose from a set of face profiles that has a "fine resolution" (recall that the step size of β for generating adjacent face profiles is 3°) for the one with the maximum altitude of the nose shape, it is expectable that the yaw estimation can be

Table 4
The pose estimation results.

Estimation error	Roll angle								
	-60°	-45°	-30°	-15°	0°	15°	30°	45°	60°
Roll estimation accuracy rates (%)									
$\leq 15^\circ$	42.3	36.2	45.9	67.9	79.8	73.6	49.6	48.6	43.81
$>15^\circ, \leq 30^\circ$	6.2	29.5	35.8	22.3	14.1	24.6	35.1	16.6	5.71
$>30^\circ, \leq 45^\circ$	10.3	19.1	16.5	8.0	3.5	0.9	11.7	22.9	9.52
$>45^\circ$	41.2	15.2	1.8	1.8	2.6	0.9	3.6	11.9	40.95
Yaw estimation accuracy rates (%)									
$\leq 15^\circ$	94.9	89.5	87.2	85.7	82.5	85.1	80.2	89.0	95.24
$>15^\circ, \leq 30^\circ$	5.1	10.5	12.8	13.4	15.8	11.4	16.2	7.3	3.81
$>30^\circ, \leq 45^\circ$	0	0	0	0	1.7	3.5	2.7	2.8	0.95
$>45^\circ$	0	0	0	0.9	0	0	0.9	0.9	0
Pitch estimation accuracy rates (%)									
$\leq 15^\circ$	62.9	65.7	65.1	64.3	63.2	59.7	58.6	59.6	54.29
$>15^\circ, \leq 30^\circ$	17.5	18.1	20.2	20.5	17.5	21.0	23.4	22.9	25.71
$>30^\circ, \leq 45^\circ$	10.3	7.6	10.1	9.8	11.4	12.3	10.8	11.9	9.52
$>45^\circ$	9.3	8.6	4.6	5.4	7.9	7.0	7.2	5.6	10.48

Note: The results are calculated only for the images in which the nose tip was successfully detected.

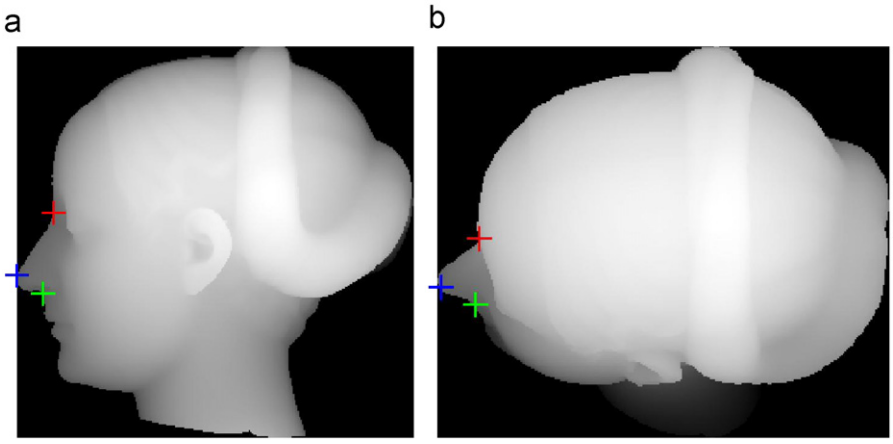

Fig. 8. Example to illustrate that the pitch estimation is more robust to roll changes than does the roll estimation: (a) corresponds to roll=0°, yaw=90°, and pitch=0° pose while (b) corresponds to roll=−60°, yaw=90°, and pitch=0° pose. The red, blue, and green crosses denote the detected nose ridge, tip and bottom on the respective images, respectively. The estimated poses in (a) and (b) are (roll=−3.0°, yaw=−87°, and pitch=2.88°) and (roll=−7.52°, yaw=−81°, and pitch=−6.0°), respectively (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

quite accurate. The roll estimation is the most inaccurate because it is based on the re-projections of the nose ridge and bottom (detected on a 2D face profile) onto the range image plane, which is error-prone when the roll angle is large. Although the pitch estimation is also based on the detected nose ridge and bottom on a 2D face profile, it is more accurate than the roll estimation because it is more robust to roll changes. To illustrate this point we provide an example in Fig. 8. Fig. 8(a) corresponds to roll$=0^\circ$, yaw$=90^\circ$, and pitch$=0^\circ$ pose while (b) corresponds to roll$=-60^\circ$, yaw$=90^\circ$, and pitch$=0^\circ$ pose. The red, blue and green crosses denote the detected nose ridge, tip and bottom on the respective images, respectively.

It can be seen from Fig. 8(b) that although the nose ridge and nose bottom detections are both incorrect, the pitch estimation is quite accurate.

We further map the pose estimation errors in 13×9 rectangles composed of small squares as shown in Fig. 9 (see the figure caption for details). The middle square of a rectangle corresponds to (yaw$=0^\circ$, pitch$=0^\circ$) pose and from left (top) to right (bottom) the pitch (yaw) angle increases in a step size of 15°.

As expected, it can be seen from Fig. 9 that in almost all the rectangles the pose estimation accuracy is better in the middle compared with the boundaries.

3.2. Experimental results on the FRGC v2.0 data

We tested the proposed method on FRGC v2.0 data set [31]. Our proposed algorithm assumes that the input to the algorithm is a facial region. However, the original FRGC v2.0 range data also

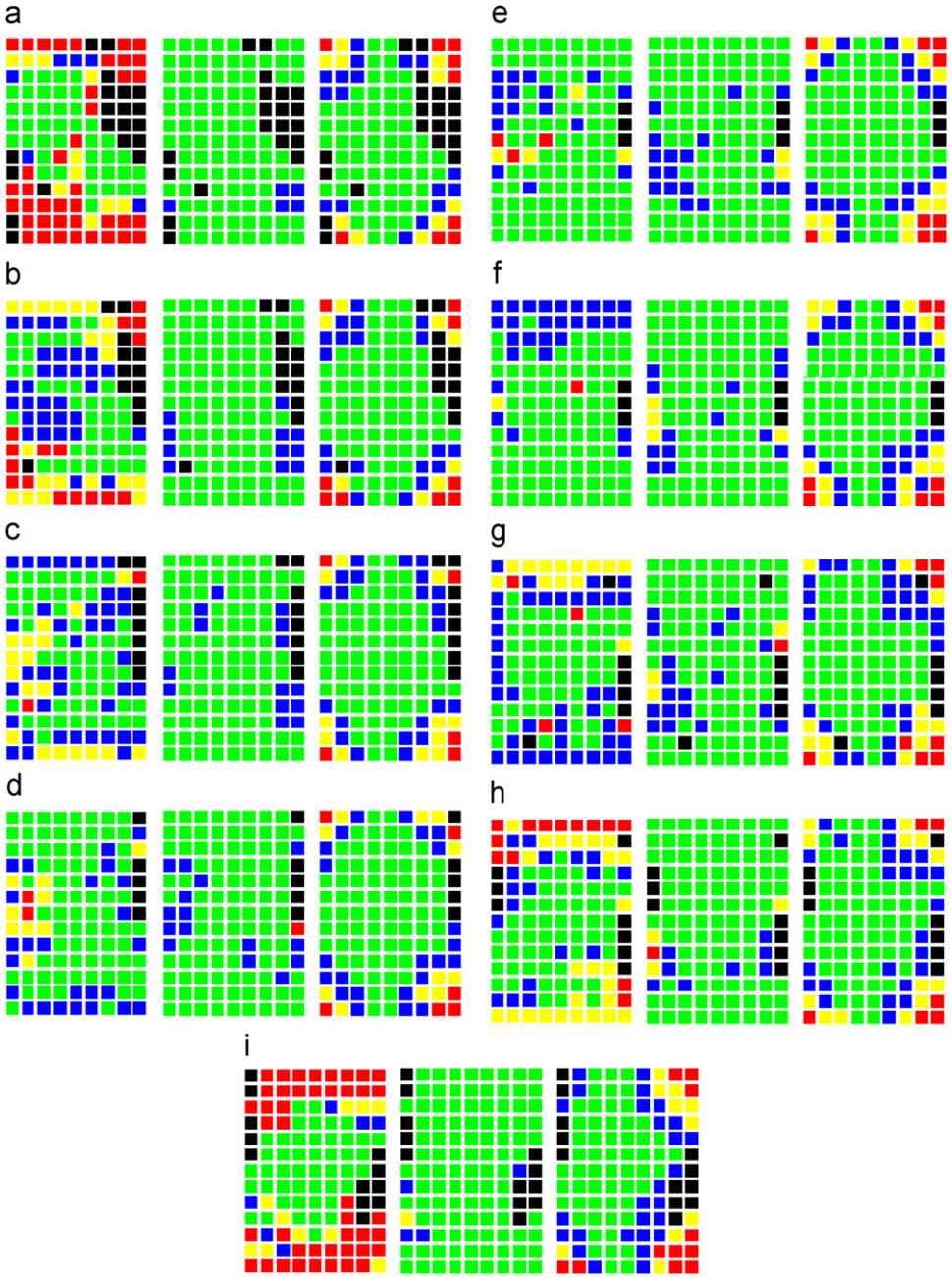

Fig. 9. The pose estimation errors distribution maps: (a)–(i) correspond to the -60°, -45°, -30°, -15°, 0°, 15°, 30°, 45°, and 60° in roll angles, respectively. The left, middle, and right columns correspond to the roll, yaw, and pitch estimations, respectively. Green, blue, yellow, and red squares denote estimation errors below 15°, within (15°, 30°], within (30°, 45°], and beyond 45°, respectively. Black squares denote detection failures (figure best seen in color) (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

contains background, clothes, and other parts of a human body besides the facial region. Therefore, existing face detection algorithms can first be used to roughly locate the facial region followed by a resampling operation of the facial region to fill in holes and make all the data points uniformly distributed. Because face detection is not a major interest in this paper, our experiments use a set of face scans with cropped faces from the original FRGC v2.0 range images. The reasons for using these face scans in our experiments are two-fold. First, these face scans have been resampled to a resolution of 1 mm/pixel in the image plane, so we can directly apply our proposed algorithm without changing its parameters (see footnote 5). Second, the nose tips in the face scans have been detected using techniques described in our previous research [1] and have been normalized to be localized at the middle of each face scan. Since the techniques described in [1] have proven to work very well on FRGC v2.0 data, we can directly use these nose tip positions as our benchmarks. The face scans are 161×161 pixels in size. Of all the 4950 face scans, one scan has no nose tip. The average time run by the proposed method was 1.96 s. For the remaining 4949 data, the proposed method achieved a detection rate of 99.43% (4921 successful detections in all the 4949 test cases) in terms of nose tip localization errors within 20 mm from our benchmark. As has been reported in Colbry et al.'s work [19], the 20 mm error threshold is acceptable as it can be tolerated in the Iterative Closest Point (ICP) algorithm which is widely used in face recognition. Of the 99.43% successful detections, 4750 cases (95.98% of all the 4949 test cases) have nose tip localization errors within 10 mm from our benchmark, while 171 cases (3.46% of all the 4949 test cases) have nose tip localization errors within the range 10–20 mm from our benchmark. These results are summarized in Table 5.

Table 5
Nose tip localization errors from our benchmark [1].

Nose tip localization errors (mm)	[0,10]	(10, 20]	>20
Percentage of the total (%)	95.98	3.46	0.56

Some successful nose tip detection examples are shown in Fig. 10.

We analyzed the 28 cases in which the nose tip localization errors are larger than 20 mm from our benchmark [1]. Two cases have nose tip localization errors of 21.38 and 21.93, respectively. The results are shown in Fig. 11(a) and (b). However, it is obvious that the results obtained using the proposed method are more accurate than our benchmark [1] because they are closer to the nose tip on the range images based on visual inspection. In another eight cases, because the regions around the nose tips are so flat that they generate very short corresponding nose tip spikes, like the example shown in Fig. 11(e), and thus result in the abandonment of the corresponding nose tip clusters. We believe that this is caused by the image quality of the face scan. In the remaining 18 cases, the failure is simply due to the limitation of our algorithm which falsely misclassifies some face points as nose tips. An example is provided in Fig. 11(g)–(j), in which the bottom lip is recognized as the nose tip. To further reveal the reason behind this phenomenon, we provide an enlarged face profile Fig. 11(j) on which we annotated the positions of the nose tip, nose ridge, and nose bottom, detected by the nose tip candidates detection step in Section 2.2, respectively. The triangle formed by these three points satisfies the nose triangle conditions in Section 2.3 and thus contributes to the spike fitness calculation in Eq. (6). Currently, the proposed method has no mechanism to cope with this problem and it will be dealt with in future work.

The detection rate of our method is comparable to that of a most recent nose tip detection method [7] which achieves 99.6% detection rate on FRGC v2.0 data set. However, the method in Ref. [7] is much more computationally expensive (see footnote 2), requires training, and was only tested on 3680 data of all the 4950 data.

4. Conclusions and discussions

In this paper, we propose a 2D face profiles based nose tip detection method. Our approach does not require any training and does not rely on any particular model. We extensively tested the

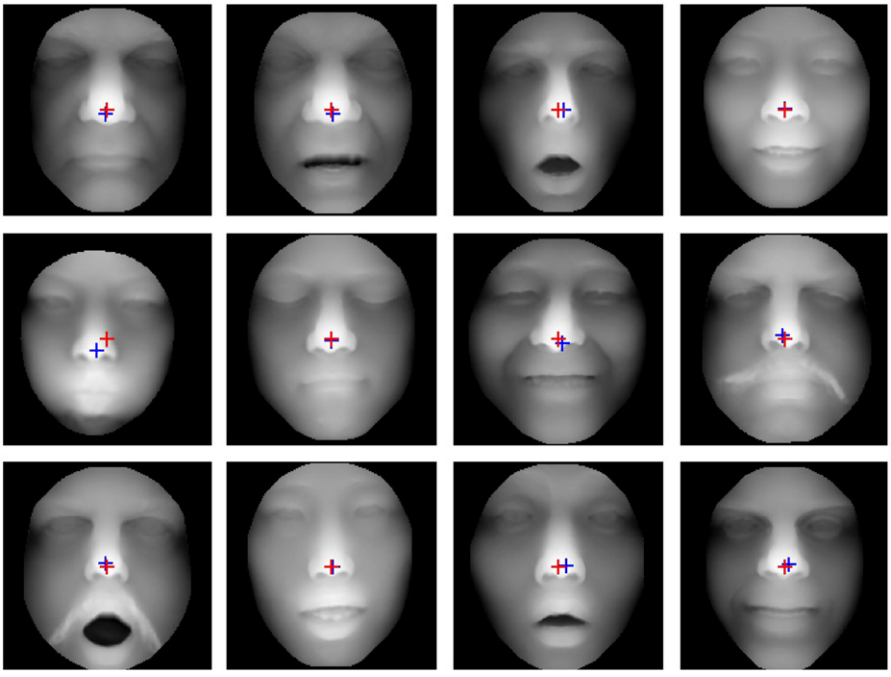

Fig. 10. Some successful nose tip detection examples. Red crosses correspond to our benchmark nose tip positions [1], while blue crosses correspond to the detected nose tips using our proposed method (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

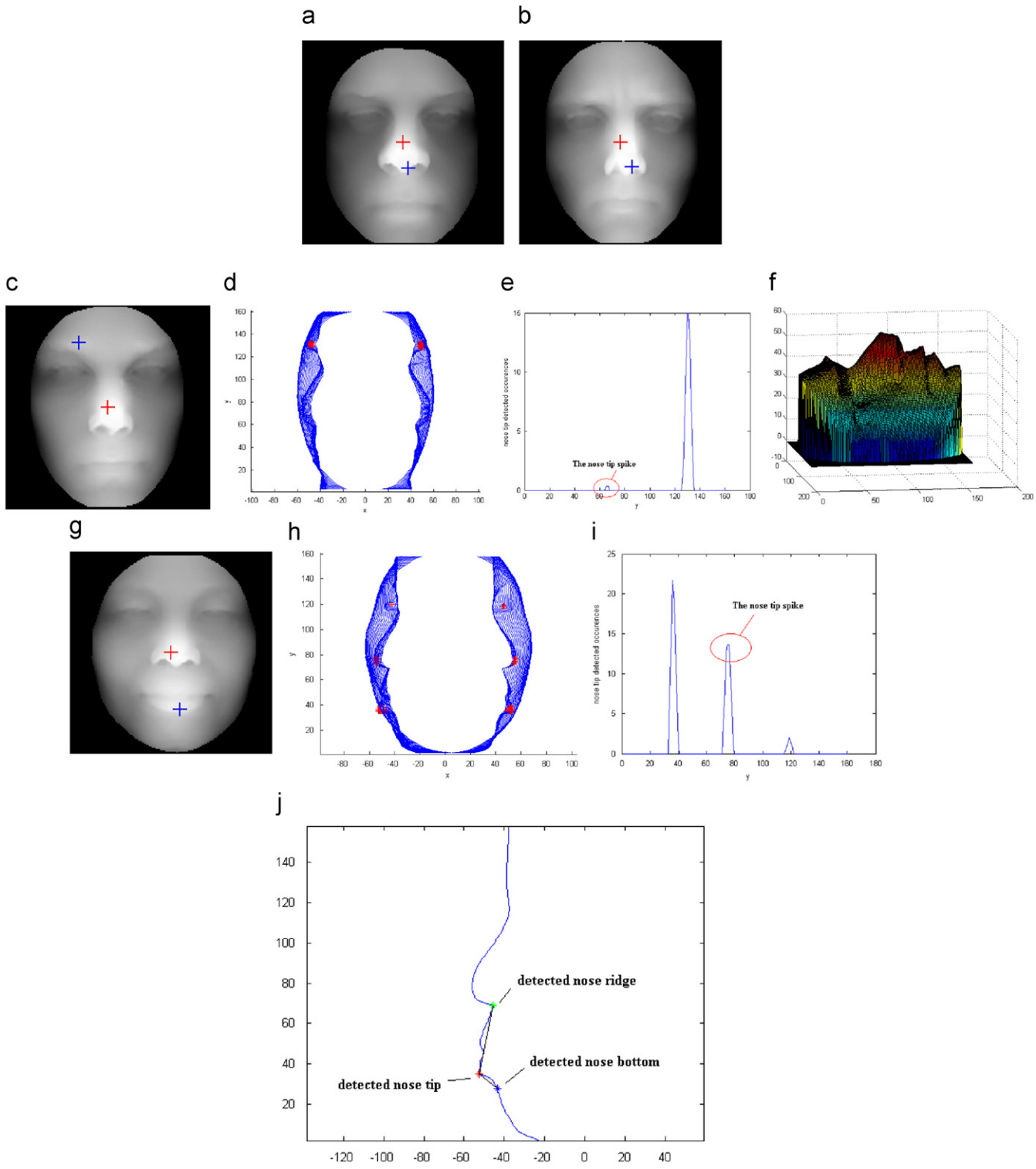

Fig. 11. Examples in which the nose tip localization errors are larger than 20 mm from our benchmark [1]. Red crosses correspond to our benchmark nose tip positions, while blue crosses correspond to the detected nose tips using our proposed method. (a, b) Two examples in which the results obtained using the proposed method are better than our benchmark. (c–f) An example in which the failure of detecting the nose tip is due to the nose tip region being too flat. (g–j) An example in which the bottom lip is recognized as the nose tip (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

proposed method on synthetic image data rendered by a 3D head model and FRGC v2.0 data set. Experimental results show that the proposed method is robust to many scenarios that are encountered in common face recognition applications. A high detection rate of 99.43% was obtained on FRGC v2.0 data set. Promising results also show that the proposed method can also be used to coarsely estimate the roll, yaw and pitch angles of the face pose.

The proposed method is quite fast, requiring only seconds to process an image of 100–200 pixels (in both x and y dimensions) with a MATLAB implementation. If implemented in hardware

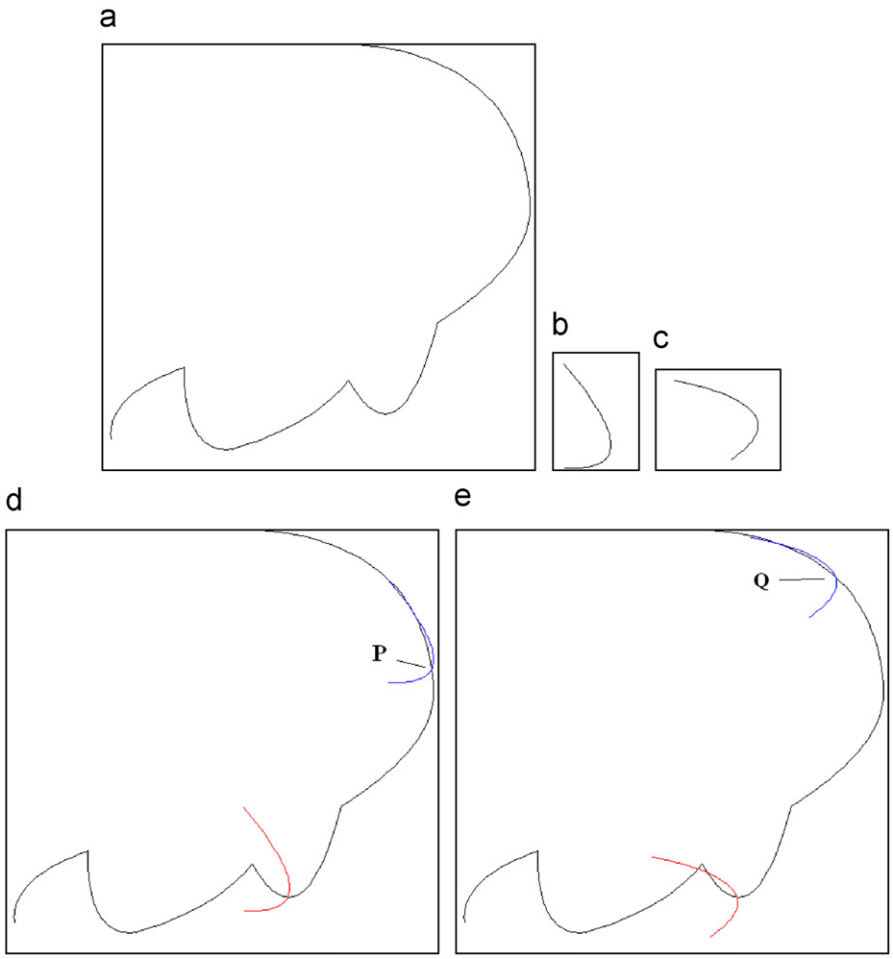

Fig. A1. An example to illustrate the problem of the two-model-matching idea in [27]: (a) A manually drawn face profile curve. (b) Nose profile Model 1 corresponding to 0° pitch. (c) Nose profile Model 2 corresponding to 45° pitch. (d) The positions of Model 1 at the best match and at the nose tip, shown in blue and red, respectively. "*P*" denotes the detected nose tip. (e) The positions of Model 2 at the best match and at the nose tip, shown in blue and red, respectively. "*Q*" denotes the detected nose tip (figure best seen in color) (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

(such as the GPU implementation of Breitenstein et al.'s method [3]), the proposed method should be able to work in real time.

One limitation of the approach occurs when ambiguous regions of the face profile are falsely detected as nose tips. This ambiguity is rare. It will however be addressed in future work to further increase the detection rate. In the future we will also explore the proposed method's potential in detection-based nose tracking in range image sequences and pose estimation.

Acknowledgements

The first author would like to thank the China Scholarship Council (CSC) for its financial support to the research reported in the paper. The research was also partly supported by a research fund (Grant No. L08010701JX0761) from University of Electronic Science and Technology of China.

Appendix

Here we will use a simple example to reveal the problem of the two-model-matching idea in [27]. Fig. A1(a) is a manually drawn face profile curve. Fig. A1(b) and (c) are two nose profile models corresponding to the 0° and 45° pitches, respectively. They were actually obtained by rotating the nose shape in Fig. A1(a) by these respective angles. We translate the two models along the face profile curve to find the points *P* and *Q* that best match the nose tips of the two models, respectively, Fig. A1(d) and (e). It is obvious that neither *P* nor *Q* or the midpoint of *P* and *Q* is the nose tip. Although these two models are not the same models as the ones mentioned in [27], this example does show that matching using a particular model is likely to be error-prone, even if the model belongs to the same person.

References

[1] A.S. Mian, M. Bennamoun, R. Owens, An efficient multimodal 2D–3D hybrid approach to automatic face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (11) (2007) 1927–1943.
[2] X. Lu, A.K. Jain, D. Colbry, Matching 2.5D face scans to 3D models, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (1) (2006) 1927–1943.
[3] M.D. Breitenstein, D. Kuettel, T. Weise, L.J. Van Gool, H. Pfister, Real-time face pose estimation from single range images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), June 2008, pp. 1–8.
[4] S. Malassiotis, M.G. Strintzis, Robust real-time 3D head pose estimation from range data, Pattern Recognition 38 (2005) 1153–1165.
[5] K.I. Chang, K.W. Bowyer, P.J. Flynn, Multiple nose region matching for 3D face recognition under varying facial expression, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1695–1700.
[6] M.L. Koudelka, M.W. Koch, T.D. Russ, A prescreener for 3D face recognition using radial symmetry and the Hausdorff fraction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 1–8.

[7] N. Pears, T. Heseltine, M. Romero, From 3D point clouds to pose-normalized depth maps, International Journal of Computer Vision . doi:10.1007/s11263-009-0297-y.
[8] Y. Wang, C.S. Chua, Y.K. Ho, Facial feature detection and face recognition from 2D and 3D images, Pattern Recognition Letters 23 (2002) 1191–1202.
[9] C. Xu, T. Tan, Y. Wang, L. Quan, Combining local features for robust nose location in 3D facial data, Pattern Recognition Letters 27 (2006) 1487–1494.
[10] A. Colombo, C. Cusano, R. Schettini, 3D face detection using curvature analysis, Pattern Recognition 39 (2006) 444–455.
[11] C. Conde, L.J. R-Aragon, E. Cabello, Automatic 3D face feature points extraction with spin images, Lecture Notes in Computer Science 4142 (2006) 317–328.
[12] A.E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (5) (1999) 433–449.
[13] C.S. Chua, R. Jarvis, Point signatures: a new representation for 3D object recognition, International Journal of Computer Vision 25 (1) (1997) 63–85.
[14] C. Dorai, A.K. Jain, COSMOS—a representation scheme for 3D free-form objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (10) (1997) 1115–1130.
[15] M.P. Segundo, C. Queirolo, O.R.P. Bellon, L. Silva, Automatic 3D facial segmentation and landmark detection, in: Proceedings of the 14th International Conference on Image Analysis and Processing (ICIAP 2007), 2007, pp. 431–436.
[16] Y. Sun, L. Yin, Automatic pose estimation of 3D facial models, in: Proceedings of the 2008 International Conference on Pattern Recognition, 2008, pp. 1–4.
[17] Y. Sun, L. Yin, 3D face recognition using two views face modeling and labeling, in: Proceedings of the 2005 International Conference on Computer Vision and Pattern Recognition, 2005, p. 117.
[18] W.J. Chew, K.P. Seng, L.M. Ang, Nose tip detection on a three-dimensional face range image invariant to head pose, in: Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1, 2009.
[19] D. Colbry, G. Stockman, A. Jain, Detection of anchor points for 3D face verification, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, p. 118.
[20] F.B. ter Haar, R.C. Veltkamp, A 3D face matching framework, in: 2008 IEEE International Conference on Shape Modeling and Applications, 2008, pp. 103–110.
[21] P. Szeptycki, M. Ardabilian, L. Chen, A coarse-to-fine curvature analysis-based rotation invariant 3D face landmarking, in: International Conference on Biometrics: Theory, Applications and Systems, 2009, pp. 32–37.
[22] M. Romero, N. Pears, 3D facial landmark localization by matching simple descriptors, in: IEEE Second International Conference on Biometrics Theory, Applications and Systems, September 2008, pp. 1–6.
[23] T. Whitmarsh, R.C. Veltkamap, M. Spagnuolo, S. Marini, F.T. Haar, Landmark detection on 3D face scans by facial model registration, in: Proceedings of the First International Workshop on Shape and Semantics, 2006, pp. 71–76.
[24] A. Slater, Y.H. Hu, N. Boston, Multiscale integral invariants for facial landmark detection in 2.5D data, in: Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP'07), October 2007, pp. 175–178.
[25] J. Goldfeather, V. Interrante, A novel cubic-order algorithm for approximating principal direction vectors, ACM Transactions on Graphics 23 (1) (2004) 45–63.
[26] X. Lu, A.K. Jain, Automatic feature extraction for multiview 3D face recognition, in: Proceedings of the Seventh International Conference on Automatic Face and Gesture Recognition, 2006, pp. 585–590.
[27] T.C. Faltemier, K.W. Bowyer, P.J. Flynn, Rotated profile signatures for robust 3D feature detection, in: Proceedings of the Eighth IEEE International Conference on Automatic Face and Gesture Recognition (FG '08), September 2008, pp. 1–7.
[28] M. Haker, M. Bohme, T. Martinetz, E. Barth, Geometric invariants for facial feature tracking with 3D TOF cameras, in: Proceedings of the IEEE International Symposium on Signals, Circuits and Systems (ISSCS), vol. 1, 2007, pp. 109–112.
[29] G. Loy, A. Zelinsky, Fast radial symmetry for detecting points of interest, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (8) (2003) 959–973.
[30] A. Rajwade, M.D. Levine, Facial pose from 3D data, Image and Vision Computing 24 (2006) 849–856.
[31] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, Proceedings of the IEEE Computer Vision and Pattern Recognition 1 (2005) 947–954.

Xiaoming Peng received his Ph.D. in Pattern Recognition and Intelligent Systems from Huazhong University of Science and Technology, China in 2005. He is an Associate Professor with School of Automation Engineering, the University of Electronic Science and Technology of China. He is also a visiting research fellow with the School of Computer Science and Software Engineering, The University of Western Australia. His research interests include image registration, computer vision, and vision based pattern recognition.

Mohammed Bennamoun received his M.Sc. from Queen's University, Kingston, Canada in the area of Control Theory, and his Ph.D. from Queen's /Q.U.T in Brisbane, Australia in the area of Computer Vision. He is a Winthrop Professor and Head of School of Computer Science and Software Engineering, The University of Western Australia. He published over 120 journal and conference publications. His research interests include control theory, computer vision, and signal/image processing.

Ajmal S. Mian received his Ph.D. in Computer Science from The University of Western Australia, Australia in 2007. He is an Australian Research Fellow and works as an associate research professor with the School of Computer Science and Software Engineering, The University of Western Australia. His research interests include pattern recognition, computer vision, multimodal biometrics, and information security.