Automatic 3D facial segmentation and landmark detection

Maurício P. Segundo, Chauã Queirolo, Olga R. P. Bellon, Luciano Silva*
IMAGO Research Group - Universidade Federal do Paraná
P.O. Box 19092 – 81531-990 – Curitiba-PR-Brazil
http://www.inf.ufpr.br/imago
{mps04,olga,luciano}@inf.ufpr.br

Abstract

This paper presents our methodology for face and facial features detection to improve 3D face recognition in a presence of facial expression variation. Our goal was to develop an automatic process to be embedded in a face recognition system, using only range images as input. To do that, our approach combines traditional image segmentation techniques for face segmentation and detect facial features by combining an adapted method for 2D facial features extraction with the surface curvature information. The experiments were performed in a large, well-known face image database available on the Biometric Experimentation Environment (BEE), including 4,950 images. The results confirms that our method is efficient for the proposed application.

1. Introduction

For many years, 2D face recognition was the principal focus on researches regarding face biometrics, and many limitations were identified, like variations in illumination, pose and expressions. The potential of 3D face recognition to solve some of these difficulties is one of the main reasons for the great growth of interest in this research area. However, variations of facial expression still present difficulties.

According to Chang et al. [5], there are at least three main approaches to improve recognition in the presence of facial expression variation. The simplest one is to create a gallery, sampling a set of different facial expression for each person, and to perform the matching between a probe and a set of images that represent a person. However, it is impractical to create a gallery that is generic enough so that no new sample contains an unknown expression. The second approach is to use only the rigid regions from the face (the most invariant regions under different facial expressions) for recognition. The third approach is to create models of the changes between neutral faces and faces with expressions of the same person, and then performing the matching between a gallery of neutral image that can be changed by the obtained models and a probe image.

The latter two approaches have a similarity: both need to locate some specific regions inside the face. One needs to find the rigid region of the face, and the other needs to find the regions changed by a facial expression and to create a model of this change. To do that, both approaches are based on the detection of some facial features.

To extract the face region and facial features, some works use both color and depth information [7, 13]. As color may present illumination problems, other works only use depth information for this process [5, 6]. In this paper we present our methodology for preprocessing a 3D face image for recognition, from face segmentation until facial feature detection by using only the input depth information.

The correctness of high level processing tasks in image analysis usually depends on how well the input image was segmented [8]. This is particularly true for face recognition. The goal of this work is to segment correctly the face from an input range image and to locate its appropriate facial features to provide sufficient information for face recognition with varying facial expression.

In our experiments, we used the two versions of the Face Recognition Grand Challenge database[1], containing a variety of facial expressions, scale, pose and others characteristics, like hairstyle.

This paper is organized as follows. Firstly, in Section 2, we introduce our segmentation approach to detect the face region from an input range image. The methodology for facial features detection is presented in Section 3. Details regarding the experimental results are discussed in Section 4, followed by the final remarks in Section 5.

*The autors gratefully acknowledge CNPq and FINEP for the financial support.

[1]http://www.bee-biometrics.org

2. Face segmentation in range images

The first goal of our approach is to extract the face region from an input range image containing only one subject, as shown in Fig. 1. In this figure we show rendered images to emphasize the shape of the surfaces. Our segmentation approach is based on traditional image segmentation techniques: edge detection [3], region detection [10] and shape detection [14], to overcome their limitations by taking advantage of the combination of their strengths.

Figure 1. Examples from the database.

Our face segmentation algorithm is composed basically by two main stages: (1) locating homogeneous regions in the input image by using clustering combined with edge data; and (2) identifying candidate regions that belong to the face region by an ellipse detection method based on the Hough Transform [1]. More details about each procedure are presented below.

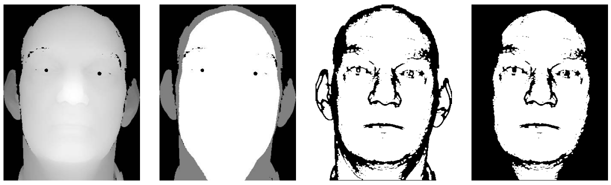
2.1. Locating face candidates by combining region and edge detection

As can be seen in Fig. 1, to segment a face we have to isolate the face region from both the background and the rest of the body. Although there is only one subject per image, there are many situations that interfere with the segmentation process, as will be discussed in this section.

We apply the K-Means algorithm [10] by setting k = 3 (*i.e.*, the number of clusters) to segment the image in three main regions: background, body, and face. Considering the databases we used, we observe that the face is the region with small depths, *i.e.*, closer to the acquisition system. This heuristic works in 100% of images from both databases, including around 5,000 images. However, this step alone is not enough to correctly extract the face region because some characteristics may interfere, *i.e.*, hairstyle, ears, neck. Then, the goal of the edge detection process is to help in eliminating these parts from the resulting segmented face region after applying the K-Means.

To perform edge detection we apply the Sobel operator [8] on the depth information from the range image and define a threshold. The gradient threshold is based only on spatial information and is easily obtained by analyzing the histogram of the gradient image. We computed the histograms for all gradient images and observed that they have similar behavior. Thus, we defined an automatic threshold based on the inflection point of the histogram curve for each gradient image to detect the face boundaries. In general, the computed threshold has a value between 8 and 12 for the tested databases and from now on the system works automatically, *i.e.*, no other threshold is necessary. Finally, to obtain the final edge map we applied a closing process [11] (dilation followed by erosion of neighbor pixels) to link some broken lines generated by the threshold operation.

After performing region and edge detection, which can be made in parallel, we combine the two resulting images by using an AND operation. The combined image can be easily processed to obtain the region of interest (ROI), *i.e.*, the face region, without irrelevant, disturbing parts. Fig. 2 shows the results of edge detection, region detection, and their combination.



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

Figure 2. Examples of (a) depth information of an input image, (b) region detection by K-Means, (c) edge detection and (d) combined image.

The resulting image, after combining the edge and region detection, undergoes a labeling process and we eliminate regions with size below 1% of the image size (usually some hair remains). From the resulting labeled image, a new edge map is generated, which will be used to find the face.

2.2. Face localization by shape detection

In this stage, we used the new edge map previously obtained to precisely locate the face region in the input image. To perform shape detection we used the Hough Transform [1], looking for an ellipse [14], which is considered to be the geometric shape most similar to the face boundary.

After the ellipse detection, we need to select the labeled regions that belong to the face region, which is done by selecting regions inside of the detected ellipse. Fig. 3 shows an example of face location using ellipse detection.

The region selected as the ROI is represented a binary image that indicates the location of the face region in the input image. However, this binary image can have some holes (*i.e.* lack of information, see Fig. 3(e)) that must be fulfilled by a closing process [8].

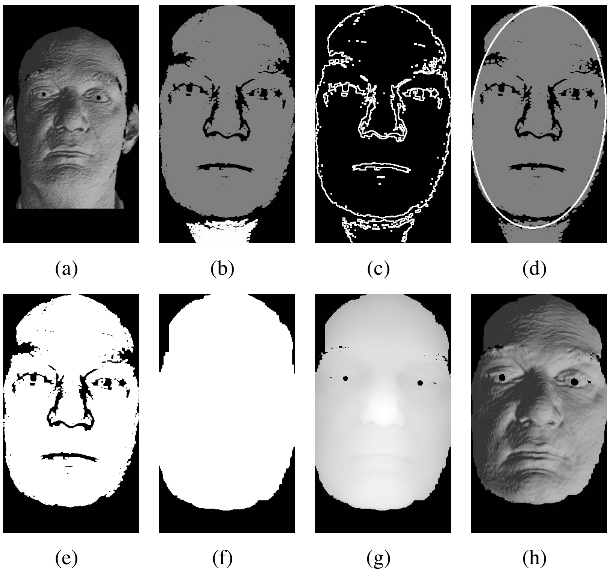


Figure 3. Examples of (a) input range image, (b) labeled face candidates, (c) new edge map, (d) detected ellipse, (e) selected face candidates, (f) binary image after normalization and (g)-(h) final segmentation.

After that, we perform a logical AND between the resulting binary image and the input range image, obtaining the final segmentation of the face region (Fig. 3(g) and Fig. 3(h)).

3. Facial features detection

Facial features detection is generally an important step to optimize some stages of face recognition, like face registration (pre-alignment), estimation and correction of the face pose (image acquisition), and locating rigid regions of the face [6] (facial expression robustness).

As well as in face segmentation, some works only use depth information to find facial features [5], and others use both depth and color information [7]. Our method uses only depth information during the entire process, from face segmentation until facial features extraction.

We are interested in a small group of facial features: nose tip, eye and nose corners and nose base. These features were chosen because they allows a precise location of the rigid regions (*e.g.*, the nose region). To locate them, we combine 2D facial features detection technique [12] with surface curvature analysis [4]. More details about how we used these methods to locate the desired features are presented below.

3.1. Curvature-based features detection

By analyzing the surface curvature information of a face, we can see that some face regions generally present the same curvature type and are not affected by facial expression, also according to [5]. In this stage we used the segmented face images that can present some noise data, and to reduce the noise a median filter of size 5×5 [8] was applied twice in those images.

To compute the curvature type for each point from the face image, we define a local surface in a small neighborhood $N \times N$ (in our case, $N = 7$). Firstly, we estimate the surface normal vector using principal components analysis (PCA) on the stipulated neighborhood [9]. The orientation is taken to be the eigenvector associated with the smallest eigenvalue, and we use this orientation as the new Z-axis in a local coordinate system for a better classification of the curvature type.

Then, we calculate the coefficients of the quadratic surface using a least square fitting technique, and we estimate the Gaussian curvature (K) and the mean curvature (H) by computing the partial derivatives. Once we have both K and H values, we can classify the quadric surface (flat, peak, pit and so on) according to Table 1 [4].

K \ H	< 0	$= 0$	> 0
< 0	saddle ridge	minimal	saddle valley
$= 0$	ridge	flat	valley
> 0	peak	(none)	pit

Table 1. Surface types.

We are interested in only two surface shapes: peaks and pits. Almost all desired face features present one of these surface types, as can be seen in Fig. 4. The surface type of both eye corners and nose base is ”pit”, and for the nose tip the type is ”peak”.

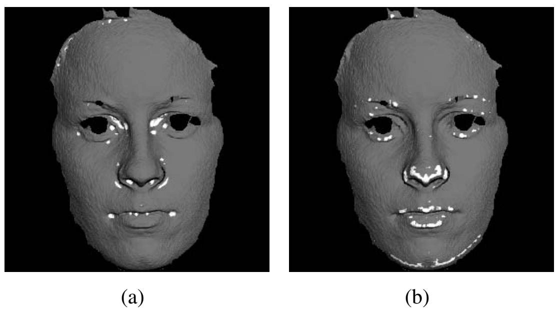


Figure 4. Examples of (a) pit curvature regions and (b) peak curvature regions.

3.2. Nose tip and nose corners detection

In this work, we present a new method for nose tip detection in 3D face images based on ideas for 2D facial features detection presented in [12]. Firstly, we find the y-coordinate of the nose tip, followed by the x-coordinate. To find the y-coordinate, we compute two y-projections of the topographic depth information relief, and we call these two projections as profile curve and median curve, respectively. This is accomplished by determining the maximum depth value (profile curve) and the median depth value (median curve) of every set of points with the same y-coordinate from the face image. Fig. 5 shows an example of these two curves.

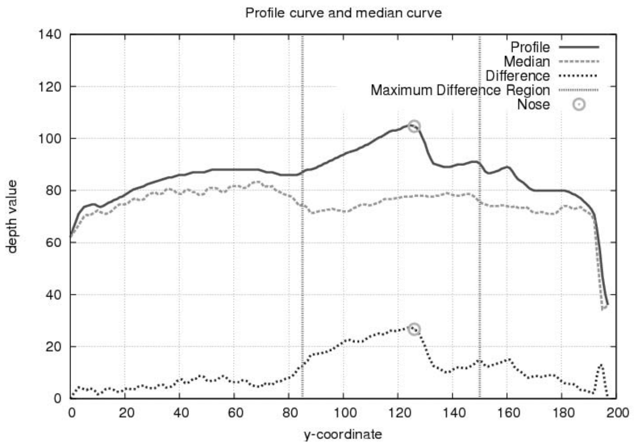


Figure 5. Example of nose tip y-coordinate detection.

The most straightforward way to find the nose tip is looking for a peak in the profile curve. However, if the face presents some pose variation and/or some noise that was not eliminated by the median filtering, this approach may results in wrong nose tip detection. To overcome this situation, we used another approach to find the nose tip. Firstly, we generate another curve that represents the difference between profile and median curves. Then, we look for the stretch from this "curve of differences" that has the maximum sum of difference values, and we assume that the nose tip is a peak at the profile curve inside this stretch. We empirically defined, based on our experimental results, that the stretch lenght corresponds to one-third of the face height. Fig. 5 shows an example of the "curve of differences", the stretch with maximum difference and nose tip detection.

Once the y-coordinate of the nose tip is detected, we compute the x-projection of the curvature image to find the x-coordinate of the nose tip. This is done by calculating the percentage of peak curvature points of 7 neighbor rows (empirically defined for the used databases) centered in the nose tip y-coordinate of every column. The nose tip x-coordinate can be determined by looking for a peak at this projection, as can be seen in Fig. 6.

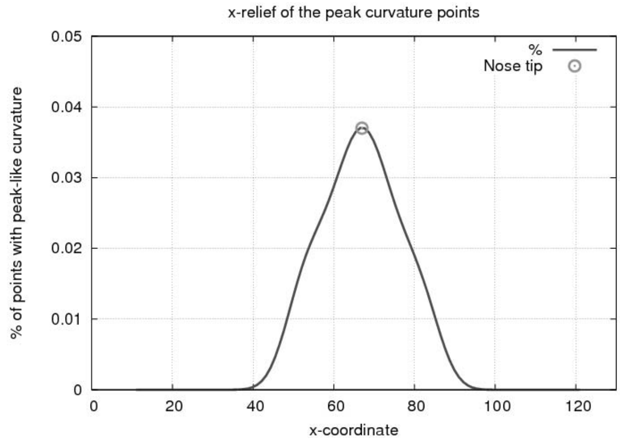


Figure 6. Example of nose tip x-coordinate detection.

From the detected nose tip, we can easily find the nose corners, which are the maximum variations in the horizontal profile curve. The horizontal profile curve is the x-projection that represents the set of points with same y-coordinate value, in this case, the nose tip y-coordinate. To detect the nose corners, we calculate the gradient information of this curve and look for one peak on each side of the nose tip, as shown in Fig. 7.

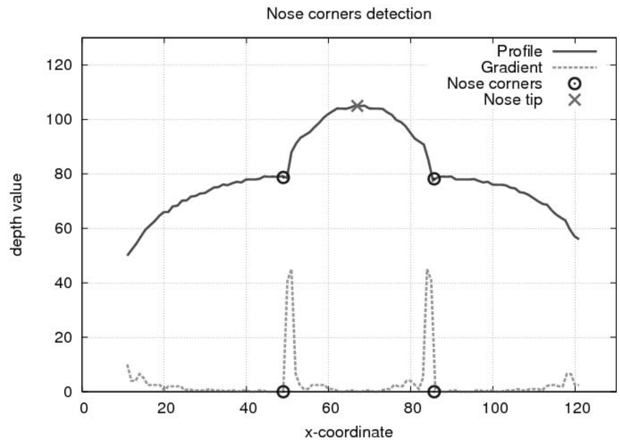
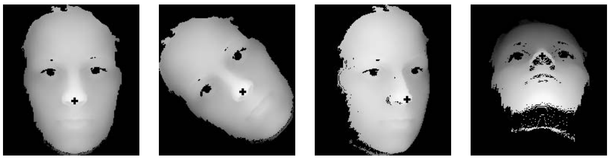


Figure 7. Example of nose corners detection.

Considering a face surface with no pose variation, we have the nose tip normal vector aligned with the z-axis. Variations in the profile-to-profile face movement are y-axis rotations. Frontal and back movements of the face (see

Fig. 8(d)) are x-axis rotations. We tested the nose tip detection under extreme pose variations, and we obtained the following results: for z-axis rotation we obtained correctly detections between -45 and 45 degrees, for y-axis between -25 and 25 degrees, and for x-axis between -55 and 55 degrees, as shown in Fig. 8. These results show the robustness of our method for nose tip detection across significant pose variations.



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

Figure 8. Examples of nose tip detection with (a) no pose variation, (b) z-axis rotation of 45 degrees, (c) y-axis rotation of 25 degrees and (d) x-axis rotation of -55 degrees.

3.3. Eye corners detection

To locate the eye corners, we compute another y-projection using the curvature image. We determine the percentage of pit curvature points of every set of points with the same y-coordinate. For a face, this projection generally presents three peaks, representing eyes, nose base and mouth. As the nose coordinates are known, we can easily assign each peak to its respective facial feature (See Fig. 9).

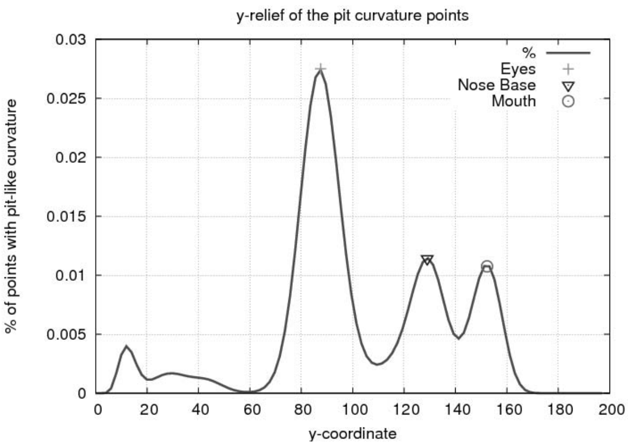


Figure 9. Example of eye y-coordinate detection.

To find the x-coordinates of the eye corners, we compute another x-projection of the curvature image, but in this case we calculate the percentage of the pit curvature points of 7 neighbor rows centered in the eye y-coordinate of every column. The left eye corner x-coordinate is the beginning of the first peak in this x-projection, and the right eye corner x-coordinate is the end of the second peak in the same projection, as shown in Fig. 10.

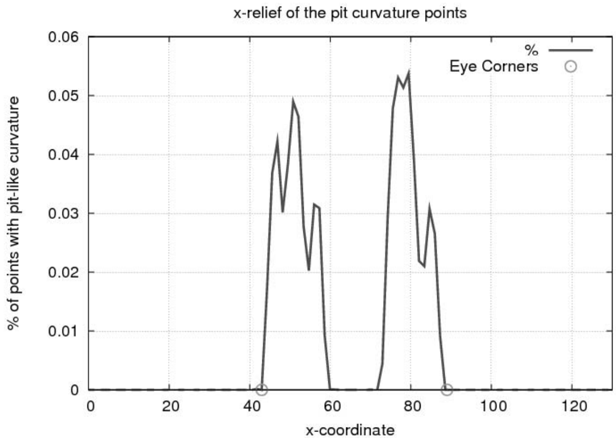


Figure 10. Example of eye corners detection.

4. Experimental results

In our face segmentation experiments, we used the two versions of Face Recognition Grand Challenge database (FRGC 1.0 and FRGC 2.0), with a total of 4,950 images of 557 subjects, containing variations of facial expression, scale, pose and other characteristics, like hairstyle[2]. The images have size 640×480 and were acquired by using a laser sensor Minolta Vivid 900[3].

These databases were chosen to support our experiments because they have been extensively used for researches regarding 3D face recognition. Our segmentation approach extract the face region in all images of the databases, but around 1% of the results (61 of 4,950) include some irrelevant parts, like hair and neck. However, these undesired parts do not affect our recognition process, as confirmed by visual inspection. Fig. 11 shows some examples of the final face segmentation.

Our segmentation approach was enbedded in a 3D face recognition system [2], and produces a verification rate higher than 99% with a false acceptance rate (FAR) of 0% for neutral expression faces. The average time to segment a face in range images was 1.1 seconds in a Pentium D 3.4 GHz, and variate with the face size.

[2]The authors would like to thank Dr. Jonathon Phillips and Dr. Patrick Flynn for allowing us to use the images.

[3]`http://www.minoltausa.com/vivid`

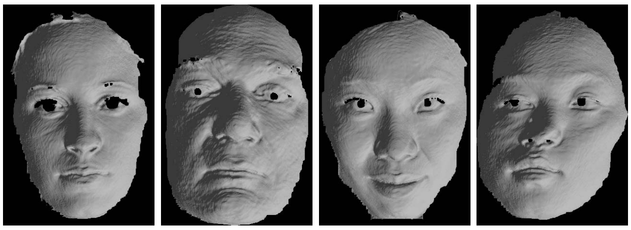


Figure 11. Face segmentation results using our approach.

For facial features detection experiments, we used only the FRGC 2.0 database, with 4,007 images of 466 subjects. The time average spent to locate all desired facial features in a segmented face image was 0.4 seconds. The results of the features detection are presented in Table 2, followed by some examples in Fig. 12.

Features	% of correctly detection
Nose tip	99,95
Nose corners	99,76
Nose base	99,98
Eye corners	99,83

Table 2. Correctly detection rate for each facial features.

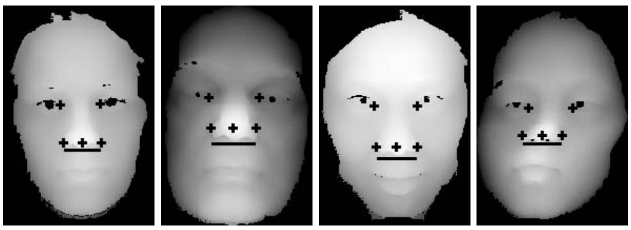


Figure 12. Facial features detection.

The nose tip was wrong detected in only two images of the database, because those images have no depth information in the nose area, and consequently have no surface curvature information, affecting the profile curve. One of those images had a small lack of depth information, but it still allows the detection of the nose base, which was wrong detected in only one image of the database. Nose and eye corners detection presented some problems when the face has a significant pose variation ($>$ 15 degrees in y and z-axis).

By evaluating the effectiveness of this approach in our face recognition system [2] using faces with expression we observed an improvement around 20% in the matching score. Details of this improvements are omitted because of space limits.

5. Conclusions

In this paper, we presented a new method for face segmentation and facial features detection, which has been successfully applied in our 3D face matching system [2]. We demonstrated the effectiveness of our approach, which segmented the face and detected the facial features correctly in almost 100% of the tested face images. Moreover, we presented a robust method for nose tip detection in 3D face images in presence of significant pose variation. As a future work we plan to increase the number of facial features to be detected, and to use them to identify facial expressions.

References

[1] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[2] O. Bellon, L. Silva, C. Queirolo, S. Drovetto, and M. Segundo. 3D face image registration for face matching guided by the surface interpenetration measure. In *13th ICIP*, pages 2661–2664, 2006.

[3] O. R. P. Bellon and L. Silva. New improvements to range image segmentation by edge detection. *IEEE Signal Processing Letters*, 9(2):43–45, 2002.

[4] P. Besl. *Surface in Range Images Understanding*. Springer-Verlag, 1988.

[5] K. Chang, K. Bowyer, and P. Flynn. Adaptive rigid multi-region selection for handling expression variation in 3d face recognition. In *Proc. of IEEE Workshop Face Recognition Grand Challenge Experiments*, 2005.

[6] K. Chang, K. Bowyer, and P. Flynn. Multiple nose region matching for 3d face recognition under varying facial expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1695–1700, 2006.

[7] D. Colbry, X. Lu, A. Jain, and G. Stockman. 3d face feature extraction for recognition. In *Michigan State University Technical Report*, 2004.

[8] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*, volume 2 edition. Addison-Wesley, 1992.

[9] P. Gotardo, K. Boyer, O. R. P. Bellon, and L. Silva. Range image segmentation into planar and quadric surfaces using an improved robust estimator and genetic algorithm. *IEEE Transaction on Systems, Man, and Cybernetics*, 34(6):2303–2316, 2004.

[10] A. K. Jain and R. C. Dubes. Algorithms for clustering data. Prentice-Hall, 1988.

[11] L. G. Shapiro and G. C. Stockman. Computer vision. Prentice-Hall, 2001.

[12] K. Sobottka and I. Pitas. A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing-Image Communication*, 12(3):263–281, 1998.

[13] F. Tsalakanidou, S. Malassiotis, and M. G. Strintzis. Face localization and authentication using color and depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):152–168, 2005.

[14] Y. Xie and Q. Ji. A new efficient ellipse detection method. In *Proc. of 16th International Conference on Pattern Recognition*, volume 2, pages 957–960, 2002.