Robust Realtime Feature Detection in Raw 3D Face Images

Ajmal Mian
Computer Science and Software Engineering
The University of Western Australia
35 Stirling Highway, Crawley WA 6009, Australia
ajmal@csse.uwa.edu.au

Abstract

3D face data contains holes, spikes and significant noise which must be removed before any further operations such as feature detection or face recognition can be performed. Removing these anomalies from the complete data is expensive as it also contains non-facial regions. We present a realtime algorithm that can detect the eyes and the nose tip in raw 3D face images in about 210 msecs. With three points, the data can be aligned to a canonical pose or registered to a reference face allowing the face area to be accurately cropped. The more expensive preprocessing steps can then be applied to the cropped region of the face only. We calculate the x and y gradients from the range image and train separate feature detectors in the three representations. Each detector is trained using the AdaBoost algorithm and Haar-like features. Haar features detect higher order discontinuities in the gradient images which form the core of the proposed algorithm. Multiple feature detections in the three images are clustered and anthropometric ratios are used to eliminate outliers. The centroids of the remaining candidates are used as feature points. Experimental results on the FRGC v2 database gave over 99% detection rates. Detailed quantitative analysis and comparison with the ground truth feature locations is provided.

1. Introduction

Three-dimensional face data has been widely used for human identification because it has two main advantages. Firstly, the facial pose can be more reliably corrected using 3D data and secondly, the 3D data is illumination invariant as it represents the shape rather than surface reflectance[1]. The raw data obtained from even the most accurate 3D scanners is far from perfect as it contains spikes, holes and significant noise. A number of preprocessing steps must be applied to remove these anomalies before any further operations can be performed.

Spikes are caused mainly by specular regions and are generally detected and removed as a first preprocessing step. The eyes and the nose tip are the main regions where spikes are likely to occur. Unfortunately, these are also the main facial landmarks used for face alignment. Glossy facial makeup can also cause spikes at other regions of the face. Spike detection works on the principle that surfaces (and faces in particular) are generally smooth. Therefore, points that are far from their neighbors along the viewing direction are regarded as outliers and removed.

In addition to the holes resulting from spike removal, the 3D data contains many other missing points due to dark regions and occlusions. Areas of the scene that are not visible to the camera or the laser cannot be acquired. Similarly, less reflective regions cannot be sensed by active scanners. Both can cause large regions of missing data. The laser power cannot be increased to acquire dark regions of the face due to safety reasons. Thus the only option is to fill the missing regions using an interpolation technique such as linear, nearest neighbor, polynomial etc. For small holes, linear interpolation gives reasonable results however, bicubic interpolation has shown to give better results [16]. For larger size holes, a model-based approach can be used to morph a model until it gives the best fit to the data points [11]. Interpolation and morphing are both computationally expensive and the latter also requires the identification of reference points to register the data to the morphable model.

The third type of error in 3D data is noise. In laser scanners, noise can be attributed to optical components such as the lens and the mirror, or mechanical components which drive the mirror or the CCD itself. Scanning conditions such as ambient light, laser intensity, surface orientation, texture, and distance from the scanner can also affect the noise levels in the scanner. Sun et al. [18] give a detailed analysis of noise in the Minolta Vivid scanner. Low pass filter and median filters have been used for noise removal.

The above three preprocessing steps must be applied to 3D face data before any other operation (e.g. feature point detection or face recognition) can be reliably performed.

[1] The acquisition of 3D data may not be illumination invariant.

978-1-4244-9497-2/10/$26.00 ©2010 IEEE

Moreover, the 3D data usually contains more than just the face region because the scanner also acquires the hair, ears, shoulders and clothing. Although most scanners additionally acquire a registered texture map which can be used to detect the face [15] and then crop the corresponding region in the 3D data, there is a trade off between detection speed and accuracy of localization. Feature detection in the 3D face image is preferred for better localization and for robustness to illumination and other appearance changes e.g. due to facial makeup. Texture based eye detection can fail if the eyes are closed. Finally, it is of theoretical interest to study feature detection in 3D face images.

This paper focuses on realtime detection of three feature points, namely the eyes and the nose tip, in raw 3D images containing spikes, holes and noise. With three feature points, the 3D face can be normalized to a canonical pose and scale. Pose and scale normalization is necessary because they can vary between scans depending upon the subjects orientation and distance relative to the scanner. The three points can also be used to register the data to a reference face so that the facial region can be accurately cropped. The above preprocessing steps can then be applied to the cropped face only. In a morphable model based approach such as [2][11], it may not even be necessary to explicitly perform the above preprocessing steps i.e. the model can be morphed to fit the raw data using a robust approach.

The proposed approach is based on Haar-like features and the AdaBoost algorithm used by Viola and Jones [19] for face detection. Our contribution is that we use three representations of the raw 3D face data to train six different classifiers for feature detection. The three representations include the range image and its x and y gradients. The feature points detected by different classifiers are clustered and outliers are removed. Statistical anthropometric relationships between the eyes and the nose tip are used to identify the correct eyes and nose clusters. The centroids of the candidates in each cluster are taken as the feature points. The proposed approach is invariant to facial expressions and robust to minor pose variations. Experiments were performed using the FRGC v2.0 database [17] containing 4007 three dimensional face images and over 99% detection rates was achieved for each feature. The average detection time was 210 msecs on a 2.4GHz machine.

1.1. Related Work

In 3D face images, the nose is perhaps the most prominent feature that has been automatically detected and used to localize or align the face. Mian et al. [16] first detected the nose ridge in horizontal slices of the 3D face using altitude of a triangle inscribed inside a moving circle along the slice. Outlier ridge points were then removed using RANSAC and the point associated with the longest altitude triangle was declared the nose tip. For pose correction, Mian et al. [16] used the nose tip alone and an iterative PCA algorithm which is computationally expensive. Even though the horizontal slices were taken from raw 3D data, they were interpolated to fill missing points before ridge detection.

Colombo et al. [5] detected the nose tip and the eyes using principal curvatures. The candidate triplet was then used by a PCA based classifier for face detection. They report 82% detection rate using 150 scans and mention that the failures were mostly due to holes in the eyebrows. They suggest including a preprocessing step to improve the results. Lu et al. [14] used the shape index [6] to automatically detect the inside eye corners and the nose tip for facial pose correction. They do not report the accuracy of detection. Since shape index is derived from principle curvatures, this technique is sensitive to noise, holes and spikes. Therefore, it requires preprocessing of the data.

Gupta et al. [8] detected 10 anthropometric fiducial points to calculate cranio-facial proportions [7]. However, only three points (nose tip and points on the outer edge of the nose) were detected using the 3D face data and the rest were detected based on 2D and 3D data combined. To detect the nose tip, they first register the query face to a nose template using the ICP [1] algorithm and then refine the nose tip localization using Gaussian and mean curvatures. They claim that the region around the nose tip has the highest elliptic and convex elliptic Gaussian curvature. Note that the ICP algorithm is already computationally expensive and the calculation of curvatures require preprocessing the data making this approach unfeasible for realtime detection.

Colbry et al. [4] found that preprocessing the scan before nose tip detection increases the success rate of the detector from 92% to 97%. They used the shape index [6] and a number of heuristics to detect anchor points on the eyes, nose, mouth and chin. They report 82.7% accuracy for 3D faces with neutral expression and 75% for non-neutral expression. Koudelka et al. [12] argue that the nose and eyes are radially symmetric and can be detected using the radial symmetry transform [13]. They detect five feature points namely the nose tip, sellion, inner eye corners, and center of the mouth. Using the FRGC v1.0 database, Koudelka et al. [12] report that 97% of the extracted features are within 10 mm of the manually marked ground truth. They also preprocess the data for removing spikes and filling holes before feature detection and report 2.5 seconds run time using Matlab on a 3.4 GHz machine.

Chang et al. [3] perform a number of preprocessing steps including morphological operations, spike removal and skin detection (using texture) to localize the face. Next they define local coordinate bases using tangent plane and fit a quadratic surface to the local region. The mean and Gaussian curvatures are then calculated from the parameters of the fitted surface. The nose tip and eyes are detected based

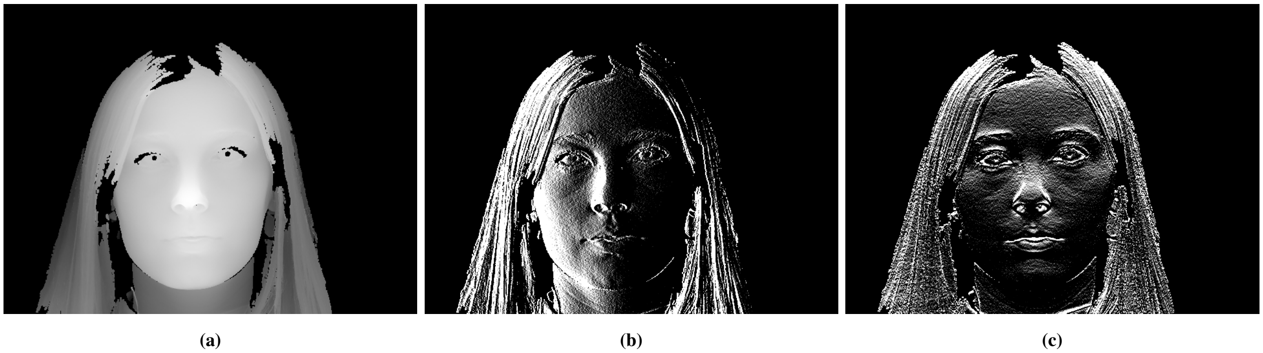


Figure 1. A sample rage image (a) and its corresponding x (b) and y (c) gradient images. Holes can be seen in all images whereas noise is more obvious in the gradient images.

on these curvatures. They report 99.4% detection accuracy using 4458 scans. Wang et al. [20] avoid 3D feature selection in dark or specular regions like the eyebrow and eyes. They construct individual PCA subspaces for four feature points including the nose tip using training data. Each point on the query face is projected to this subspace and the closest one is regarded as the feature. This process is computationally expensive especially if the 3D image contains more than just the face. Husken et al. [9] detected feature points on the 2D image and mapped them to the 3D image for Hierarchical Graph Matching. Besides being a texture based approach, they do not report the accuracy of feature detection.

Our survey shows that existing 3D facial feature detection techniques either perform heavy perprocessing steps leading to slow detection speed or have a very low detection accuracy. The main significance of our approach is that it is efficient and accurate at the same time. The latter is achieved by detecting features in multiple representations of the raw 3D face (see Fig. 1) and then clustering them.

1.2. Theoretical Justification

Fig. 2 shows the basic Haar-like features i.e. two edge features, two line features and a center-surround feature. A feature value is the sum of the pixels in the white region minus the sum of pixels in the dark region which can be efficiently calculated from the integral image [19]. Haar features basically detect discontinuities e.g. edges and lines in an image. In a range image, Haar features detect depth discontinuities which are small (and hence noisy) for a smooth surface like the face. In x and y gradient images, Haar features measure the discontinuities in the derivatives along the x and y dimension respectively which are more profound. For example, there is minor depth discontinuity between the eye lid and the eye whereas the discontinuity in its y gradient is significant and can be seen more obviously in Fig. 1-c. Using gradient images along two dimensions gives a richer set of Haar features that capture horizontal and vertical inflection lines. Therefore, the classifier achieves better detection rate and more accurate localization of the features points (see results).

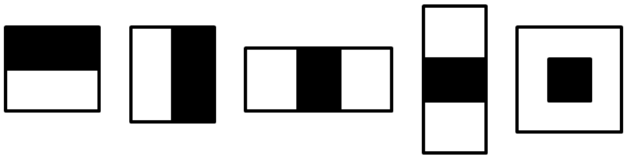


Figure 2. Representative Haar features.

2. Proposed Algorithm

The input to our algorithm is a raw 3D image (i.e. range image) from a scanner. We calculate further two representations from the range image namely the gradient images in the x and y directions. Note that we do not preprocess the input data to remove spikes and nose or to fill holes. Therefore, all anomalies present in the input range image are also reflected in the gradient images. Fig. 1 shows a sample range image and its corresponding gradient images.

We used the FRGC v2.0 data [17] for our experiments which provides 953 3D face scans for training and another 4007 face scans for testing. We trained the Haar classifier cascade with the first 500 3D face images from the training data by manually identifying two points for each feature. We trained classifiers for the nose and the right eye leaving the left eye to be detected in the mirrored image during run time. The number of training samples for the right eye as well as the nose were doubled (i.e. 1000) by considering the horizontally flipped version of the image as well.

Since the FRGC v2.0 provides ground truth for the outer corners of the eyes in the test data, we take that as our feature point of interest and manually identify it for the right eye in the training data. We additionally identify the inner corner of the eye to determine the scale of the feature window. Since it is not compulsory to center the feature window over the point of interest (the outer corner of the

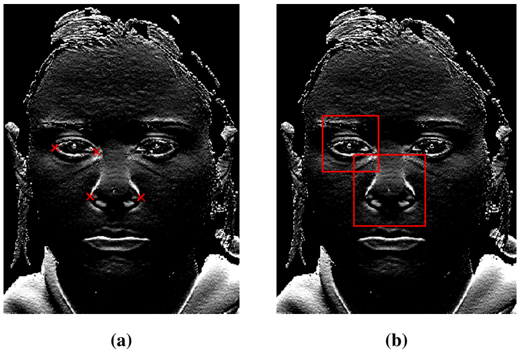


Figure 3. Generating training data for right eye outer corner and nose tip detection. (a) Two points each (more obvious in color) are manually identified on the right eye and the nose. (b) The corresponding feature windows.

eye), we horizontally center the window at the mid point between the inner and outer corner of the eye. Moreover, we shift the window up by a fraction of the scale (i.e. 20%) so that it includes the eyebrows. This positioning of the feature window around the eye ensures that maximum features are included while the window is still tight. In our experiments, we used a square window with sides equal to $\frac{4}{3}$ times the distance between the inner and outer corners of the eye. Fig. 3-a shows the two points identified on a training face and the corresponding window is shown in Fig. 3-b.

Our second feature point of interest is the nose tip. Various definitions of the nose tip exist in the literature e.g. using the height or the principle curvatures. These definitions are quite sensitive to noise and would be valid only after the data has been preprocessed. Since our data is noisy and we would like to avoid preprocessing before feature detection, we define the nose tip as the mid point between the outer most horizontal edges of the nose. These edges can be easily identified on the y gradient of the range image (see Fig. 3-a). Once again a square window is chosen with each side equal to $\frac{4}{3}$ times the distance between the identified points. The window is horizontally centered on the nose tip but shifted up by a fraction of its scale (i.e. 20%) to include more of the nose and avoid the lips which can significantly change during expressions.

The right eye and the nose are manually identified in 500 training images (y gradient) and their mirrored versions. The two feature points in the corresponding locations of the range image, the x gradient and the y gradient images are then used to train different detectors for the right eye outer corner and the nose tip. Therefore, a total of six detectors were trained.

It is worth mentioning that while training the right eye detector, the negative samples generated from the training images did not contain the left eye as well. We wanted the detector to differentiate the right eye from the rest of the face, hair and clothing but not from the left eye. Otherwise, due to the similar shape of the left and right eyes, the detector would have discarded many potentially useful features. Consequently, our right eye detector worked quite well in terms of true positives but at the cost of many false positives on the left eye. However, this was not a problem as the false positives on the left eye would be easily removed during the second stage of the combined feature detection.

We used the Haar classifier cascade provided with the OpenCV library [10] to train these six feature detectors. During testing, we used the classifiers to detect the two feature points separately in the three image representations. We found that the x gradient image did not provide any improvement for the right eye corner detection. Therefore, the eye corner detection was performed using only the range image and the y gradient image. To detect the left eye outer corner, the two images were horizontally flipped and a second pass of detection was performed.

The nose tip detection was performed using all three representations. Moreover, since two of the images were already flipped for eye detection, a second pass of nose tip detection was performed over these images. In total, there were two detection attempts for each eye and five detection attempts for the nose tip. The main reason for this imbalance in the number of detection attempts is because the eyes are richer in Haar-like features compared to the nose tip. This is in contrast to geometric techniques which are able to detect the nose more reliably than the eyes.

The feature detectors give many (1 to 6 each) candidate outer eye corner and nose tip locations along with their labels. Each detection provides a location and scale of the feature. The scale is used to undo the shift of the feature point location performed during labeling of the training data. The candidates are clustered to eliminate obvious outliers. However, clustering may not be helpful if there is only one candidate for a feature or in the unlikely case of widely placed candidates. Therefore, in the next step we fit a triangle to all possible combinations of the right eye, left eye and nose tip. Anthropometric constraints [7] are used to quickly eliminate incorrect triangles. Out of the remaining triangles the one which provides the best fit to anthropometric ratios is taken as the final choice. The corners of this triangle serve as the likely locations of the eyes and the nose tip. Candidate points that are relatively far from their respective triangle corner are removed. This step removes poorly localized candidate points around the nose tip and the eyes. The centroid of the remaining points is taken as final location of the feature point. Fig. 4 shows the candidate feature points after initial detection, after clustering and fitting a triangle and the final points along with the ground truth.

3. Results

Experiments were performed on the FRGC v2.0 test data containing 4007 face scans. Fig. 4 shows qualitative results

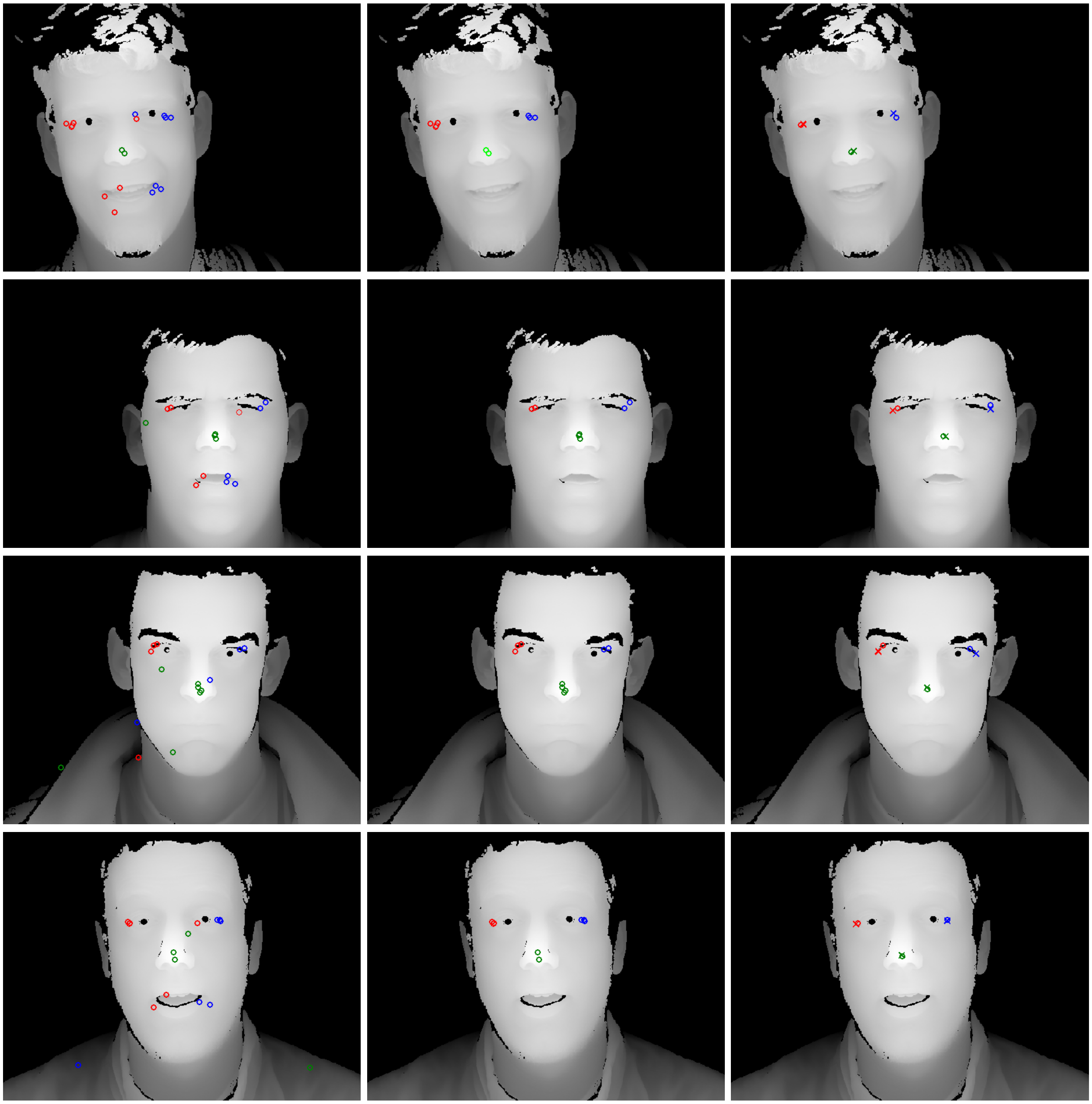
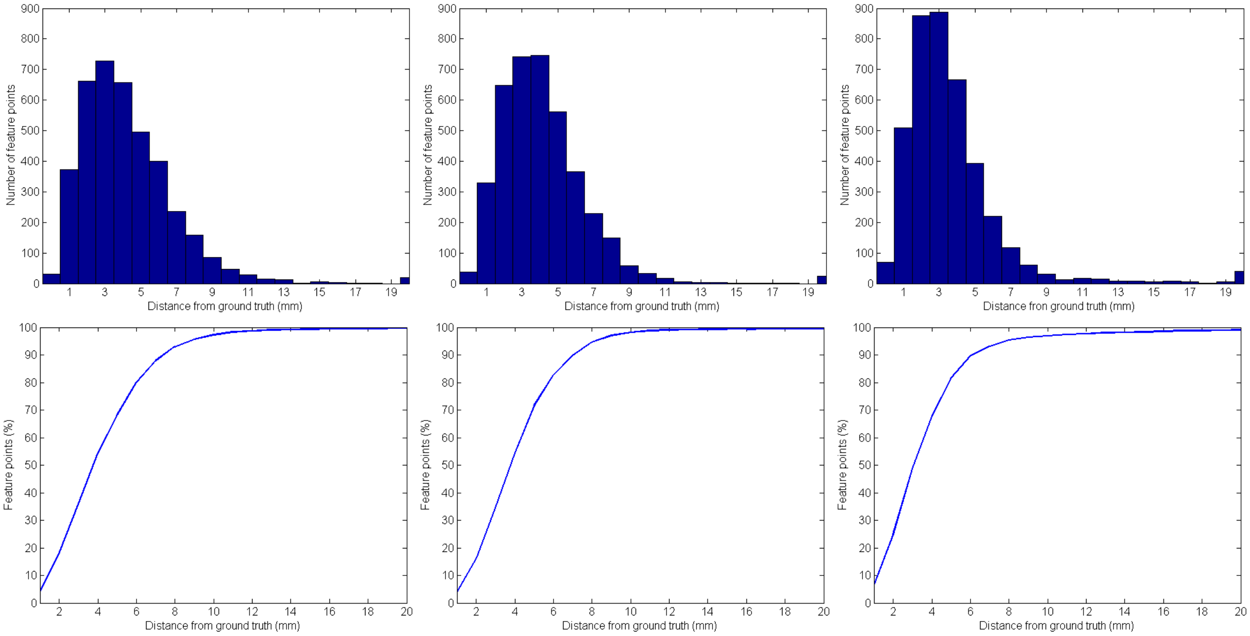


Figure 4. First column: Candidate feature detection. Candidates for the right and left eye outer corners are marked with red and blue circles respectively. The nose tip candidates are colored green. Second column: Outliers are removed after clustering and fitting a triangle. Third column: The final feature points along with the ground truth marked as crosses of the same color.

of our algorithm at each step. Note that features are correctly detected in the presence of pose variations, holes and expressions such as squinched eyes (second row) and raised eyebrows (last row). Localization of features is almost perfect with respect to the ground truth except in the third row where significantly large holes in eyebrows have caused a small error in the localization of the outer eye corners.

Fig. 5 shows quantitative results of our algorithm by measuring the Euclidean distance between detected features and the ground truth provided with the FRGC v2.0 database. In the top row, histograms of errors (in mm) are given whereas the second row plots the percentage of feature points that are below a certain distance from their ground truth. Table 1 summarizes our results. The detection rates for the right and left outer eye corners was 99.7% and 99.9% respectively. The nose tip detection rate was 99.2%.



(a) Right eye **(b) Left eye** **(c) Nose tip**

Figure 5. Feature detection results for the (a) right eye, (b) left eye and (c) nose tip using 4007 test scans from the FRGC v2.0 database. Top row: Histogram of distance errors from ground truth. Bottom row: Cumulative percentage of features that were within a certain distance from the ground truth.

Fig. 6 show a case of failure for each feature. Outer eye corner detection failed mostly due to occlusions from hair where other algorithms are also likely to fail even after preprocessing the data. The holes in and around the eyes did not effect its detection but had a minor effect on the localization of the outer corner. Nose detection failures were due to excessive holes and due to the fact that the nose region is not as rich in Haar-like features as the eyes.

The average localization error (from ground truth) for the outer eye corners was 4.4 mm. The average localization error for the nose tip was lower however, its standard deviation was high. A possible explanation for this is that there are variations in the ground truth nose tip locations. The nose tip is not as conspicuous as the eye corners and therefore, cannot be perfectly manually identified in an image. For comparison, we measured the performance of the range image based feature detection alone and it achieved an overall detection rate of 87.6%.

Table 1. Feature detection results using 4007 test scans from the FRGC v2.0 database.

	Right eye	**Left eye**	**Nose tip**
Detection rate (%)	99.68	99.90	99.18
Avg error (mm)	4.39	4.38	4.03
Std deviation (mm)	4.49	5.80	8.23
No. of failures	13	4	33

4. Conclusion

This paper presented a realtime algorithm for eye corner and the nose tip detection in raw 3D face scans. We save computation time by avoiding preprocessing steps such as spike removal, hole filling and noise filtering. We achieve realtime detection by using classifier cascades and Haar-like features which are efficiently calculated from the integral image. Accuracy of feature detection and localization is improved by clustering multiple detections in three different representations i.e. the range image and its x and y gradient images. The gradient images allows the extraction of richer Haar features. Unlike previous techniques, we give detailed quantitative results in terms of accuracy of feature localization using the largest available 3D face database. Our results show that the proposed algorithm is fast, accurate, invariant to facial expressions and robust to minor pose variations.

5. Acknowledgements

Thanks to FRGC organizers for the 3D face data [17]. This research was supported by ARC grant DP0881813.

References

[1] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. on PAMI*, 14(2):239–256, 1992. 2

[2] V. Blanz, K. Scherbaum, and H. Seidel. Fitting a morphable model to 3D scans of faces. In *CVPR*, pages 1–8, 2007. 2

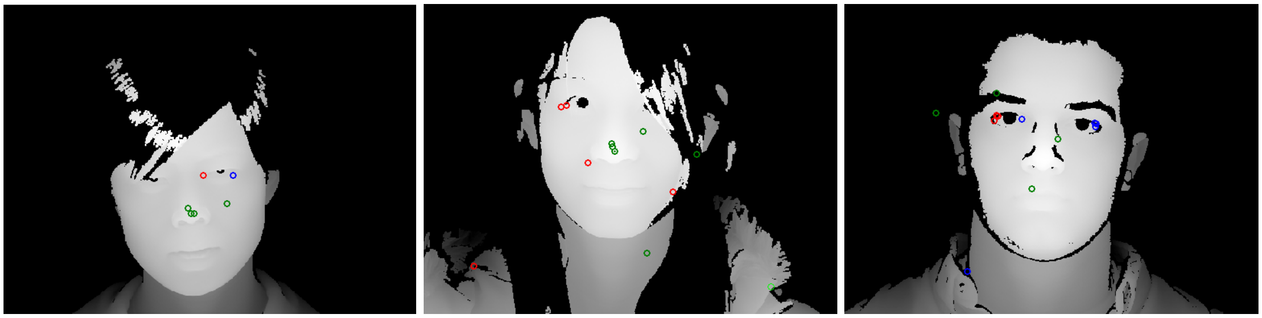


Figure 6. Examples of detection failures for the right and left eye outer corners and the nose tip (better seen in color).

[3] K. Chang, K. Bowyer, and P. Flynn. Multiple nose region matching for 3D face recognition under varying facial expression. *IEEE TPAMI*, 28(10):1695–1700, 2006. 2

[4] D. Colbry, G. Stockman, and A. Jain. Detection of anchor points for 3D face verification. In *CVPR Workshops*, page 118, 2005. 2

[5] A. Colombo, C. Cusano, and R. Schettini. 3D face detection using curvature analysis. *Pattern Recognition*, 39(3):444–455, 2006. 2

[6] C. Dorai and A. Jain. COSMOS-A Representation Scheme for 3D Free-Form Objects. *IEEE TPAMI*, 19(10):1115–1130, 1997. 2

[7] L. Farkas. Anthropometry of the head and face. *Raven Press*, 1994. 2, 4

[8] S. Gupta, M. Markey, and A. Bovik. Anthropometric 3D Face Recognition. *Int'l Journal of Computer Vision*, 90(3):331–349, 2010. 2

[9] M. Husken, M. Brauckmann, S. Gehlen, and C. V. der Malsburg. Strategies and benefits of fusion of 2d and 3d face recognition. 3, 2005. 3

[10] Intel. Open computer vision library. *http://www.intel.com/technology/computing/opencv/, 2010.* 4

[11] I. Kakadiaris, G. Passalis, G. Toderici, M. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-Dimensional Face Recognition in the Presence of Facial Expressions: An Annotated Deformable Model Approach. *IEEE TPAMI*, 29(4):640–649, 2007. 1, 2

[12] M. Koudelka, M. Koch, and T. Russ. A prescreener for 3D face recognition using radial symmetry and the Hausdorff Fraction. In *CVPR Workshops*, pages 1–8, 2005. 2

[13] G. Loy and A. Zelinsky. Fast Radial Symmetry for Detecting Points of Interest. *IEEE TPAMI*, 25(8):959–973, 2003. 2

[14] X. Lu, A. Jain, and D. Colbry. Matching 2.5D scans to 3D models. *IEEE TPAMI*, 28(1):31–43, 2006. 2

[15] M. Yang and D. Kriegman and N. Ahuja. Detecting faces in images: A survey. *IEEE TPAMI*, 24(1):34–58, 2002. 2

[16] A. Mian, M. Bennamoun, and R. Owens. An Efficient Multimodal 2D-3D Hybrid Approach ot Automatic Face Recognition. *IEEE TPAMI*, 29(11):1927–1943, 2007. 1, 2

[17] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the Face Recognition Grand Challenge. In *IEEE CVPR*, pages 947–954, 2005. 2, 3, 6

[18] X. Sun, P. Rosin, R. Martin, and F. Langbein. Noise analysis and synthesis for 3D laser depth scanners. *Graphical Models*, 71(2):34–48, 2009. 1

[19] P. Viola and M. J. Jones. Robust Real-Time Face Detection. *IJCV*, 57(2):137–154, 2004. 2, 3

[20] Y. Wang, C. Chua, and Y. Ho. Facial feature detection and face recognition from 2d and 3d images. *Pattern Recognition Letters*, 23(10):1191–1202, 2002. 3