Face Recognition by 3D Registration for the Visually Impaired Using a RGB-D Sensor

Wei Li[1], Xudong Li[2], Martin Goldberg[3] and Zhigang Zhu[1,3]

[1] The City College of New York, New York, NY 10031, USA
[2] Beihang University, Beijing 100191, China
[3] The CUNY Graduate Center, New York, NY 10016, USA
lwei000@citymail.cuny.edu, xdli@buaa.edu.cn,
mgoldberg@gc.cuny.edu, zhu@cs.ccny.cuny.edu

Abstract. To help visually impaired people recognize people in their daily life, a 3D face feature registration approach is proposed with a RGB-D sensor. Compared to 2D face recognition methods, 3D data based approaches are more robust to the influence of face orientations and illumination changes. Different from most 3D data based methods, we employ a one-step ICP registration approach that is much less time consuming. The error tolerance of the 3D registration approach is analyzed with various error levels in 3D measurements. The method is tested with a Kinect sensor, by analyzing both the angular and distance errors to recognition performance. A number of other potential benefits in using 3D face data are also discussed, such as RGB image rectification, multiple-view face integration, and facial expression modeling, all useful for social interactions of visually impaired people with others.

Keywords: Face Recognition, Assistive Computer Vision, 3D Registration, RGB-D Sensor

1 Introduction

Visually impaired people have difficulty in recognizing people in their daily life, therefore their visual impairment greatly limits their social lives. Without hearing voice or getting close enough to touch a person, it is hard for them to know exactly who the person is and how the person looks like. We would like to help them in this kind of situation by employing 3D computer vision based methods. By using face recognition techniques and a popular RGB-D camera, we can build computer vision system to detect and recognize people in different scenes, and then inform the visually impaired users of the recognition results.

In order to help a visually impaired person recognize people, we utilize a Kinect RGB-D sensor and propose a 3D registration based approach to face recognition. First, real-time face tracking is realized using the Kinect SDK and 3D facial feature points are extracted. The 3D pose (position and orientation) of the sensor to the observed face is estimated. Then the current 3D face (the probe) is compared with all the 3D faces in a database (the gallery) by using a one-step Iterative Closest Points (ICP) approach. The *registration* errors of the probe face and the gallery faces indicate the uncertainty in face matches: the smaller the error, the more similar the two matched faces. By picking up the gallery face that has the smallest registration error with the probe face, the probe face is recognized.

The major goal of this work is to investigate what can be done with only 3D data for face recognition, and this paper has the following four contributions. 1) A highly effective ICP registration based approach is proposed for face recognition. Since a very good estimation of each face's pose is known, the initial estimation of the relative pose between the two faces (for the same person) in comparison is fairly close to the true value and therefore a one-step ICP will be able to obtain an accurate registration. For faces of different people, the approach just assume they are of the same person and the registration error of the one-step ICP indicates they are of two different faces. 2) To evaluate the robustness of the approach, an error analysis is performed by adding various degrees of errors to the *ideal* 3D face data in order to determine the required accuracy in 3D estimation of face features using the RGB-D sensor. This is compared against the accuracy of a Kinect sensor used in our experiments to evaluate the feasibility of 3D face recognition using the sensor. 3) An online recognition system prototype is built with the goal to help the visually impaired recognize people in front of them. We have also designed experiments to analyze how the system performs with various standoff distances and face orientations. 4) A number of other benefits in using 3D face information are explored, such as RGB image rectification and normalization, multiple-view face integration, and facial expression modeling, which are all useful for assisting the interactions of visually impaired people with others.

The paper is organized as follows. Section 2 introduces the background and related work. Section 3 describes the basis of our approach – 3D face feature extraction and the ICP-based face registration. In Section 4 we describe our ICP-based face recognition approach, evaluate the error tolerance of our approach, and analyze the measuring errors of the Kinect sensor. In Section 5, we provide the testing results using the proposed approach in online recognition. Then in Section 6, we discuss a number of other possible uses of the 3D face data for face modeling. Finally we conclude our work in Section 7.

2 Related Work

Face recognition has already been a topic in research for decades; researchers have gone a long way from 2D to 3D image approaches to achieve the goals of using face recognition for various applications. A lot of 2D image based face recognition methods have been proposed, and these methods have been roughly divided in to two categories: feature based and holistic based [6]. Among these methods, the LBP (Local Binary Pattern) based method [16] and Gabor based method seem to be the most outstanding ones [8] [9]. However, 2D methods face great difficulty when dealing with the problem of face pose variation in real life.

Although researchers have tried to find the facial features robust to face pose variation, or align faces by using the features extracted [4] [18] [1], it is still an open problem for applications when face pose information is not available.

Another promising branch to solving the problem is using the 3D data of a RGB-D sensor. Comparing to using only 2D images, 3D data is invariant to the face pose (both position and orientation). Therefore, if we can obtain accurate 3D data, then 3D data based methods are supposed to be more appealing than 2D methods in real applications. However, due to various face poses, we may not be able to see the full view of face. The authors in [10] proposed a method to use facial symmetry to handle the face pose problem; each face is assumed to be symmetry so if part of the face is hidden, we can still obtain the full detail by using its symmetry.

Many approaches have been proposed to work on the common dataset FRGC V2[11]. In [5] both MS-LBP (Multiple scale Local Binary Patten) depth maps and Shape Index Maps are employed, then the Scale-Invariant Feature Transform (SIFT) is used to extract local features from the maps. Then the local and holistic analysis are combined to perform the matching of faces. In [12] an registration method is proposed, using an approach based on Simulated Annealing (SA) for range image registration with the Surface Interpenetration Measure (SIM). The SA starts the registration process using the M-estimator. Sample Consensus proposed in [14] is used and a final tuning is achieved by utilizing the SIM. The resulting SIM value is then used to assess the final 3D face matching. [15] uses an isogeodesic stripes graph descriptor obtained from a 3D Weighted Walk to represent a 3D face and transformed the 3D face matching to a graph matching. [2] introduces a Signed Shape Difference Map (SSDM) to compare aligned 3D faces, both the interpersonal and intrapersonal SSDMs are trained to distinguish different people and find their common features. For each test, the candidates SSDM with all the saved models are computed and matching scores are obtained by comparing with trained models to recognize the candidate.

Kinect has become a very popular RGB-D sensor, since it can provide the regular RGB images as well as Depth images in real time, therefore it is natural to integrate the two kinds of data. Using the 3D information, we can easily obtain the human face features in both 2D and 3D images. A number of researchers have worked on Kinect-based 3D face recognition. For example, [7] presents an algorithm that uses a low resolution 3D sensor for robust face recognition under challenging conditions. The method first locates the nose tip position of a face, then uses the position to find the face region. Sparse coding is used to encode the face, and then the encoded vector of the face is used as the input to a classifier to recognize the face. But due to Kinect depth measurement error, a reliable tip localization method is still in need.

3 3D Face Feature Extraction and Registration

Most of the methods discussed above can obtain good face recognition performance in experiments with well-defined datasets, but many of them have high computational cost and suffer in the cases of both non-frontal face images and changing illuminations. In our work, we would like to build a portable system that can help the visually impaired in real-time face recognition. Therefore we select Kinect as the input sensor and propose a highly effective ICP registration method. In this section, we will discuss two basic modules of the approach: face feature extraction and ICP-based face matching, before we describe our ICP-based face recognition method.

3.1 3D facial feature point extraction

The method that Kinect used for locating facial points is a regularized maximum likelihood Deformable Model Fitting (DMF) algorithm [17]. By employing this algorithm, we can obtain the 3D facial feature points of a face, as shown in Figure 1. There are 121 face feature points in total, which are distributed symmetrically on the face based on the model proposed in [17]. After we obtain the 121 feature points, we use the symmetry relations of these points to build a full view face. This is important when only part of the face is visible in the case of side views. The symmetry relationship is found by calculating the face feature position relationship with a frontal face. In Table 1, the numbers are the indices for the 121 face feature points. We list every pair of symmetric corresponding feature points as *pl* and *pr* in the heading row of the table. Some pairs of the point indices are the same because they are on the middle axis of the face.

The 121 facial feature points can be obtained in real time, even with a large rotation angle; the DMF model can track a face and obtain all the feature points, as shown in Figure 2. Having the information of the symmetry relationship, we are able to find the corresponding face patches to build a full-view face in both 3D and 2D images. This will be very useful if we want to integrate both the 3D and 2D data for face recognition, since even though the 3D face feature points are complete by using the face model, the RGB-D data of a side view is not complete, e.g. the image in the middle in Figure 2.

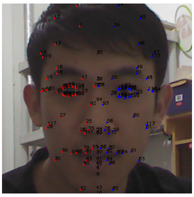

Fig. 1. 3D feature points

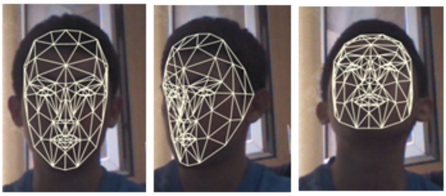

Fig. 2. Face tracker in multiple angles

3.2 3D feature point registration

Different from the method proposed by others to find 3D data descriptor through very complicated computation, we directly register the 3D facial feature points

Table 1. The symmetry relationship of the facial feature points

pl	pr	pl	pr	pl	pr	pl	pr	pl	pr	pl	pr
0	0	1	34	2	2	3	3	4	4	5	5
6	6	7	7	8	8	9	9	10	10	11	44
12	45	13	46	14	47	15	48	16	49	17	50
18	51	19	52	20	53	21	54	22	55	23	56
24	57	25	58	26	59	27	60	28	61	29	62
30	63	31	64	32	65	33	66	34	1	35	35
36	36	37	37	38	38	39	39	40	40	41	41
42	42	43	43	44	11	45	12	46	13	47	14
48	15	49	16	50	17	51	18	52	19	53	20
54	21	55	22	56	23	57	24	58	25	59	26
60	27	61	28	62	29	63	30	64	31	65	32
66	33	67	69	68	70	69	67	70	68	71	73
72	74	73	71	74	72	75	76	76	75	77	78
78	77	79	80	80	79	81	82	82	81	83	84
84	83	85	86	86	85	87	87	88	89	89	88
90	91	91	90	92	93	93	92	94	94	95	96
96	95	97	98	98	97	99	100	100	99	101	102
102	101	103	104	104	103	105	106	106	105	107	108
108	107	109	110	110	109	111	112	112	111	113	117
114	118	115	119	116	120	117	113	118	114	119	115

between two faces. The inputs are two sets of 3D facial feature points of two persons. If the two sets of data are from the same person, the points can be one-to-one registered with a very small registration error, since the 3D feature points are unique facial features on the face. If the points are from two different people, the error should be much larger due to either mismatches of the two sets or the differences in the 3D structures of the two faces even with correct feature matching. For most people, their face structures are stable but are different from each other, so if the 3D measurements are accurate enough, the difference between self-matching error (of the same person) and cross-matching error (of different persons) will be obvious. We then can use the error measurements as a similarity measures of two faces to recognize different people. A commonly used method in registering 3D points is the Iterative Closet Point (ICP) approach [3]. As we have known the identities of the 121 face feature points, their correspondences can be established between two face images. Let us use P and Q to represent the two sets of feature points:

$$\begin{gathered} P=\{p_1,p_2,p_3...p_n\} \\ Q=\{q_1,q_2,q_3...q_n\} \end{gathered} \tag{1}$$

where $p_i(i=1,2...n), q_i(i=1,2...n)$ are the 3D feature points in the two datasets, respectively. Assume the rotation matrix is R and translation vector is T, between the two datasets, then the relation between the two groups of 3D points can be represented as:

$$Q = RP + T \tag{2}$$

Denote the centers of the two sets as p and q respectively, which are:

$$p = \frac{1}{N}\sum_{i=1}^{N} p_i, q = \frac{1}{N}\sum_{i=1}^{N} q_i \tag{3}$$

the two sets of data can be represented in coordinate systems with origins as their centers:

$$p'_i = p_i - p, q'_i = q_i - q \tag{4}$$

Then we can obtain the covariance matrix H as

$$H_{(m,n)} = \sum_{i=1}^{N} p'_{(i,m)} q'_{(i,n)} \tag{5}$$

Using SVD to decompose the covariance matrix into

$$[U, D, V] = SVD(H) \tag{6}$$

the R and T can be calculated as

$$\begin{aligned} R &= VU^T \\ T &= q - Rp \end{aligned} \tag{7}$$

In the ICP algorithm, the biggest problem is finding the right initial matching point pairs. Since in a general 3D registration problem, we have no idea of the point matching relationship, we can only assume the closest points between the two datasets as the right pairs. If the initial relative pose between the two datasets is large, the iterative processing will likely converge to a local minimal and make the registration fail. However, in our face registration case, we do not have this problem since we know exactly the indexes of the 3D feature points, as shown in Figure 1. Therefore we can simply assign the point pairs and do a one-step registration, which is both robust and time efficient. By applying the one-step ICP computing, we can get the rotation matrix and the translation vector between the two sets of the points in real-time.

4 3D Registration-Based Face Recognition

4.1 ICP-based approach: registration for recognition

In Section 2, we have described the procedures in both 3D face feature point extraction and 3D face registration. The DMF algorithm is able to find the face feature points even the face has a large rotation angle to the camera. The ICP has been typically used to register multiple-view 3D data of the same object, but

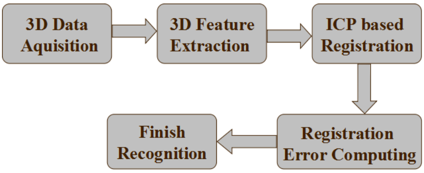

Fig. 3. Flow chart of the ICP-based face recognition

we employ it as a recognition method because the matching error is discriminative between self 3D feature matching and cross matching. The ICP-based face recognition approach has the following five steps (Figure 3):

Step 1. *3D Data Acquisition.* The 3D data is captured using and RGB-D sensor.

Step 2. *3D Feature Extraction.* Real-time face tracking is realized using the Kinect SDK and 3D facial feature points are extracted. The 3D pose (position and orientation) of the sensor to the observed face is also estimated.

Step 3. *ICP-based Registration.* The current 3D face (the probe) is compared with all the 3D faces in a database (the gallery) by using the one-step Iterative Closest Points (ICP) approach. By doing ICP, the relative pose between each pair of faces are also computed.

Step 4. *Registration Error Computing.* The registration errors of the prober face and the gallery faces are calculated as the average alignment error of the two datasets, indicating the uncertainty in face matches: the smaller the error, the more similar the two matched faces.

Step 5. *Face Recognition.* By picking up the gallery face that has the smallest registration error with the probe face, the probe face is recognized. The face recognition result can also be confirmed by the calculated relative pose between the two faces using ICP in Step 3, and the relative pose estimated from the pose estimation results in Step 2.

4.2 Robustness analysis of the ICP-based approach

If the 3D face feature points are ideal data, i.e., they are accurate, the approach would be straightforward and accurate. But in reality, the 3D data have errors. In the next two subsections, we will analyze the tolerance of our approach to 3D measurement errors, and then to look into the accuracy of the Kinect sensor we use in our experiments. This will provide insights in the requirements of 3D sensors in applying in the proposed approach, and the capacity of the current Kinect sensor.

We have captured 3D facial feature points of eight people with front faces and compute every two peoples cross matching errors as well as the self matching error. We add noise to the original data at different accuracy levels. We also add rotations to the data to simulate the pose rotations; together with added errors, this simulates 3D face captures with various poses. The result would be shown in the following tables to see the difference between self matching and cross matching error.

We add random errors with uniform distributions to original data at four different accuracy levels: 1mm, 2mm, 5mm and 10mm random errors, with no rotation. Then we add the same levels of errors with rotation angles of 20, 30, and 40 degree along the x, y and z axes, respectively. In online recognition, the 3D measures could have different levels of errors depending on the sensor accuracy as well as the distances and view angles of faces. If the 3D data are captured with the same condition, for example, the same view angle, distance and resolution, high accuracy are possible to achieve whereas if the capture conditions changes, the error may increase. Different levels of measuring errors are simulated and various viewing cases are covered. We would like to note that in many practical applications the noise is not randomly distributed but follows some geometric deviation of the model polarizing the overall result in a specific direction. However we use this simplified error model as the first step to understand the tolerance of the proposed approach with noisy input.

The 3D matching errors given different levels of errors of 3D measures and view angles are summarized in Table 2 and Figure 4 .

Table 2. The average self matching error and cross matching error when applying different error and rotation

	Self ave	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8
Err 1mm	0.0251	0.2576	0.2290	0.2117	0.3531	0.2412	0.2555	0.3488	0.2551
Err 2mm	0.0508	0.2623	0.2313	0.2072	0.3563	0.2369	0.2528	0.3462	0.2539
Err 5mm	0.1235	0.2688	0.2387	0.2128	0.3608	0.2402	0.2575	0.3466	0.2518
Err 10mm	0.2571	0.3458	0.3329	0.3059	0.4200	0.3094	0.3171	0.3884	0.3406
Err.1mm+R	0.0246	0.2578	0.2288	0.2115	0.3530	0.2409	0.2553	0.3490	0.2550
Err 2mm+R	0.0495	0.2624	0.2314	0.2112	0.3569	0.2391	0.2550	0.3425	0.2599
Err 5mm+R	0.1236	0.2861	0.2520	0.2267	0.3658	0.2505	0.2619	0.3481	0.2627
Err 10mm+R	0.2395	0.3628	0.3289	0.3039	0.4111	0.3296	0.3272	0.3900	0.3433

Table 2 shows a statistical summary with different levels of adding errors with and without rotation angles, of the comparison of self matching error and cross matching errors. The $Selfave$ column shows the average of the 8 persons' self-matching error in each simulated case. Columns from Sub 1 to Sub 8 are the averages of cross-matching errors of each person to one of the eight persons. We also show the comparison in plots (Figure 4): In each of the eight cases, the first (blue) bar is the average self-matching error, and second (red) bar is the average of the smallest cross-matching error of 8 candidates. Note the recognition is achieved by choosing the smallest matching error from all the matches. The relative height shows the average matching errors. With accurate 3D measuring data (which is the case of the 1mm 3D measurement error), the discrimination using the matching error in recognition is obvious. However, when the 3D measurement error increases to 10 mm (i.e. 1 cm), even though we still see some the discrimination but the ratio between the self-matching error and cross-matching errors are not so obvious. But we observe that with an error level of 5 mm in 3D measurements, the ratio of the self-matching and cross-matching errors is still obvious, so this implies that our proposed approach can be effective with measuring data error of about 5mm. In Table 3 we list the matching results of both self matching and cross-matching of individual people with 5mm error.

Table 3. The self and cross matching error when applying error of 5mm

No.	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8
Sub1	0.1191	0.1668	0.2638	0.4057	0.2083	0.2359	0.3978	0.2828
Sub2	0	0.1219	0.1789	0.3475	0.2306	0.2471	0.3261	0.2288
Sub3	0	0	0.1233	0.3276	0.1914	0.2272	0.2520	0.1936
Sub4	0	0	0	0.1331	0.2548	0.3222	0.5040	0.4598
Sub5	0	0	0	0	0.1244	0.2038	0.3713	0.2655
Sub6	0	0	0	0	0	0.1245	0.3507	0.2639
Sub7	0	0	0	0	0	0	0.1186	0.2488
Sub8	0	0	0	0	0	0	0	0.1254

Table 2 and Figure 4 also shows the matching results with or without rotation angles; with added rotation angles, the results have not changed much, so the approach is invariant to 3D pose rotation. This is due to the one-step ICP registration method that uses correct one-to-one face feature matching.

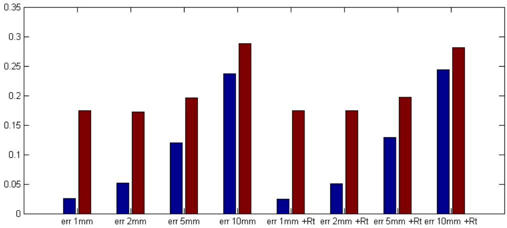

Fig. 4. The self and cross matching error under various error levels without rotations (the first four pairs) and with rotations (the last four pairs). In each pair the blue is for the self-matching error, and the red is for cross-matching error

4.3 Kinect calibration and error analysis

The RGB-D sensor we used to carry out the experiments is Kinect for Windows. It can generate depth images at a speed of 30fps with a resolution of 640x480.

Since the 3D data accuracy affects the face recognition performance, we have evaluated the measurement accuracy of the device we used in our experiments. A large 66-inch LCD TV screen is used as a flat board to accomplish the evaluation. The Kinect is placed in front of the screen at 7 different positions, ranging from 60cm to 120cm, with an interval of 10cm, which is assumed the predefined working range for our system for face recognition. At each position, the 3D data of the LCD screen is obtained and is fitted with a plane. Then the distance of each measured point to the fitted plane is calculated and the standard deviation of them is used as a measure to evaluate the accuracy of the Kinect. The result is shown in Table 4.

Table 4. The accuracy evaluation of the Kinect

Nominal Distance (cm)	Average Distance (cm)	Standard Deviation of the Points to Plane Distance (cm)
120	119.41	0.89
110	109.25	0.83
100	99.10	0.68
90	88.74	0.59
80	77.56	0.42
70	67.61	0.40
60	57.99	0.38

From Table 4 we can see that the accuracy of the Kinect is better than 10 mm (i.e. 1 cm), which is from 4 mm to 9 mm. This is between the just discriminative level and the well discriminative level as we have observed in Section 4.2. This means that the Kinect is still be okay being a 3D data capturing device for the 3D face recognition for a closer range (smaller than 1 meter), but a RGB-D sensor with higher accuracy is much desirable since in real applications, due to human motion or illumination, the 3D measure errors usually are larger than the evaluation errors in an ideal case.

When using Kinect, it outputs the depth images of *Depth* values rather than point cloud data of (x, y, z) coordinates. So we need to convert the depth data into 3D data first by using the following formula obtained by a calibration procedure using targets of known distances:

$$\begin{cases} z = a_3 D \\ x = a_1(i - x_0)z \\ y = a_2(j - y_0)z \end{cases} \tag{8}$$

where D is the depth value at the position (i, j) in the output depth image. $(x_0, y_0) = (320, 240)$ is the center of the projection, $a_3 = 4.095$ and $a_1 = a_2 = 0.00175$ are the three scale parameters estimated by the calibration.

5 Online Testing and Error Analysis

From the experiments above, we know the ICP-based registration approach can be discriminative for face recognition with a relative high error tolerance, but the Kinect data error is large in real applications, in the order of 1-2cm (10 - 20 mm) at the distance of 100 cm (1 meter), a comfortable distance between two persons for a face recognition. However, we still implement the recognition system to see how well it works. To analyze the recognition error quantitatively, we design two experiments to see how distances and face angles affect the recognition results.

First we capture 3D data of 10 people standing in front of the Kinect sensor at a distance of 1 meter . Then in the online testing, we ask one person (the tester) to step into the Kinects field of view, and the tester will be captured with different distances to the Kinect and with various face angles. We save all the registration error data and plot them to show how the distances and face angles affect the recognition results.

In the first experiment, the tester stands in front of the sensor to be captured, then the tester moves back and forward smoothly, so the tester are captured at different distances. The results of matching errors are plotted in Figure 5. In this figure, the horizontal axis is the distance whereas the vertical axis is the error. The cross-matching errors of every person in the dataset are represented by a connected line segment grouping their matching error together with a specific color, and the black line is for the self-matching errors of the tester.

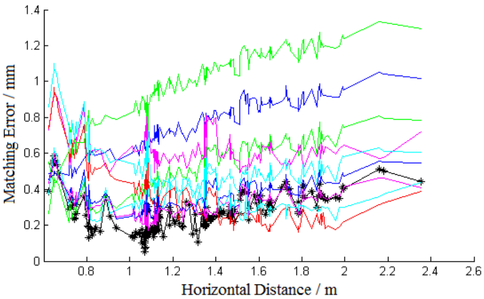

Fig. 5. Matching error plots against face distances

We can see from Figure 5, the trend of the curve is similar, and at the distance of about 1 meter, the smallest matching error provides the correct face recognition result. In most cases, within the distance range of 0.7 m to 1.4 m, the matching result for recognition is most likely to be correct, whereas in other distance ranges, the approach fail to recognize the right person.

After obtaining the relationship of distance and registration error, we also have designed the second experiment to test the angle influence on recognition results. This time we keep the tester stand at the same spot, 1m to the sensor, and ask the tester to change his facing orientation in both horizontal and vertical directions. The results are plotted in Figure 6.

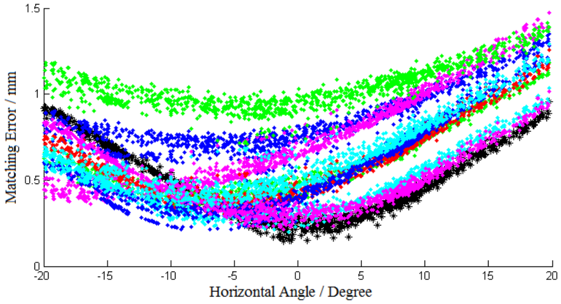

Fig. 6. Error distributions against face orientations

In the angle error figure, the horizontal axis is the horizontal face orientation angle to front and the vertical axis is the matching error. Like Experiment 1, we use the same color to represent the cross-matching between the tester and every saved person and the black curve represent the testers self-matching. We can see that with the angle increasing the matching error is growing. In theory, the 3D data should be invariant to the pose changes, but in reality, the results show that the data captured at large angles suffer from 3D measuring errors and only the measurements in near frontal angles can indicates the right matching results.

The two experiments show that due to the 3D measurement errors of the Kinect sensor, the approach can only provide robust results when the person being viewed is within a certain distance range and with a viewing angle close to the ones saved in the face database. It cannot provide the right recognition result if the distance or angle is far from the ones when we save the dataset. However we still can use the sensor for a practical face recognition of a small pool of candidates, thanks to the real time performance of the sensor, such that we can wait the best view to be captured. Furthermore, if the error can controlled as indicated in our analysis in Section 3, the approach can tolerate large distance ranges and view angles. So the approach is promising with a better version sensor like Kinect 2 for us to implement a straightforward, efficient and robust 3D facial recognition system for the visually impaired.

6 Discussions: More Uses of 3D Face Data

In this section, we will discuss a number of other cases where 3D face data can be useful in real-time face modeling, which are all important in assisting visually impaired people in socially interacting with other people. In these cases the RGB images can also be utilized, and integrated with the pure 3D-based face recognition method proposed in the paper to improve the face recognition performance. These are our ongoing work for assisting the visually impaired people in recognizing and interacting with other people in their daily life.

6.1 2D view-based face recognition

Even though the 3D face data captured by the current RGB-D sensor we used is not sufficiently accurate for 3D face recognition, the real-time estimations of 3D face poses/distances can be used to rectify and normalize the accompanying RGB images to obtain frontal faces. Or a database of faces of varying known view angles, as in Figure 7, can be bult. Then we can use the large body of 2D face recognition methods such as the eigenface approach [6], the deep face method [13], etc., to perform face recognition. The 2D face recogntition results can also be integrated with the 3D face recogntion results.

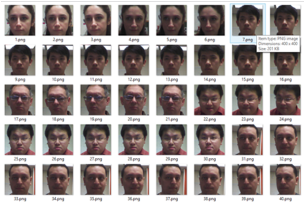

Fig. 7. Faces with varying angles

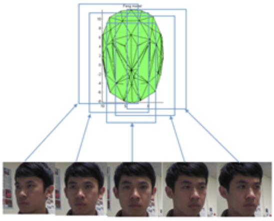

Fig. 8. Integrating multiple views

6.2 Multi-view face integration

Thanks to the real-time performance of the RGB-D sensor, we could capture a sequence of face images in real-time, and therefore build a fuller model of the 2D/3D face. We have noted the 3D face model is always complete with the 121 facial feature points, but some of the RGB views may not show the full face, such as the two side views in Figure 8. By using the real-time 3D face pose estimation and the ICP-based 3D face registration methods, an integrated view of multiple original face images can be generated. This representation can include additional features such as ears and back of the head, which are not included in the original 3D facial feature model.

6.3 Facial expression modeling

We can also use the real-time feature of the 3D facial features to model facial expressions, since varying views can be aligned, and all the 121 facial features can be identified and matched. Then we can create temporal trajectories of key facial features such as eyes, mouths, etc., for facial expression recognition. We could also generate various facial expressions using the 3D facial feature data and facial expression model. Figure 9 shows an interface of our current integrated demo system for 3D face tracking, recognition, multi-view integration and new view synthesis. In this figure, different face views and expressions are generated from the integrated model of multiple RGB-D face images. Even though the results are still preliminary, we believe they have very promising potentials for assisting the social interaction between visually impaired people with normal sighted people.

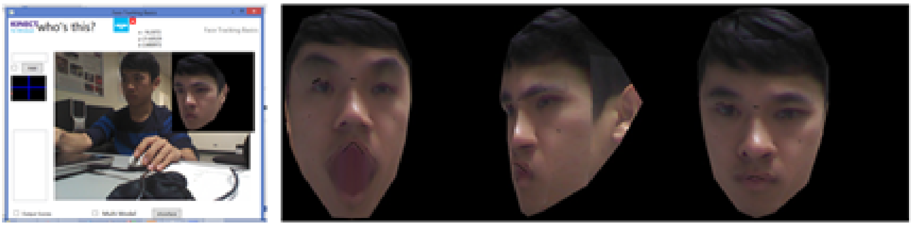

Fig. 9. System interface and expression synthesis

7 Conclusions

In this paper, we proposed a pure 3D feature data registration based approach to recognize people for visually impaired people. We analyze our error tolerance of our method against the current used Kinect sensors accuracy, and implement the approach with the Kinect RGB-D sensor. Although due to the Kinect measuring error the recognition is limited to distance and angle, we are confident that with sensor accuracy improvement like Kinect 2, the real-time 3D approach will have much better performance. The ongoing work includes the integration of both depth and RGB data for face recognition. Since we have obtained full-view face images of any view angles using the symmetric relations, we believe the integration of both the 3D geometric and RGB appearance features of the 121 face feature points and the integrated face images would improve the performance of our real-time face recognition system for assisting visually impaired people.

Acknowledgements

The authors would like to thank the US National Science Foundation (NSF) Emerging Frontiers in Research and Innovation Program for the support under award No. EFRI-1137172. The authors are also grateful to the anonymous reviewers of this paper for their valuable comments and suggestions.

References

1. Asthana, A., Marks, T.K., Jones, M.J., Tieu, K.H., Rohith, M.: Fully automatic pose-invariant face recognition via 3d pose normalization. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 937–944. IEEE (2011)
2. Berretti, S., Del Bimbo, A., Pala, P.: 3d face recognition using isogeodesic stripes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32(12), 2162–2177 (2010)
3. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Robotics-DL tentative. pp. 586–606. International Society for Optics and Photonics (1992)
4. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. International Journal of Computer Vision 107(2), 177–190 (2014)
5. Huang, D., Zhang, G., Ardabilian, M., Wang, Y., Chen, L.: 3d face recognition using distinctiveness enhanced facial representations and local feature hybrid matching. In: Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on. pp. 1–7. IEEE (2010)
6. Jafri, R., Arabnia, H.R.: A survey of face recognition techniques. JIPS 5(2), 41–68 (2009)
7. Li, B.Y., Mian, A.S., Liu, W., Krishna, A.: Using kinect for face recognition under varying poses, expressions, illumination and disguise. In: Applications of Computer Vision (WACV), 2013 IEEE Workshop on. pp. 186–192. IEEE (2013)
8. Liu, C.: Gabor-based kernel pca with fractional power polynomial models for face recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26(5), 572–581 (2004)
9. Pang, Y., Yuan, Y., Li, X.: Gabor-based region covariance matrices for face recognition. IEEE Transactions on Circuits and Systems for Video Technology 18(7), 989–993 (2008)
10. Passalis, G., Perakis, P., Theoharis, T., Kakadiaris, I.A.: Using facial symmetry to handle pose variations in real-world 3d face recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33(10), 1938–1951 (2011)
11. Phillips, P.J., Flynn, P.J., Scruggs, T., Bowyer, K.W., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on. vol. 1, pp. 947–954. IEEE (2005)
12. Queirolo, C.C., Silva, L., Bellon, O.R., Pamplona Segundo, M.: 3d face recognition using simulated annealing and the surface interpenetration measure. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32(2), 206–219 (2010)
13. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human level performance in face verification. In: IEEE CVPR (2014)
14. Torr, P.H., Zisserman, A.: Mlesac: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78(1), 138–156 (2000)
15. Wang, Y., Liu, J., Tang, X.: Robust 3d face recognition by local shape difference boosting. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32(10), 1858–1870 (2010)
16. Yang, H., Wang, Y.: A lbp-based face recognition method with hamming distance constraint. In: Image and Graphics, 2007. ICIG 2007. Fourth International Conference on. pp. 645–649. IEEE (2007)
17. Zhang, Z.: Microsoft kinect sensor and its effect. MultiMedia, IEEE 19(2), 4–10 (2012)

18. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2879–2886. IEEE (2012)