Automatic Reconstruction of Dense 3D Face Point Cloud with A Single Depth Image

Shu Zhang^{1,2}, Hui Yu^{*2}, Junyu Dong¹, Ting Wang^{1,2}, Honghai Liu²

¹Ocean University of China, Qingdao, China

²University of Portsmouth, Portsmouth, UK

hi.shu.z@outlook.com

hui.yu@port.ac.uk

Abstract—Human face analysis is the basis for many other computer vision tasks, such as camera surveillance, entrance authorization and age estimation. With 3D face models, the vision task based on facial analysis can usually achieve a higher accuracy than the 2D cases since it provides more information with the additional dimension. However, most existing 3D face reconstruction methods suffer from complicated processing and high computation. This paper presents a novel method that simplifies the 3D face reconstruction process with only one shot of Kinect data. The output of the system is a high density of 3D face point cloud with smoother surface. This provides rich details of the human face for other computer vision tasks. Experiments with real world data show promising results using the proposed method.

Keywords—3D face region; reconstruction; Kinect; k-means; RBF; interpolation

I. INTRODUCTION AND RELATED WORK

A. Introduction

Human face analysis is an important research area that provides fundamental functionality for many other computer vision tasks, such as camera surveillance, authorized entrance decision, age estimation, and gender estimation. Since 3D reconstruction is a hot topic in computer vision, 3D face analysis becomes popular in the research community. It has shown the advantages in various computer vision tasks since the additional dimension brings more information. The detail of the 3D face model is crucial to aforementioned researches. However, in practical, 3D information obtained from RGB-D data and some reconstruction methods usually has low resolution due to limitation of the hardware and the structure of the scene themselves such as depth shadowing, influence of the materials with reflection or refraction or infrared absorption in the scene [1]. Moreover, detailed 3D face reconstruction always comes with complicated or special setup during the 3D acquiring process. Some of them even involve expensive equipment.

This paper presents a novel method that can reconstruct 3D human faces with more than a hundred thousand 3D points. Furthermore, it only requires one simple shot using Kinect with fast processing. The process is fully automatic without any

manual intervention. Our method is achieved by two main steps: (a) initial 3D facial point cloud acquisition and (b) dense facial point cloud production. Experiments in section III illustrate the promising results of our method with real world data.

B. Related Work

There is a range of methods that can reconstruct 3D human faces. However most of them either require complicated and special setup, or come up with low accuracy results [2-5]. Jo et al. [6] presented a 3D face reconstruction method. They combined the simplified 3DMM and the SFM to achieve the 3D recovery. However, their method required manual annotated Facial Feature Points. Lee and Choi [7] proposed a 3D face estimation method from 2D frontal face image by the approach of the rank constraint relaxing. However, the input images should be cropped into certain size filling with face region before applying their method and the resolution of the result is fixed.

II. PROPOSED ALGORITHM

In this paper, an automatic method for generating dense 3D face point cloud is proposed combining the Kinect and Radial Basis Function (RBF). With only one shot of the Kinect, our method can obtain a dense 3D facial point cloud without any manual intervention. The work flow of the proposed method is shown in Fig. 1. The experiments with real world data show encouraging results using our method.

Figure 1. Work flow of the proposed method.

A. Initial 3D Facial Point Cloud Acquisition

The first stage of the proposed method is to acquire initial 3D points of the human face. We use the Kinect [8] as the input device to obtain the raw 3D data. Kinect was designed for human machine interaction for game play in the first place. However, the characteristics of the data captured by Kinect, especially 3D depth information, have also attracted the researchers in computer vision community in recent years. Kinect is actually an RGB-D sensor that can simultaneously provide both RGB colour and depth images. The RGB colour image is captured by a built-in normal digital camera inside the Kinect. The depth data is achieved by an infrared laser projector and an infrared video camera mounted within the Kinect, as shown in Fig. 2.

Figure 2. The Kinect sensor.

Through the RGB image and depth image are captured synchronically, it produces a spatial shift between these two images since the normal camera and the infrared camera are spatial location various [9]. Thus, we apply a registration process to the RGB image and depth image before we can utilize them. This alignment process can be obtained by taking into account of the constant distance between the RGB sensor and the Infrared sensor in the Kinect device. With the knowledge of field of view (FOV) of the Kinect, we can modify every pixel in the depth image accordingly to make them align with the pixel in RGB image. With alignment, for every coordinate of 2D point in RGB image, we retrieve the corresponding 2D coordinate in depth image. Then coordinates of 3D point in the space can be obtained with the (1).

$$\frac{x_p}{u - u_0} = \frac{y_p}{v - v_0} = \frac{z_p}{f} \quad (1)$$

After calculating 3D points, face detection is carried out on the aligned RGB image in order to filter the 3D points to restrict them within the facial region as shown in Fig. 3. We implement the facial detection process using Haar Cascade Classifier [10]. It firstly trains the Haar feature classifiers for facial features such as eyes, mouth, and nose with two sets of images. One set contains pure human face without other objects. The other set contains one or more instance of objects. With the trained classifiers, we can easily recognize the facial region within an RGB image. According to the corresponding relationship between RGB image and 3D points, we filter the 3D points to limit them inside the facial region.

Though facial detection can filter most of the 3D points not belonging to facial region, it can hardly provide the precise region of a human face. It only offers a general area of the face.

To achieve more accurate facial region, we analyze the rough facial 3D points such as in fig. 3 (b), and find that the point cloud is divided into many clusters. Only one cluster is the true facial region. Therefore, naturally, we apply a K-means clustering algorithm [11] to divide the point cloud into parts, and only reserve the cluster that contain the center points of the facial region.

(a)

(b)

Figure 3. 3D point cloud filtered by face recognition. (a) The green rectangle is the facial region recognized in RGB image; (b) is the 3D point cloud filtered within facial region from corresponding RGB image.

K-means algorithm achieves the clustering process by calculating the distance between points and centroids of the point groups. The initial centroids are randomly selected. After iterations, a clustering solution can be found that the similarity within a cluster is maximum and similarity inter-cluster is minimum. Since the facial detection has already provided a rough but tight facial region, we implement K-Means algorithm with 3 initial clusters, and reserve only one from them. The result is shown in Fig. 4 which is a more precise facial region than that in Fig. 3 (b). In Fig. 3 (b), the 3D point cloud includes the points that belong to the background, which makes the red frame obviously much larger. Meanwhile, in Fig.4, the 3D point cloud only includes the points that belong to human face.

(a)

(b)

Figure 4. More accurate facial region after applying k-means algorithm.

B. Dense Facial Point Cloud Producing

As shown in Fig. 4 there are the reconstructed 3D facial point cloud is low quality with missing data and holes on the surface. Moreover, the point cloud is not dense enough to

provide the details of the human face. Therefore, we proceed to second stage to apply an interpolation process to fill up the depth blank, and make the point cloud denser, which results in a smoother surface.

We treat the 3D point cloud as a set of spatial variables (X, Y, Z), where (X, Y) follows 2D coordinate, and (Z) is a set of scalar variables that seated in aforementioned 2D coordinate. This concept is consistent with the nature of the depth data that come from the Kinect. Accordingly, the interpolation process is to find the (Z) value in (X, Y) coordinate system. Therefore, we apply Radial Basis Function (RBF) [12] to perform interpolation with Gaussian Kernel as the Basis Function. RBF model is built up by a set of Basis Functions with radial symmetry. It follows the form as in (2).

$$f(x) = \sum (w_i \cdot \varphi(x - c_i)) \quad (2)$$

where φ is a basis function, w_i are the weights for each basis function. And c_i are the interpolation centers, which coincide with the original 3D points in the facial point cloud. Basis functions can follow several forms, such as Gaussian basis function, Multi-Quadric Basis Function, Thin Plate Spline Basis Function, Cubic Basis Function and poly-harmonic Basis Functions. According to the values of the neighbour original points and the distances to the neighbour original points, the basis function provides a value for a certain place in the multi-dimensional coordinate. We choose Gaussian Basis Function as φ expressed in (3) since Gaussian Basis Function is localized and compact. A distance about $6 \cdot R_0$ will make the value almost equal zero with the Gaussian Basis Function. This will prevent interpolation from over-smooth in order to reserve details on the facial surface.

$$\varphi(r) = e^{\frac{-r^2}{2 \cdot \sigma^2}} = e^{\frac{-r^2}{R_0^2}} \quad (3)$$

As a result, the choice of R_0 is crucial. If R_0 is too small, the interpolated points will prone to ground zero value since there're not enough original points within valid influential distance. The value will degrade rapidly when apart from the position of the original point. If R_0 is too large, the calculation process becomes more time-consuming while interpolation performance doesn't improve too much. Studies show that a better result emerges when R_0 is slightly larger than the average distance between original points in the coordinate. To achieve full automatization of the process, we employ Kd-Tree [13] to calculate the average distance for points in original point cloud.

Kd-Tree is a binary tree with k dimensional data. It's similar to Binary Search Tree, to divide the data in k dimension into several partitions. As illustrated in Fig 5, for example, there are six point in a 2 dimensional space. A Kd-Tree is built by dividing the points in the space according into two dimensions once a time. For instance, in Fig. 5, the blue vertical line on the left and blue circle in the first layer of the Kd-Tree on the right imply the division of the points along the first dimension. The green horizontal lines on the left and green circle in the second layer of the Kd-Tree on the right imply the

division of the points divided by previous division along the second dimension. In each layer of the Kd-Tree, it chooses the dimension with max invariance of the points, and divides the points along this dimension. Following the nodes in the Kd-Tree as shown in Fig. 5, any point in the space can be located in the leaf node very fast. With the structure of the Kd-Tree, we can quickly search for k points that have k nearest distance with one certain point in the space. This helps form the neighbourhood of the 3D point cloud with their distance to each other.

Figure 5. Illustration of the building of the Kd-Tree. Left part is the distribution of the points in 2 dimension; Right part is the Kd-Tree related to the 2D points.

With obtained neighbourhood of the points in point cloud, the average distance between points can be calculated easily. We use 6 times of average distance as the R_0 to perform RBF interpolation. The RGB colours of the interpolated points are simply the average colour of the neighbour points. As shown in Fig. 6, a denser point cloud of 3D face can be obtained with full details of the surface. As we observed, most holes are filled up, and the surface is much smoother. More experiments with other subjects are presented in the next section.

Figure 6. Dense 3D facial point cloud at final stage of the method.

III. EXPERIMENTS AND RESULTS

The method has been implemented using C++ language, which is compiled by Microsoft Visual Studio. Microsoft Kinect for Xbox 360 was used to acquire the depth information. The initial depth image for the whole scene is under the resolution of 640x480. There are six subjects in our experiments. Kinect takes one shot of each subject's face, and our method successfully reconstructs the dense 3D point cloud of their faces.

Figure 7. Illustration of the interpolation result in nose area.

As Fig. 7 shows, in the facial point cloud obtained in the first stage of the method, there are noticeable holes on the surface of the area of nose. After the final stage of the method, the major holes in that area have been filled up to make it a denser point cloud. The surface is continuous without the loss of details. The dense 3D facial point clouds reconstructed for six subjects are illustrated in Fig. 8. As we can see from results, our method can extract human 3D face region exactly from the scene, fill up major holes on the surface of the point cloud with stable performance. The density of the reconstructed point cloud is high enough to convey detailed information of a human face. The performance of the reconstruction of dense 3D point is shown in Table I, which contains the 3D point counts in two stages of our method for comparison. According to Table I, Our method fills up almost 1000% of the original points, which make the surface of the face smoother.

TABLE I. INTERPOLATION PERFORMANCE

Subjects No.	3D Points Count in Facial Point Cloud	
	Two Stages	
	<i>Initial Facial Point Cloud</i>	<i>Dense Facial Point Cloud</i>
(1)	13067	106765
(2)	13332	135993
(3)	9591	95050
(4)	9760	81542
(5)	9851	99881
(6)	8111	74243

Subject No. is corresponding to the one in Fig. 8 (a).

Such dense facial point cloud could provide sufficient information for face analysis applications that require every detail of the facial surface, such as expression recognition,

facial paralysis evaluation, and facial verification. More facial features can be extracted from the dense facial surface obtained by proposed method. This is essential to aforementioned applications.

(c)

(d)

(e)

(f)

(g)

Figure 8. Dense 3D facial point cloud producing results. (a) are six subjects' RGB image with facial recognition; (b-g) are six group of experiments with one subject in each group. In each group, (1-4) are the initial 3D facial point cloud obtained after the first stage of proposed method, while (5-8) are the dense 3D facial point cloud obtained after final stage of proposed method. More dense 3D facial point cloud demonstrations from larger view angles can be found from (9) to (12) in (b) and (c).

IV. CONCLUSION

In this paper, a novel method generating dense 3D face point cloud is presented. The novelty of the proposed method is that we simplify the process of dense 3D facial reconstruction, which is easily to operate with only one simple shot using Kinect aimed at a human face without any further manual intervention. The reconstructed point cloud has a smooth surface with detailed information of human face along with facial texture. The total count of the points in the cloud is around a hundred thousand. The 3D face produced by proposed method can be further utilized by many other computer vision tasks related to human face analysis, such as facial recognition, facial paralysis evaluation.

However, the main drawback of the proposed method is that lack of real-time processing due to the interpolation for mass of points. Further optimization can be done to improve the computation cost through both software and hardware, such as multi-thread approaches and GPU computation. In future, we will try to improve the quality of the reconstructed 3D face data combining some image-based methods for applications such as rehabilitation, facial palsy and human machine interaction.

ACKNOWLEDGMENT

This work was supported by EU seventh framework programme under grant agreement No. 611391, Development of Robot-Enhanced Therapy for Children with Autism Spectrum Disorders (DREAM); This work was also supported by National Natural Science Foundation Of China (NSFC) (No. 61271405, 61401413); and the Ph.D. Program Foundation of Ministry of Education of China (No. 20120132110018).

REFERENCES

- [1] J. Zhu, L. Wang, R. Yang, and J. Davis, "Fusion of time-of-flight depth and stereo for high accuracy depth maps," In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2008, pp. 1-8
- [2] E. Garcia, J. L. Dugelay, and H. Delingette, "Low cost 3D face acquisition and modeling," in *Proc. IEEE Comput. Soc. Conf. Information Technology: Coding and Computing*, April 2001, pp. 0657-0657.
- [3] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High resolution passive facial performance capture," *ACM Trans. Graphics (TOG)*, vol. 29, no. 4, pp. 41, 2010.
- [4] M. S. Hossain, M. Akbar, and J. D. Starkey, "Inexpensive construction of a 3D face model from stereo images," In *Proc. 10th IEEE Int. Conf. Computer and Information Technology (ICCIT)*, December 2007, pp. 1-6.
- [5] H. Yu, O. G. B. Garrod, and P. G. Schyns, "Perception-Driven Facial Expression Synthesis," *Computer & Graphics*, vol. 36, issue 3, pp. 152-162, May 2012.
- [6] J. Jo, H. Choi, I. J. Kim, and J. Kim, "Single-view-based 3D facial reconstruction method robust against pose variations," *Pattern Recognition*, vol. 48, no. 1, pp. 73-85, 2015.
- [7] M. Lee, and C. H. Choi, "Real-time facial shape recovery from a single image under general, unknown lighting by rank relaxation," *Computer Vision and Image Understanding*, vol. 120, pp. 59-69, 2014.
- [8] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultiMedia*, vol. 19, no. 2, pp. 4-10, 2012.
- [9] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Trans. Cybernetics*, vol. 43, no. 5, pp. 1318-1334, 2014.
- [10] P. I. Wilson, and J. Fernandez, "Facial feature detection using Haar classifiers," *Journal of Computing Sciences in Colleges*, vol. 21, no. 4, pp. 127-133, 2006.
- [11] P. V. Rao, and S. K. M. Rao, "Performance Issues on K-Mean Partitioning Clustering Algorithm," *Int. Journal of Computer (IJC)*, vol. 14, no. 1, pp. 41-51, 2014.
- [12] G. E. Fasshauer, and M. J. McCourt, "Stable evaluation of Gaussian radial basis function interpolants," *SIAM Journal on Scientific Computing*, vol. 34, no. 2, pp. A737-A762, 2012.
- [13] R. A. Brown, "Building k-d Tree in $O(k \log n)$ Time," *Journal of Computer Graphics Techniques*, vol. 4, no. 1, 2015.