FROM KINECT VIDEO TO REALISTIC AND ANIMATABLE MPEG-4 FACE MODEL: A COMPLETE FRAMEWORK

Laura Turban[1], Denis Girard[2], Neslihan Kose[3], Jean-Luc Dugelay[4]
[1,2,3,4] EURECOM, Sophia Antipolis, France
[1,2,3,4] turban, girard, kose, dugelay@eurecom.fr

ABSTRACT

The recent success of the Kinect sensor has a significant impact on 3D data based computer applications. This study aims to obtain MPEG-4 compliant realistic and animatable face models from Kinect video. The complete framework for this process includes initially the computation of high quality 3D scans from RGB-D Kinect video, and then the computation of animatable MPEG-4 face models using these high quality scans. This study shows that it is possible to obtain high quality 3D scans and realistic and animatable face models of subjects using lower quality Kinect data.

Index Terms- Kinect, MPEG-4, animatable face model

1. INTRODUCTION

Since several years, the emerging RGB-D cameras such as the Kinect sensor [1] are very popular. They have been successfully used for many 3D based applications.

The Kinect sensor has received significant attention from several research communities [2], such as computer vision [3], computer graphics [4], augmented reality [5], human computer- interaction [6], instrument measurement [7], and robotics [8]. The vast utilization of the Kinect sensor in several domains is due to its efficiency, low-cost, ease of RGB-D mapping, and multi-modal sensing. In biometrics domain, Kinect sensor has been used in the analysis of body parts segmentation and tracking [9], gait recognition [10], and body anthropometric analysis [11]. The utilization of the Kinect sensor for face recognition is limited due to the lack of a standard database. In [12], several face databases are listed. In this list, the EURECOM Kinect Face Dataset (EURECOM KFD) [13] is one of the two databases which provides video data collected by the Kinect sensor. Hence, in the present study, we used this database for our analysis.

Recent surveys [14, 15] show that in face recognition, 3D cues provide complementary information in addition to 2D. 3D shape information is robust to illumination and pose variations [16]. However, the utilization of high quality 3D scans leads to an unbalanced comparison between 2D and 3D data in terms of acquisition efficiency. 3D face scanning needs careful user cooperation hence it is inefficient and too long for users to keep steady during data collection.

The Kinect sensor overcomes the above problem by providing both 2D and 3D data simultaneously at interactive rates. As a result, 3D or 2D+3D data is provided for real-time and online processing [17]. However, in comparison to the high quality laser scans, the quality of 3D data captured by the Kinect is relatively low. The problems encountered are missing data in blind points [2], relatively low depth resolution, noise at large depth transitions at boundaries, and spatial calibration of RGB and depth images [18].

The main contributions of this paper are as follows:

- In the EURECOM KFD (test database), 3D scans (.obj files) are already provided. In this study, in addition to these 3D scans, higher quality 3D scans are computed using the video data available in the database. These higher quality 3D scans show almost complete 3D face.
- Then, realistic and animatable face models that are in compliance with MPEG-4 specifications are computed for each subject in the database by applying Thin Plate Spline (TPS) warping [23] to the high quality 3D scans.
- The outputs are both the high quality 3D scans and the corresponding animatable MPEG-4 face models of the subjects involved in our test database.
- The proposed process for obtaining animatable models from Kinect video is explained in details which makes this work reproducible. Following the acquisition protocol explained in this paper using a Kinect sensor, it is possible to obtain realistic and animatable models of any subject with the proposed approach.

The rest of this paper is as follows. In Section 2, the specifications of the EURECOM KFD are given. In Section 3, the complete framework for the computation of high quality 3D model from Kinect video and the corresponding animatable MPEG-4 face models is described. Section 4 explains the additional patch proposed for the EURECOM KFD. Finally, conclusions are given in Section 5.

2. KINECT FACE DATABASE

In this section, the summary of the specifications of the EURECOM KFD [13] is given.

In the database, 52 subjects appeared. 38 of them are males, 14 of them are females. The participants are from different ethnicity, and were born between 1974 and 1987.

Their ethnicities are categorized into the following classes (with the number of participants in each class): Caucasian (21), Middle East/ Maghreb (11), East Asian (10), Indian (4), African American (3), and Hispanic (3).

The database was captured in two sessions happened at different time period (about half month), where the same recording protocol was applied. In each session, 2D RGB image, 2.5D depth map, 3D point cloud and RGB-D video sequence were captured for each identity. 9 facial variations were involved in both sessions, which are neutral face, smiling, mouth open, strong illumination, occlusion by sunglasses, occlusion by hand, occlusion by paper, right face and left face profile.

The database was captured in a controlled environment (natural light at daytime). The participants were asked to follow the predefined acquisition protocol. The protocol to record the RGB-D video sequences for each person consisted of slow head movements in both the vertical (pitch) and the horizontal (yaw) directions. The video sequence allows extraction of multiple frames with different poses. A white board was placed behind each participant with fixed distance to the Kinect, to produce a simple background which can be easily filtered.

A software application (based on the OpenNI [20] library) was used for the database recording. The captured RGB image and the depth image were cropped using a pre-defined ROI (with the size of 256×256).

The dataset comes with the manual landmarks of 6 positions in the face: left eye, right eye, the tip of nose, left side of mouth, right side of mouth and the chin to perform facial region extraction and normalization.

3. THE COMPLETE FRAMEWORK

The EURECOM KFD provides RGB-D video data. Thus, it can be used to develop animatable MPEG-4 models from low quality Kinect video data. Animatable MPEG-4 models are animatable models which are computed by warping an animatable generic model using 3D frontal face models of the subjects and some of the MPEG-4 specified feature points on that models. Commercial players use animatable face models to perform animation encoded in a MPEG-4 data stream. Hence realistic animations of synthetic faces is possible using these MPEG-4 animatable models.

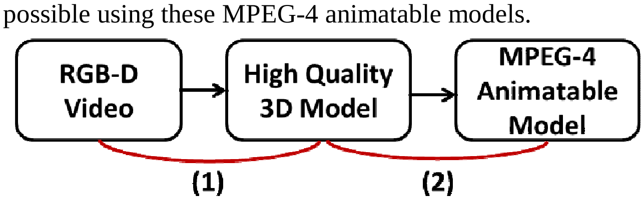


Figure 1. The complete framework for the proposed study

Figure 1 shows the flowchart of the complete framework of this study. In the first step (1), we were inspired from the study [19], in which high quality 3D models are computed from RGB-D video via 3D accumulation and refining. The differences in our technique compared to [19] are defined in section 3.1.4. In the second step (2), we compute animatable MPEG-4 models using the pre-computed high quality 3D models. In order to obtain MPEG-4 animatable models, we benefited from the study [22].

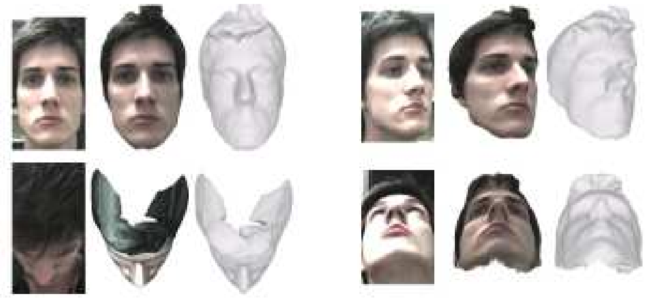


Figure 2. An example of a 3D face model generated using a video of cooperative head movement (images taken from [19]).

Section 3.1 and 3.2 explain the first and the second steps of the proposed approach (Figure 1), respectively.

3.1. Kinect Video to High Quality 3D Model (Step 1)

Recently, dense 3D modelling using the Kinect has attracted vast amounts of attention. Pioneering works [4, 5] have demonstrated how to build a dense 3D map of indoor scene/ object by camera tracking using sparse features and optimize the 3D points aggregation, taking advantage of the real-time, low-cost, and ease of RGB-D mapping from the Kinect sensor. More recently, 3D face modelling using the Kinect is introduced in [19, 21] to generate a 3D model for video conferences and Massive Multi-player Online Games.

An example of the 3D model generated from video data is illustrated in Figure 2. The video based face modelling [19] shown in Figure 2 aggregates and averages data points from multiple single depth frames in a cylindrical coordinates system, so as to capture the complementary information brought by the given video sequence. Bilateral smoothing is applied to remove noise while keeping edges. The generated 3D faces from the Kinect video sequence have demonstrated comparable accuracy to laser scanned 3D faces [19].

The EURECOM KFD provides for each subject video records including two sets of RGB and depth frames. In the present study, we initially developed a technique aiming to rebuild high quality 3D faces from these low resolution RGB-D videos. The proposed method is explained in the following subsections.

The whole process to create 3D faces from Kinect videos in this paper is described in Figure 3. First, frames from the 3D videos are extracted and converted to a readable format, bitmap (1). Then, they are cropped and their background is deleted so that the resulting image focuses on the face (2). In order to compute head pose, features are extracted from the frame by comparing it to training models (3). From the position of the eyes and the mouth corners, we can identify the head pose (4). Given the head pose and the depth frames extracted from the videos, a depth frame is computed for the whole face and a 3D file representing the face is written (5). Finally, the texture map is computed from the frames, and

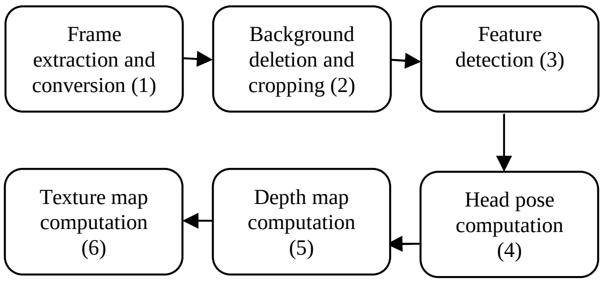


Figure 3. 3D face computation process

applied to the 3D face (6).

3.1.1. Frame extraction and conversion ((1) in Fig. 3)

Extracting frames from Kinect records has to be achieved using tools provided with the OpenNI open source library [22]. These tools allow obtaining RAW format images from ONI record for both RGB and depth frames. Frames are then converted into basic BMP format (Kinect frames are initially 640x480, with 3 interlaced RGB channels for color and 2 interlaced Grey and Alpha channels for depth).

3.1.2. Background deletion and cropping ((2) in Fig. 3)

We cannot isolate the image from the rest of the background by simply removing all pixels in the depth frames which are further than a given threshold, since there are indeed unwanted noise and outliers due to errors in the depth determination and depending on the background material.

The purpose is then to assume that the background has a uniform color; the average background color is determined and removed at each frame. To achieve this, we first have to determine all the non-moving pixels in a video sequence. The aggregated difference of each frame with the initial one is computed assuming that all pixels which have an aggregated difference value close to zero correspond to the background. Thus, the median value of all concerned pixels gives an approximation of the average background color. Finally, for each depth frames, we delete all pixels which have a corresponding color value close to this median value. However, there are still some outliers in depth frames which have to be excluded from the process. These outliers are mainly very bright pixels due to noise from the sensor or because of features being too close to the depth camera. A simple exclusion using a grey value threshold is enough to avoid the outliers.

3.1.3. Feature detection ((3) in Fig. 3)

Head pose estimation is needed in order to recompose the 3D face. Given the fact that depth values have low precision, we have chosen to estimate head pose by relying on RGB data. Eyes and mouth corners are used to compute rotation angle for head pose estimation (cf. Section 3.1.4).

The feature detection algorithm processes face detection and landmarks positioning by computing similarity scores between the considered frame and a set of training models. These landmark positions enable to compute the approximate positions of eyes and mouth corners (Figure 4).

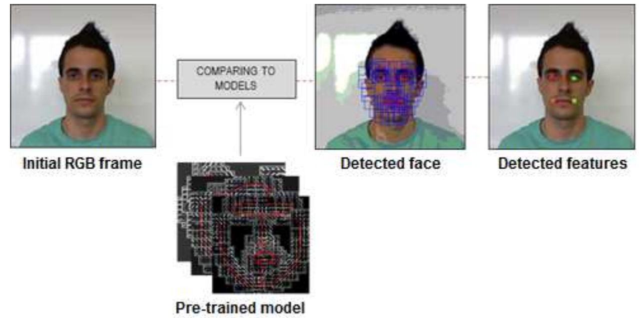


Figure 4. Landmarks detection

3.1.4. Head pose computation ((4) in Fig. 3)

The head pose estimation is mainly based on the distance between paired features such as eyes or mouth corners. Such distances between landmarks have to be computed from real world coordinates; thus initially a conversion is needed. For a given pixel $p(x, y)$ on the depth frame, the associated real-world coordinates are:

$$X = \frac{x - c_x}{f_x} \times (Z - f_{ir}), \qquad Y = \frac{y - c_y}{f_y} \times (Z - f_{ir}) \quad (1)$$

where $Z = p(x, y)$, c_x and c_y are the optical center position of the camera, f_x, f_y are the focal lengths of the RGB camera (along X and Y axis) and f_{ir} is the focal length of the infrared depth camera. Assuming that the initial frame is a front face image, (the origin of the coordinates located in the middle of the face for the initial pose), using the real world coordinates (Eq. 1), trigonometric analysis gives the following relations:

$$\theta = \tan^{-1}\left(\frac{b}{a}\right) \quad (2)$$

$$\mathrm{a} = \left(\frac{1}{x_{right}} + \frac{1}{x_{left}}\right)\sin\gamma, \quad \mathrm{b} = \left(\frac{1}{x_{right}} - \frac{1}{x_{left}}\right)\cos\gamma \quad (3)$$

where x_{right} and x_{left} are the position of right and left eye after rotation on the x-plan, θ is the angle between the Z axis and head rotation axis for the initial frame, γ is the angle between the axis passing from the initial location of right eye and the axis passing from the location of right eye after rotation (i.e. rotation angle) (Figure 5). These equations enable to compute head pose of the subject.

Note that in [19], for reconstructing 3D face from Kinect records, ICP algorithm [25] is used to determine head pose. In our study, head pose estimation is achieved through basic feature detection and simple geometric operations. Hence the applied technique is simple and enable to reach real-time with classic CPU programming.

3.1.5. Depth map computation ((5) in Fig. 3)

In this step, the 3D face is modeled through a cylindrical depth map. We consider a virtual cylinder surrounding the face and project each head's pixel on it. The depth map contains all the data needed to reconstruct the 3D model and directly gives the cylindrical coordinates.

Thus for each depth frame, an associated "unitary" depth map has to be computed. The set of all unitary depth maps

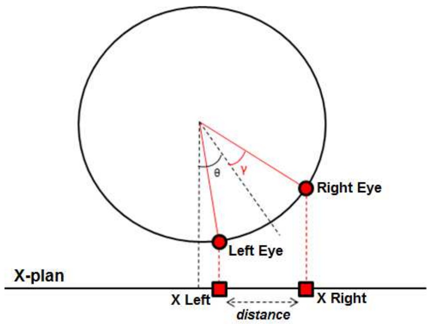


Figure 5. Head pose estimation

will then be used to generate the final depth map corresponding to the 3D reconstructed face.

The purpose of this step is to determine precisely how to locate a depth frame's pixel on its unitary depth map and what its corresponding value is. As introduced before, it is mainly a matter of conversion between the real-world Cartesian coordinates of each pixel into cylindrical ones, and projection of the depth value on the considered cylinder. Here, we assume that the original axis of the cylinder is vertical and placed 10 cm behind the closest point of the initial frame (this approximation gives good results). The pixel's value is computed given its distance from this axis, and its position on the depth map is obtained from its height on the depth frame and its angle with the front axis (and an additional offset equal to the current face pose).

Once the complete set of unitary depth map is obtained (Figure 6), an array of all non-null pixels' values is filled for each pixel's location. The final value corresponds to the median value of these arrays (Figure 7).

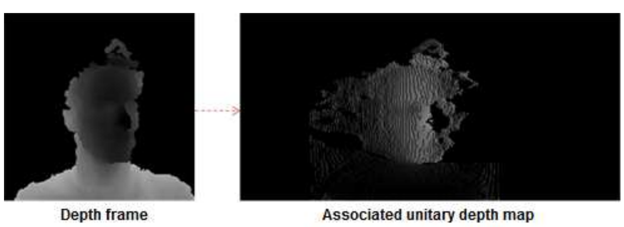


Figure 6. Unitary depth map

Pre-processing is also important in order to fill holes which are due to the lack of data and to smooth the result. In this study, a simple linear interpolation on the horizontal axis is used to complete the depth map. Then a bilateral filter is applied to smooth the result, which preserves edges while reducing noise in other parts of the image.

3.1.6. Texture map computation ((6) in Fig. 3)

The previous method can be easily applied to compute a texture color map. The algorithm applied is almost exactly the same; we just complete the unitary maps with color instead of depth, and process the sequence three times (one time per color channel) (Figure 8).

From the final depth map and texture map, a 3D object is generated in .OBJ/.MTL format. These models complete the existing EURECOM KFD (which already contains raw 3D

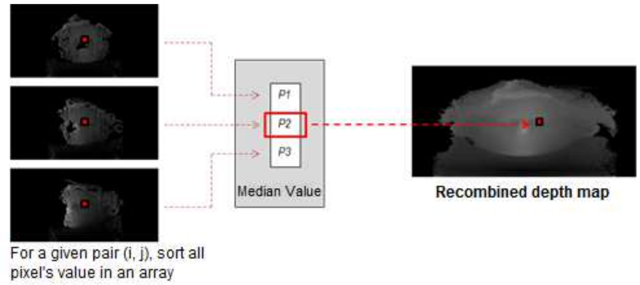


Figure 7. Final depth map computation

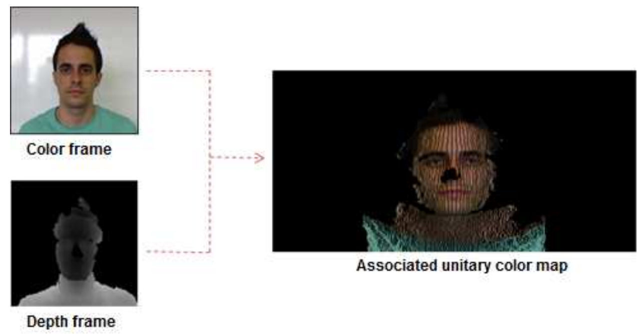


Figure 8. Unitary texture map generation

conversions of the previously included depth frames) with complete 3D heads (Figure 9). This figure shows the 3D head reconstruction performance of the proposed approach for the identity shown in Figure 4.

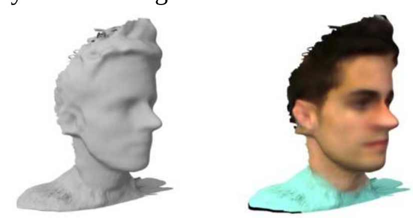


Figure 9. 3D head reconstructed

Note that our reconstruction program is still highly dependent on the recording conditions. For many sessions, the subject was too close to the camera and a high number of corrupted depth frames did not enable a proper reconstruction. Using the EURECOM KFD, we observed that 59% of all records (i.e. 61 faces over 104) were successfully computed in 3D.

3.2 From High Quality 3D Model to MPEG-4 Animatable Model (Step 2)

In order to obtain an animatable model from a 3D face, the method used here is based on warping a generic animatable face, which is compatible with MPEG-4, in order to make it look similar to the target 3D faces. The generic face that was used for warping is shown in Figure 10. Its feature points were also available.

The method used for warping was first introduced in [22]. The generic face is warped using TPS method [23]. However, before applying the TPS warping, the generic face needs to be rescaled and then aligned to the target face (i.e. the reconstructed 3D face). The warping is done in two

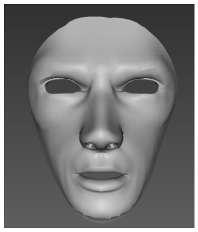


Figure 10. Generic Face

steps. First a coarse warping is computed, by warping the face according to a non-linear transformation defined by the pairing of the feature points in the initial 3D face with the corresponding feature point in the generic model and interpolating other points with the TPS method. A second warping is done by pairing all the points of the original face with the closest one in the generic face. The original texture of the 3D face from the database is applied to the warped face afterwards. Finally, a list of the feature points of the warped face is retrieved, by using the already existing feature points of the generic face and finding the corresponding coordinates in the new face.

We have to note though that in our process, 3 feature points were annotated as opposed to 29 manually annotated feature points for all the 3D faces in [22] for alignment and coarse warping. Although the utilization of less feature points decreases the alignment and coarse warping performance slightly, it still provides significant performance with the advantage of less computation complexity. The faces that were successfully computed in 3D (i.e. 3D faces computed in the previous section) were also successfully warped, except for 5 of them, probably due to the low number of feature points annotated in the initial 3D face. After warping, the associated feature points for the center of the eyes were computed.

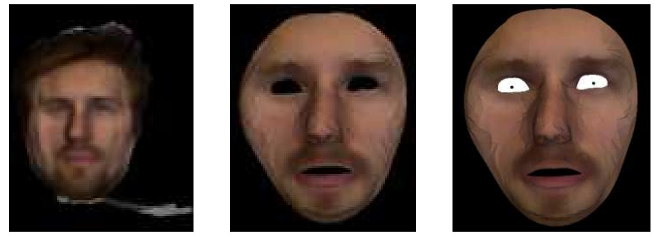
The different steps of the computation of the animatable faces are shown in Figure 11. 56 VRML files describing 3D faces after the warping of the generic animatable face with the associated FDP (Face Definition Parameters) files describing their feature points were obtained.

3.2.1 MPEG-4 Animatable Model

MPEG-4 Face and Body Animation is part of the MPEG- 4 standard developed by the Moving Picture Experts Group, describing a standard to represent humans with key feature points called Face Definition Parameters and how to move them with the definition of Face Animation Parameters.

84 feature points and 68 Face Animation Parameters are defined in the standard. Face Animation Parameters include both high level descriptions, such as expressions (joy, sadness, anger) and visemes (position of the feature points when the model says a specific phoneme) and low level parameters (specific movements of one part of the face, such as raising an eyebrow).

The software used for the animation of the faces is visage|life, which was developed by Visage Technologies [24] to animate faces (and bodies) in a way that was



(a) (b) (c)

Figure 11. Generating the animatable model: (a) 3D face, (b) warped generic face with texture, (c) warped generic face with texture and eyes

compatible with the MPEG-4 standard. The idea behind the animation in visage|life is to import a VRML description of a 3D face, with its feature points (in a .fdp file with the same name) and clone the animations from an existing animatable model. Feature points positions can be verified and modified in the software as well.

The cloning process consists first in the computation of the facial motion for the existing animatable face. The facial motion is then applied to the new input 3D face. All the faces that were successfully warped were also successfully animated.

In the visage|SDK, which is a Software Development Kit made available by Visage Technologies, a face tracking demo is available. It finds and tracks a face in real time from a camera feed and outputs Face Animation Parameters and face position to animate the 3D animatable face in the same way as the tracked face (Figure 12). Figure 9 shows the performance of the proposed approach for high quality 3D head reconstruction. The example videos under the link *http://rgb-d.eurecom.fr/* show the performance of the proposed approach for MPEG-4 animatable model computation. In this link, one video shows the speech synthesis process achieved using the output animatable model of this study. The other video shows the face tracking process with expression variations using a computed MPEG-4 animatable model in this paper (Note that in this video, the animatable face is not the animatable model of the tracked face). These videos clearly show that it is possible to compute realistic and animatable MPEG-4 face models using the RGB-D video data collected with a cheap 3D scanner such as Kinect.

4. ADDITIONAL PATCH TO THE EURECOM KFD

In this study, a new patch was created for the EURECOM KFD. It contains for each subject, the 3D face computed during the first step (Section 3.1) and the VRML file for the creation of the animatable face, the final animatable face and its feature points when available.

The architecture of the patch is as follows:

0001: subject number (4 digits, 0001 to 0052)

s1: session number (s1 or s2)

3D Face: contains the 3D face after computation, eventually cleaned using the Autodesk Maya software (in .OBJ format), its associated .mtl file and texture file.

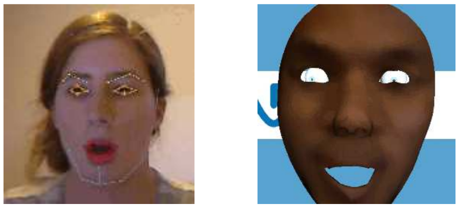


Figure 12. Example of a tracked expression: (a) tracked face, (b) animated face from the database

Animatable Face: contains the 3D face used for animation (in VRML format), the animatable face (.afm), its associated feature points file and the associated texture file.

According to the information in [13], the EURECOM KFD was prepared for the studies regarding biometrics domain. However, with the proposed patch, the utilization of the database can be more general. Since it includes the 3D animatable models, it can be used in several other domains.

5. CONCLUSION

In this paper, we used the video data in EURECOM KFD [13], which is collected with the Kinect sensor. Although the performance of the proposed approach is tested only using this database, following the acquisition protocol explained in this paper, it is possible to obtain animatable models of any subject with the proposed approach using a Kinect sensor.

In this study, we aim to produce realistic and animatable MPEG-4 face models using the Kinect video data. For this purpose, first, we built high quality 3D faces from low resolution RGB-D videos. After extracting the video frames, this process is achieved via 3D accumulation and refining. In the next step, these complete 3D faces of high quality are used to obtain animatable MPEG-4 models. For this purpose, TPS warping is applied using a generic face model and the animatable models of each subject are evaluated.The faces which are successfully warped are also successfully animated using the visage|SDK, which is a Software Development Kit made available by Visage Technologies.

This study provides the high quality 3D face, the VRML file for the creation of the animatable face, the final animatable face and its feature points for each subject in the existing EURECOM KFD. This study proves that it is possible to obtain high quality 3D faces using a cheap 3D scanner such as Kinect. The techniques used for the whole process are explained in details, which makes the work reproducible. Since the output of this study is a 3D animatable model, it can be used not only in biometrics but also in other domains. Our future perspective is to compare the performance of the proposed approach with other existing techniques for the evaluation of animatable models.

REFERENCES

[1] [Online]. Available: http://www.xbox.com/en-US/KINECT

[2] Z. Zhang. Microsoft kinect sensor and its effect, MultiMedia, IEEE, vol. 19, no. 2, pp. 4 –10, feb. 2012.

[3] Special issue on computer vision for rgb-d sensors: Kinect and its applications, IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 42, no. 4, pp. 1295–1296, 2012.

[4] S. Izadi, D. Kim, O. Hilliges, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera, in UIST, 2011, pp. 559–568.

[5] M. Elendt, Ed., Int. Conf. on Computer Graphics and Interactive Techniques, SIGGRAPH 2011, Talks Proceedings. ACM, 2011.

[6] R. Johnson, K. O'Hara, A. Sellen, et al. Exploring the potential for touchless interaction in image-guided interventional radiology, Proc. of the SIGCHI Conf. on Human Factors in Computing Systems, 2011, pp. 3323–3332.

[7] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications, Sensors, vol. 12, no. 2, pp. 1437–1454, 2012.

[8] M. F. Fallon, H. Johannsson, and J. J. Leonard. Efficient scene simulation for robust monte carlo localization using an rgb-d camera. IEEE ICRA., 2012, pp. 1663–1670.

[9] J. Shotton, A. Fitzgibbon,et al. Real-time human pose recognition in parts from single depth images, IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'11. pp. 1297–1304.

[10] M. Gabel, R. Gilad-Bachrach, E. Renshaw, et al. Full body gait analysis with kinect, Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC), 2012.

[11] C. Velardo and J.-L. Dugelay, Real time extraction of body soft biometric from 3d videos, Proc. of the 19th ACM Int. Conf. on Multimedia, 2011, pp. 781–782.

[12] http://www.face-rec.org/databases/

[13] R. Min, N. Kose, J.-L. Dugelay, KinectFaceDB: A Kinect Database for Face Recognition, Systems, Man, and Cybernetics: Systems, IEEE Transactions on , vol.44, no.11, pp.1534-1548.

[14] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2d and 3d face recognition: A survey, Pattern Recogn. Lett., vol. 28, no. 14, pp. 1885–1906, 2007.

[15] K. W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition, Comput. Vis. Image Underst., vol. 101, no. 1, pp. 1–15, 2006.

[16] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes, IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 2, pp. 239–256, Feb. 1992.

[17] R. Min, J. Choi, G. Medioni, and J.-L. Dugelay. Real-time 3D face identification from a depth camera, Int. Conf. on Pattern Recognition, ICPR, Japan, 2012.

[18] D. Herrera C., J. Kannala, and J. Heikkila. Joint depth and color camera calibration with distortion correction, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 34, no. 10, pp. 2058–2064, 2012.

[19] M. Hernandez, J. Choi, and G. Medioni. Laser scan quality 3-d face modeling using a low-cost depth camera, European Signal Processing Conf., EUSIPCO, pp. 1995–1999, 2012.

[20] OpenNI User Guide, http://www.openni.org/documentation

[21] M. Zollhofer, M. Martinek, et al. Automatic reconstruction of personalized avatars from 3d face scans, Comput. Animat. Virtual Worlds, vol. 22, no. 2-3, pp. 195–202, 2011.

[22] N. Erdoğmuş, R. Ethève, J.-L. Dugelay, Obtaining MPEG-4 compliant animatable 3D face models by using TPS method, IEEE Signal Processing and Communications Applications Conference (SIU), pp.157-160, 22-24 April 2010

[23] F. L. Bookstein, Principal Warps: Thin-Plate Splines and the Decomposition of Deformations, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 11, no.6, 1989.

[24] http://www.visagetechnologies.com

[25] G. Medioni and R. Waupotitsch. Face modeling and recognition in 3-d, pp. 232–, 2003.