3D face reconstruction and verification using multi-view RGB-D data

1st Radostina Petkova
Faculty of Telecommunications
Technical University of Sofia
Sofia, Bulgaria
rapetkova@tu-sofia.bg

2nd Agata Manolova
Faculty of Telecommunications
Technical University of Sofia
Sofia, Bulgaria
amanolova@tu-sofia.bg

3rd Krasimir Tonchev
Faculty of Telecommunications
Technical University of Sofia
Sofia, Bulgaria
k_tonchev@tu-sofia.bg

4th Vladimir Poulkov
Faculty of Telecommunications
Technical University of Sofia
Sofia, Bulgaria
vkp@tu-sofia.bg

Abstract—**The development of computer graphics and technologies leads to the emergence of different methods for presenting the three-dimensional information in the virtual world. In this paper we present an algorithm for 3D model reconstruction of a human face. A key aspect is that the face is captured from multiple views by a stationary low cost sensor providing both color and depth data. The different views are then registered into a common model. A verification method for assessing the accuracy of the reconstructed face is also presented.**

Keywords—3D face reconstruction, verification

I. Introduction

Three-dimensional (3D) object reconstruction is a common task in the fields of computer vision and computer graphics. Determining the right method for it exerts great influence on the development of many applications in medicine, architecture, biometrics, film and video game industries, virtual and augmented reality, teleconferencing, etc. The object reconstruction is the process of restoring a scene or an object in 3D virtual space in a realistic way. Although recent 3D techniques and processing algorithms require much more computational power and time, they have their essential advantages over methods operating on two-dimensional (2D) data. These benefits manifest themselves in the fact that 3D information is substantially more detailed as a result of the capacity to preserve depth knowledge, i.e. distance on the z axis. Thus, there is an opportunity to preserve the shape of the objects, one of their important characteristics, resistant to factors by which 2D representation is often affected unfavorably (e.g. insufficient lighting, different orientation, and closure).

Two main approaches for the reconstruction process are distinguished in the literature [1]–[3]. The first approach relies solely on the acquired 3D information and is known as "model-free" [1] or "data-driven" [2], [3]. The second one uses statistical priors of the reconstructing objects and is found in the literature with the terms "model-based" [1], [3] or "template fitting" [2]. Some studies [4], [5] restore objects from depth data captured just from a single view. Such type of reconstructions, however, are often destitute of completeness, especially when objects' surface is complex (e.g. facial skin). In such cases, multi-view approaches are more applicable. The work of [6] is demonstrative in this regard. The authors make a comparison between the reconstructions achieved by following single-view and multi-view approaches, respectively. Object's restoration from multiple views, however, require the implementation of an additional registration step in the reconstruction process. Studies [6]–[11] present different alignment strategies, mostly based on Iterative Closest Points (ICP) algorithm. The accuracy of the reconstructed objects needs to be evaluated in order to assess the performance of the proposed algorithm. However, such an assessment is difficult to be made due to the need of ground truth measurements. Most of the studies discussed avoid implementing the verification step because real measurements can be obtained only by an expensive laser scanners. To the best of our knowledge, in just one of them [8] the results are compared with ground truth data which causes the other algorithms' accuracy to be quite questionable.

The job of accurately reconstructing the 3D shape of human faces is difficult but many researchers work actively on it because of the many applications, including face identification, manipulation, and animation [12], [13]. Faces allow us to communicate and express ourselves through expressions, mimics, and gestures, but contain intricacies and require wide range of degrees of freedom. Thus, 3D face reconstruction is one of the popular tasks of 3D object reconstruction and is the focus at this work. We concentrate our efforts on the data-driven approach as Kinect version 2 sensor is used to acquire the facial data. The contributions of our work are:

- designing a 3D face reconstruction algorithm following multi-view approach;
- proposing a verification method by which accuracy performance of 3D reconstruction algorithms can be evaluated without the usage of laser scanners.

The rest of this paper is organized as follows. The proposed reconstruction algorithm is presented in section II. Experimental results and verification are given in III. Results analysis and future work are discussed in section IV. Finally, in section V a conclusion is presented.

II. Description of the proposed 3D face reconstruction algorithm

This section describes the reconstruction process. Block diagram of the proposed algorithm is shown in Fig. 1.

First, a 3D data acquisition step is performed by capturing data from multiple views of the human face. Then, face detection is applied on the RGB images to extract the face bounding boxes. Noise filtering is applied on the 3D data of the detected faces, which data is extracted using the one-to-one correspondence of the RGB and 3D values. RGB values remain unchanged. Afterwards, two parallel processings are done. The first one, detects facial key points on the RGB face

2022 Global Conference on Wireless and Optical Technologies (GCWOT) | 978-1-6654-7105-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/GCWOT53057.2022.9772920

978-1-6654-7105-3/22/$31.00 ©2022 IEEE

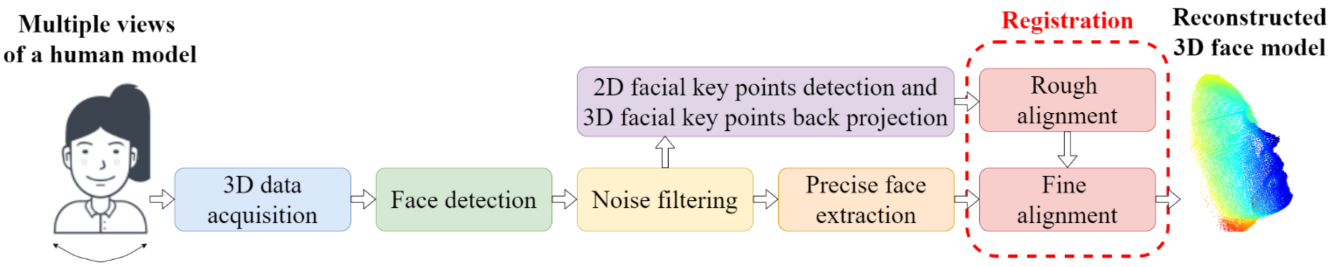

Fig. 1. Proposed 3D face reconstruction algorithm

bounding boxes. The detected points are then back projected to extract the corresponding 3D facial points. These 3D facial points are used in the rough alignment step. During the second processing, the region of the filtered face is further refined, and the result is used in the fine registration step.

A. *Data acquisition*

The data acquisition step is performed using the Kinect v2 sensor which can provide both RGB and corresponding 3D data based on the depth image. Kinect's position stays fixed but the model rotates his/her head in front of it. Acquired data is passed to a server where point clouds from the different views are recorded. The point cloud is a set of 3D points where each point is represented by its x, y, z spatial coordinates. The point clouds acquired by the Kinect sensor are kept in $N \times 6$ dimensional arrays where N is the number of points in the cloud and 6 corresponds to the respective x, y, z, and R, G, B values of the color image.

B. *Face detection and face extraction*

Point clouds acquired by the sensor contain a lot of surrounding environments in addition to the face. This information is no of interest to the reconstruction process and must be removed. The face detector operates only over the RGB part of the point cloud. In this work the Max-Margin Object Detector Convolutional Neural Network (MMOD CNN), integrated in dlib library [14], is chosen. The work of [15] is utilized but is later modified to ensure that the whole face is present in the detected area. The result is a bounding box surrounding the face in the RGB image. Thus, the unnecessary information of the cloud can be discarded, and further processing can be done.

C. *Noise filtering*

For noise filtering, the guided point cloud filter from [16] is employed. The basic idea is that each 3D point of the filtered output cloud is a linear combination of the same point in the original cloud:

$$\boldsymbol{p}_i' = a_i \boldsymbol{p}_i + \boldsymbol{b}_i \tag{1}$$

where $\boldsymbol{p}_i$ is a point from the original cloud $\boldsymbol{P}$, and $\boldsymbol{P} = \{\boldsymbol{p}_i \in R^3\}$, a_i and $\boldsymbol{b}_i$ are the linear model parameters, and $\boldsymbol{p}_i'$ is a point from the filtered cloud. For the definition of a_i and $\boldsymbol{b}_i$, and more details, refer to [17].

D. *Facial key points detection and back projection*

The step of facial key points detection is implemented via the algorithm proposed in [18] called Deep Alignment Network (DAN). The 2D RGB images of detected faces extracted from the filtered point clouds representing different views are passed through DAN. As a result, pixel's coordinates of 68 facial key points per image are found. By using the one-to-one correspondence between the RGB values and the 3D points, the 2D coordinates of the detected facial key points serve to find their respective 3D spatial coordinates. In our work, this is done by back projection. After facial key points' spatial coordinates are found, the coordinates of each 8 surrounding 3D points are also considered in order to achieve better accuracy. This is done via interpolation following the equation:

$$\boldsymbol{p}_i' = \frac{1}{2}\boldsymbol{p}_i + \frac{1}{16}\sum_{j \in N(\boldsymbol{p}_i)} \boldsymbol{p}_j \tag{2}$$

where $\boldsymbol{p}_i = \left[\mathrm{x}_{\boldsymbol{p}_i}, \mathrm{y}_{\boldsymbol{p}_i}, \mathrm{z}_{\boldsymbol{p}_i}\right]$ is a 3D facial key point, result of the back projection, $\boldsymbol{p}_i' = \left[x_{\boldsymbol{p}_i}', y_{\boldsymbol{p}_i}', z_{\boldsymbol{p}_i}'\right]$ represent the same facial key point but as a result of the interpolation, and $j \in N(\boldsymbol{p}_i)$ represents the set of 8 neighboring points, $\boldsymbol{p}_j$.

E. *Precise face extraction*

Once the spatial coordinates of the facial key points have been successfully extracted, the filtered point clouds of detected faces representing different views can be refined. The idea is only the oval part of the face to be obtained – without hair and other unnecessary points, caught in the images of detected faces. For this purpose, an ellipsoid filter is created considering the following inequality:

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} + \frac{(z - z_0)^2}{c^2} < 1 \tag{3}$$

where x_0, y_0, z_0 refer to the center of the ellipsoid coordinate system; a, b, c are the distances to the ellipsoid's sides, respectively on x, y, and z axes; x, y, z are the coordinates of the points from the cloud to be checked whether represent the oval facial area. The ellipsoid's center is positioned at the tip of the nose, found as the closest to the camera point, considering just the facial cluster of points. The ellipsoid filter should keep all points of the cloud falling inside it. So, each point of the cloud with coordinates x, y, z that satisfy (3) is preserved. All others are, thus, discarded.

F. *Rough alignment*

The rough alignment process aims to find the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ between the 3D facial key points of the different views and their corresponding points in the reference view by minimizing the following function:

$$E(\boldsymbol{R},\boldsymbol{t}) = \sum_{i=1}^{M} \|\boldsymbol{q}_i - \boldsymbol{R}\boldsymbol{p}_i - \boldsymbol{t}\|^2 \qquad (4)$$

where $\boldsymbol{p}_i$ is the corresponding 3D facial key point of $\boldsymbol{q}_i$ from the reference face, and M is the total number of facial key points. By performing Global Procrustes Analysis (GPA) [19], (4) can be written as:

$$E(\boldsymbol{R},\boldsymbol{t}) = \sum_{i=1}^{M} \|\boldsymbol{q}_i' - \boldsymbol{R}\boldsymbol{p}_i'\|^2 \qquad (5)$$

where $\boldsymbol{p}_i'$ and $\boldsymbol{q}_i'$ represent $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ after subtracting from them the centroids of the respective sets of facial key points. The solution of (5) is then found by computing the covariance matrix $\boldsymbol{W}$ (6) and decomposing it via Singular Value Decomposition, SVD (7):

$$\boldsymbol{W} = \sum_{i=1}^{M} \boldsymbol{q}_i'\boldsymbol{p}_i'^{T} \qquad (6)$$

$$\boldsymbol{W} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T \qquad (7)$$

where $\boldsymbol{D}$ is the diagonal matrix containing singular values and is not needed for further processing. $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices, on the other side, are used for computing the rotation matrix:

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T \qquad (8)$$

Knowing $\boldsymbol{R}$, $\boldsymbol{t}$ is therefore found by:

$$\boldsymbol{t} = \mu_{\boldsymbol{Q}} - \boldsymbol{R}\mu_{\boldsymbol{P}} \qquad (9)$$

where $\mu_{\boldsymbol{P}}$ and $\mu_{\boldsymbol{Q}}$ are the centroids of the respective facial key points sets. The calculated $\boldsymbol{R}$ and $\boldsymbol{t}$ are applied on the precisely extracted facial point clouds. As a result, roughly aligned point clouds from different views are obtained.

G. Fine alignment

During the rough alignment step $\boldsymbol{R}$ and $\boldsymbol{t}$ are found in a single iteration since the correspondence between facial key points is known. This process is performed to prepare the data for the next fine alignment step implementing ICP algorithm [19]. All the computations from the rough alignment stage are calculated iteratively at this stage, as corresponding facial key points are no longer used but instead, initial prediction for corresponding points is made at each iteration. The Nearest neighbor classifier is used for this task. At each iteration, $\boldsymbol{R}$ and $\boldsymbol{t}$ computed are applied on the respective views. At the end, the fine aligned facial point clouds are obtained. They are merged into a common model.

III. Experimental Results and Verification

A. Our data set

To complete the 3D face reconstruction task following the multi-view approach, we collect our own data set of 10 subjects captured from different views. The acquisition step is performed by the fixed Kinect v2 sensor. We assume that the test subjects rotate their heads with uniform speed, so we take approximately 28 frames, equally spaced from start to end of the recorded sequences. The acquired point clouds are then passed to the proposed reconstruction algorithm. Fig. 2 shows some of the RGB images extracted from the point clouds of multiple views for one of the models.

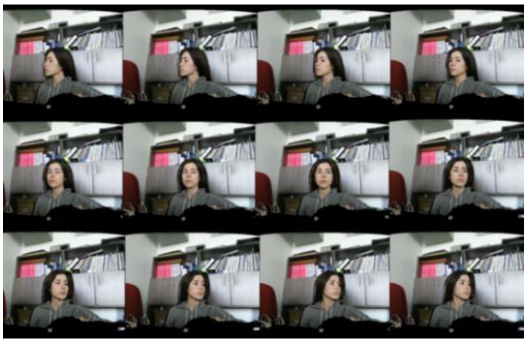

Fig. 2. RGB part of the acquired point clouds from different views for one of the models

B. Artificially generated data set

To evaluate the accuracy performance of the designed 3D face reconstruction algorithm, comparing the results with ground truth measurements is required. In this work, we avoid operating with the expensive laser scanners by making use of the 3D morphable face models (3DMFM), specifically Basel Face Model (BFM) [20]. This is the second main contribution of our work. Block diagram of the process of artificially generating data is depicted in Fig. 3.

For artificially generating 3D data, we partially modify the approach proposed in [21]. By changing BFM parameters, 30 different Basel faces are created. To ensure correctness of the proposed alternative, the generated faces must be registered in the coordinate system of an imaginary camera system that mimics the action of the real Kinect v2 sensor (similar scale and position to the fixed imaginary camera). The position of the real faces is defined by composing a histogram of distances for each of the 3 spatial axes. Then, the generated Basel faces are appropriately translated to the same positions in the simulated environment. They are of the same scale as the real ones, so no scaling problems appear. Afterwards, the Basel faces are rotated in order to be captured from different views as in the actual process of data acquisition. The render used to simulate the imaginary camera [21] is developed by us in a such way that it replicates

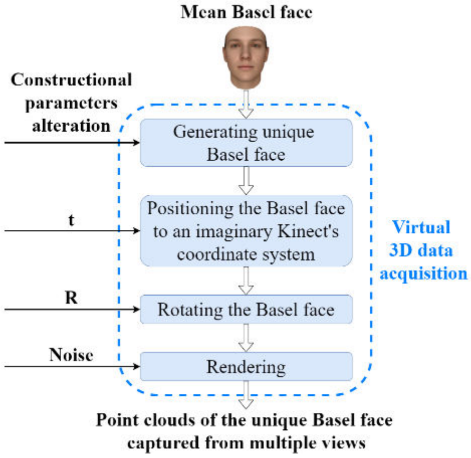
Mean Basel face
Constructional parameters alteration
Generating unique Basel face
t
Positioning the Basel face to an imaginary Kinect's coordinate system
R
Rotating the Basel face
Noise
Rendering
Virtual 3D data acquisition
Point clouds of the unique Basel face captured from multiple views

Fig. 3. Block diagram of the process of artificially generating data

as much as possible the Kinect device. The render originally captures only that RGB part of the Basel face, which is visible to it, and deletes all other points of the statistical models. However, we modify it to keep not only the colorful information of the Basel faces, and thus providing only 2D images, but also to keep the saved points' spatial coordinates. Therefore, 3D RGB point clouds from different views are obtained. For better realism, Gaussian noise is added to each of them. The artificially generated noised clouds for each of the Basel faces are then passed to the same reconstruction algorithm described above. The reconstructed Basel faces, as we call them, are finally compared with the initial Basel faces, deemed as reference, or ground truth. The difference between them is an indicator for the performance accuracy of the developed 3D face reconstruction algorithm. Fig. 4 visualizes the RGB images extracted from the virtually generated point clouds of multiple views for one of the Basel models.

C. Results

Qualitative representation. Following, we present the visual results obtained by executing the reconstruction process. Fig. 5(a), 5(b) and 5(c) show the point cloud for one of the real models and for one of the views before face detection, after face detection and after precise face extraction, respectively. Fig. 6(a) and 6(b) visualize the same point cloud before filtering the noise and after applying the guided point cloud filter. The unaligned and the aligned facial key points for all the views of the same human model are depicted in Fig. 7(a) and Fig. 7(b), respectively. Again, for the same human model, Fig. 8 shows the unaligned facial point clouds (Fig. 8(a)), the roughly aligned facial point clouds (Fig. 8(b)) and the fine aligned facial point clouds (Fig. 8(c)). The artificially generated point clouds are also passed to the 3D face reconstruction algorithm. One of the reference Basel faces and the respective reconstructed Basel face are depicted in Fig. 9(a) and Fig. 9(b).

Quantitative representation. Next, the numerical results obtained by comparing one of the reference Basel faces with the respective reconstructed Basel face are presented. This is done by taking all 3D points from the reference face individually and looking for their 5 nearest neighbors in the reconstructed face via K-Nearest Neighbor (KNN) algorithm (K is set to 5). Thus, for each 3D point from the reference face, a set of 5 distances is obtained. These 5 distances are then averaged to a mean value. A histogram over all obtained mean values for all the points is given in Fig. 10. The average distance over all the mean distances is computed to be: 1.506mm.

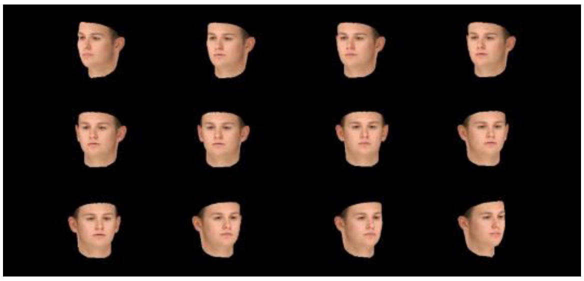

Fig. 4. RGB part of the virtually generated point clouds from different views for one of the Basel models

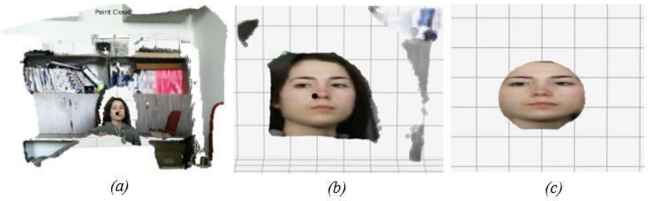

Fig. 5. Point cloud before face detection (a), after face detection (b), and after face extraction (c)

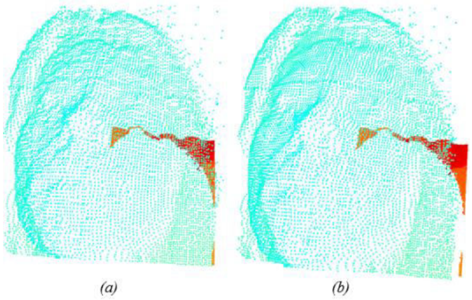

Fig. 6. Point cloud before (a) and after (b) noise filtering

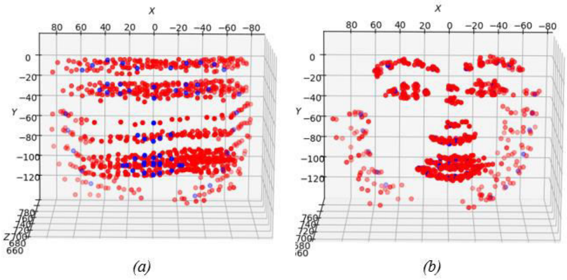

Fig. 7. Facial key points before (a) and after (b) alignment

IV. Results discussion and future work

Based on the experimental results, we conclude that the initial point clouds contain a lot of unnecessary information that has to be removed. Applying just the face detector is not enough since many non-facial points still exist in the point clouds. In order to delete them, we apply ellipsoid filtering. The results show that just the oval facial part remains. Noise filtering implemented before the precise face extraction is the other pre-processing step that needs to be conducted before the registration. It brings to smoother face surface and removed outliers which positively affects the subsequent alignment. The purpose of the registration is finding the best alignment among different views, so they can be combined into a common model. The rough alignment, using the corresponding sets of facial key points, cannot find the optimal transformations but is quite necessary to ensure good initialization for the following ICP algorithm implemented in the fine registration stage.

To verify the operation accuracy of the designed algorithm, the reconstruction process is executed again but with the artificially generated data as an input. The numerical comparison between the points from one of the reference Basel models and these from the respective reconstructed Basel model shows that for about 70% of the points the mean

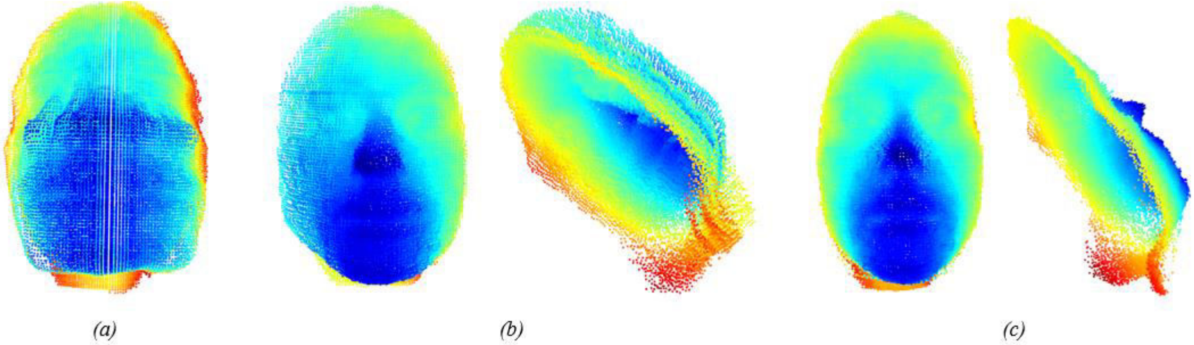

Fig. 8. Unaligned facial point clouds (a), roughly aligned facial point clouds (b) and fine aligned facial point clouds (c)

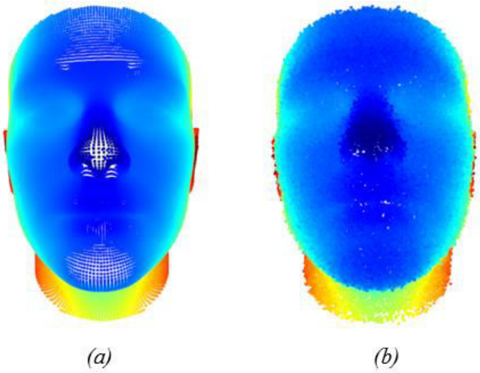

Fig. 9. Reference Basel face (a) and reconstructed Basel face (b)

distance is between 0 and 1mm and for the 90%, it is less than 2 mm. The averaged value over all the mean distances is 1.506 mm which confirms high accuracy of the proposed algorithm.

The idea of verifying the operation of the proposed reconstruction algorithm by utilizing the artificially generated data and comparing the statistical models before and after the reconstruction is a promising approach. However, "Is it applicable?", is a question whose answer varies depending on the intended application and the accuracy we want to achieve. In order to make our verifying approach appropriate for evaluating real object reconstructions and to ensure their high accuracy of restoration, here we mark some very important algorithm's improvements that must be done and are subject to our future work.

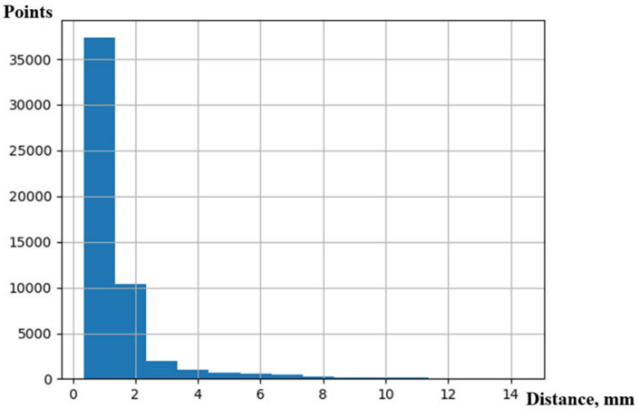

Fig. 10. Histogram of the mean distances for each point of the reference Basel model to the respective 5 nearest points in the reconstructed Basel model

First, as the noise is the factor that affects the reconstruction process the most, we need to model the noise distribution that the Kinect adds to the acquired point clouds, so we can add the same noise when generating the artificial data. Second, when adopting the statistical model, we need to simulate the actual device capturing process. This means that we have not only to register the model in the coordinate system of the imaginary camera by applying just the rotation and translation, but we also need to take into account the intrinsic parameters of the real camera used. Thus, when rendering the statistical model from different views, we must consider simulating the focal length, field of view, lens distortion of the Kinect sensor. Third, when rotating the model's head in order to capture it from different views, we have to define the appropriate angles to take samples from, so complete analogy to the real acquisition process to be achieved. For future, in the end of the reconstruction process, we also would like to transform the point cloud model into an accurate mesh structure.

V. Conclusion

In this paper we were focused on two main ideas. The first one was on developing a 3D face reconstruction algorithm from real Kinect measurements following the multi-view approach. The second idea was about designing a verification method for evaluating the proposed algorithm's accuracy which can effectively replace the expensive laser scanners. Although both ideas were successfully realized and promising results were obtained, some questions about the reconstructed model's accuracy still arise. However, we propose a clear path to solve them.

Acknowledgment

The authors acknowledged support from the project HOLOTWIN (D01-285/06.10.2020) financed by the Ministry of Education and Science of Bulgaria.

References

[1] M. Zollhofer, J. Thies, M. Colaianni, M. Stamminger, and G. Greiner, “Interactive model-based reconstruction of the human head using an rgb-d sensor,” Computer Animation and Virtual Worlds, vol. 25, no. 3-4, pp. 213–222, 2014.

[2] P. Anasosalu, D. Thomas, and A. Sugimoto, “Compact and accurate 3-d face modeling using an rgb-d camera: let’s open the door to 3-d video

conference," in Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 67–74, 2013.

[3] D. Fidaleo and G. Medioni, "Model-assisted 3d face reconstruction from video," in International Workshop on Analysis and Modeling of Faces and Gestures, pp. 124–138, Springer, 2007.

[4] R. Siv, I. Ardiyanto, and R. Hartanto, "3d human face reconstruction using depth sensor of kinect 2," in 2018 International Conference on Information and Communications Technology (ICOIACT), pp. 355–359, IEEE, 2018.

[5] S. Zhang, H. Yu, J. Dong, T. Wang, Z. Ju, and H. Liu, "Automatic reconstruction of dense 3d face point cloud with a single depth image," in 2015 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1439–1444, IEEE, 2015.

[6] M. Haque, A. Chatterjee, V. Madhav Govindu, et al., "High quality photometric reconstruction using a depth camera," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2275–2282, 2014.

[7] R. Toldo, A. Beinat, and F. Crosilla, "Global registration of multiple point clouds embedding the generalized procrustes analysis into an icp framework," in 3DPVT 2010 Conference, vol. 2, p. 5, 2010.

[8] M. Hernandez, J. Choi, and G. Medioni, "Near laser-scan quality 3-d face reconstruction from a low-quality depth stream," Image and Vision Computing, vol. 36, pp. 61–69, 2015.

[9] R. Y. Takimoto, M. d. S. G. Tsuzuki, R. Vogelaar, T. de Castro Martins, A. K. Sato, Y. Iwao, T. Gotoh, and S. Kagei, "3d reconstruction and multiple point cloud registration using a low precision rgb-d sensor," Mechatronics, vol. 35, pp. 11–22, 2016.

[10] J. ZHANG, N. LI, J. ZHANG, W. WANG, and Y. DUAN, "A study on face reconstruction using data from a low cost sensor".

[11] D. Kim, J. Choi, J. T. Leksut, and G. Medioni, "Accurate 3d face modeling and recognition from rgb-d stream in the presence of large pose changes," in 2016 IEEE International Conference on Image Processing (ICIP), pp. 3011–3015, IEEE, 2016.

[12] L. Ulrich, E. Vezzetti, S. Moos, and F. Marcolin, "Analysis of rgbd camera technologies for supporting different facial usage scenarios," Multimedia Tools and Applications, vol. 79, no. 39, pp. 29375–29398, 2020.

[13] M. Zollhofer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. P ¨ erez, ´ M. Stamminger, M. Nießner, and C. Theobalt, "State of the art on monocular 3d face reconstruction, tracking, and applications," in Computer Graphics Forum, vol. 37, pp. 523–550, Wiley Online Library, 2018.

[14] D. E. King, "Dlib-ml: A machine learning toolkit," The Journal of Machine Learning Research, vol. 10, pp. 1755–1758, 2009.

[15] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3d face reconstruction and dense alignment with position map regression network," in Proceedings of the European Conference on Computer Vision (ECCV), pp. 534–551, 2018.

[16] K. He, J. Sun, and X. Tang, "Guided image filtering," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 6, pp. 1397– 1409, 2013.

[17] X.-F. Han, J. S. Jin, M.-J. Wang, and W. Jiang, "Guided 3d point cloud filtering," Multimedia Tools and Applications, vol. 77, no. 13, pp. 17397– 17411, 2018.

[18] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep alignment network: A convolutional neural network for robust face alignment," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 88–97, 2017.

[19] C. Stachniss, "Point cloud alignment using icp", https://www.youtube.com/watch?v=djnd502836w

[20] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3d face model for pose and illumination invariant face recognition," in 2009 sixth IEEE international conference on advanced video and signal based surveillance, pp. 296–301, Ieee, 2009.

[21] YadiraF, "face3d: Python tools for processing 3d face", https://github.com/YadiraF/face3d