Real-time Upper Body Reconstruction and Streaming for Mixed Reality Applications

Dimitrios Laskos
Electrical and Computer Engineering Department
University of Patras
Patras, Greece
Email: dlaskos@ece.upatras.gr

Konstantinos Moustakas
Electrical and Computer Engineering Department
University of Patras
Patras, Greece
Email: moustakas@ece.upatras.gr

***Abstract*—In view of the challenges of real-time 3D reconstruction and transmission, the research on tele-immersion systems has been quite intense. We present an end-to-end, real-time 3D reconstruction system of the human body's upper part in mixed reality applications, implemented with the use of a single depth camera on the capture side, whereas no special setup is required. Our system captures the scene, extracts the user's point cloud and by quantizing it, achieves real-time mesh generation and streaming. This system, together with the appropriate virtual (VR) or augmented (AR) reality equipment, creates a sense of a more direct, face-to-face communication, as if both users were in the same environment.**

***Keywords*-Tele-Immersion; 3D reconstruction; Real-time; Mixed Reality**

I. Introduction

Tele-immersion refers to an emerging technology that allows remote users to co-exist in the same virtual or augmented space supporting realistic communication, while trying to reach the level of direct human contact. In recent years, and especially after consumer depth cameras appeared in the market, there has been a particularly extensive research on immersive real-time 3D systems and several works with impressive results have been published. Nevertheless, most of them, require the use of many depth cameras, specially adapted premises and, in general, costly equipment. The proposed real-time 3D reconstruction system of the human body's upper part is implemented by using a single depth camera on the capture side. On the receiving side, the visualization may take place in a virtual or augmented reality environment, depending on the available equipment.

II. Related Work

The first tele-immersion systems appeared in the 1990s; [1]–[3] however, increased computational demands of 3D reconstruction in combination with hardware limitations at that time, led to poor quality models although the results were promising.

The development of computers was followed by several works [4]–[6], the results of which were impressive at the time but not fully satisfactory. In some of them (e.g. [7], [8]), entire scenes were reconstructed in real-time, which was quite a complex problem given the massive data to be reconstructed.

Following the marketing of depth cameras (e.g. Microsoft Kinect), highly effective systems were developed [9]–[11]. Nevertheless, all of them require a great number of depth cameras in specific positions, specially designed studios as well as highly expensive equipment.

Viewport [12] is a system similar to the proposed one, as regards the goal and the upper body reconstruction. Nevertheless, on the capture side it requires a camera rig consisting of a Kinect Camera, three IR cameras, three color cameras and two IR laser projectors. Microsoft's holoportation [13] system achieves high quality real-time 3D reconstruction and transmission of whole scenes; however, its hardware requirements are particularly high since it requires multiple high-end desktops and cameras on the capture side. SLAMCast [14] is an impressive system for large-scale environments, which supports streaming to multiple clients. However, in addition to the end point systems it requires an additional PC which serves as a server to manage the reconstruction and streaming.

The sole equipment required for the proposed system is a single RGB-D camera with a conventional PC on the capture side, whereas on the rendering side AR or VR equipment is necessary. No specially designed area is required, as the user may place the camera immediately in front of him/her resulting in a user-friendly environment. In our system, user motion is not strictly limited and the data size can be adjusted according to the available bandwidth of the network.

III. Proposed system

This work presents an end-to-end, real-time, tele-immersive, 3D reconstruction system of the upper part of a user's body. Such system allows for the user's projection on one or more remote users by means of relatively conventional equipment. Our system is composed of five basic steps, as shown on the figure 1.

On the capture side, there is a StereoLabs® Zed Mini RGB-D camera, which is placed just as a plain web camera.

A. Scene capture and user extraction

From the camera, the point cloud of the entire scene is received. Each point is given in the form of a 3D coordinate

Figure 1. The proposed system pipeline

(x, y, z) and the axis of coordinates starts at the left camera's lens, as it is shown on the figure 2. The number's values are mentioned in meters. The user extraction took place by using each point's depth and rejecting points at a distance from the camera exceeding a value of z_{max}. Such value is at the user's discretion, but is usually measured within the interval $[0.5, 2]$ meters. In this manner, the system is capable of supporting the reconstruction of other objects or more users and user motion is not strictly limited.

Figure 2. Zed's Mini coordinate units used [15]

Figure 3. User extraction was accomplished by rejecting points with $z > z_{max}$

Therefore, in the present paper point cloud shall henceforth refer to the point cloud derived following the user's extraction.

B. Geometry Processing

Mesh generation from a point cloud is a classic problem in graphics and computational geometry and a huge amount of work has been published (e.g. [16], [17]). In the proposed system, the Marching Cubes [18] algorithm was used used after quantizing the point cloud. For setting the algorithm's logical restriction, three basic steps were taken.

We initially calculate the point cloud's minimum bounding box by finding the minimum and maximum values of the vertices in all dimensions.

We subsequently divide the bounding box into smaller cubes, thus creating a grid. We select a cube having the same size in all its dimensions. We refer to the size as $step$. The $step$ may be determined by the user and its selection is particularly crucial for the result's quality as well as for the frames per second rate that the system is capable of supporting. The proposed step is between 1 and 0.6cm.

Based on this $step$, we quantize the point cloud by matching each vertex with the nearest vertex of a cube. In parallel, we insert a three-dimensional logical array that we call $gridArray[X][Y][Z]$ and constitutes an index for the grid, and of course its dimensions depend on the bounding's box dimensions combined with the $step$. For example, the array element $gridArray[1][5][2]$ will correspond with the grid vertex having the coordinates $(X, Y, Z) = (X_{min} + 1 * step, Y_{min} + 5 * step, Z_{min} + 2 * step)$ and will indicate whether a point cloud vertex is near this area.

Following the creation of the logical array, we go across the sampled area using the table as an index, as stated herein above. We start from the first point $P_{start} \in \mathbb{R}^3$ of the grid, namely the one under the coordinates $(X_{min}, Y_{min}, Z_{min})$ and, after traversing all three dimensions with a step, we end up at the point $P_{end} \in \mathbb{R}^3$ where $P_{end} = (X_{min} + X_{max} - step, Y_{min} + Y_{max} - step, Z_{min} - Z_{max} - step)$. Considering a grid point $P \in \mathbb{R}^3$, where $P = (X, Y, Z)$ with $X_{min} \leq X \leq X_{min} + X_{max} - step, Y_{min} \leq Y \leq Y_{min} + Y_{max} - step, Z_{min} \leq Z \leq Z_{min} + Z_{max} - step$, we determine the cube vertices including point P as a vertex $(P = V_1)$ by adding vector P to vector $A_i \in \mathbb{R}^3, i = 1, .., 8.$ of each line in matrix A. Namely $V_i = P + A_i$, $i = 1, .., 8.$

With regard to each cube within the grid, based on the logical values of its 8 vertices, we form a surface by using the Marching Cubes algorithm.

$$A = \begin{pmatrix} 0 & 0 & 0 \\ step & 0 & 0 \\ step & step & 0 \\ 0 & step & 0 \\ 0 & 0 & step \\ step & 0 & step \\ step & step & step \\ 0 & step & step \end{pmatrix}$$

C. Data Transmission

For the purposes of the present implementation, we consider that geometrical processing takes place within the PC of the transmitter, whereas the texture mapping is implemented on the receiver's PC. Therefore, the information concerning the mesh and the image to be used as texture have to be transmitted. Understandably, the quantity of data depends on the point cloud vertices after the user's extraction, the algorithm $step$ during the process of mesh creation from the point cloud, the image resolution and format of the texture and the frames per second (fps) rate.

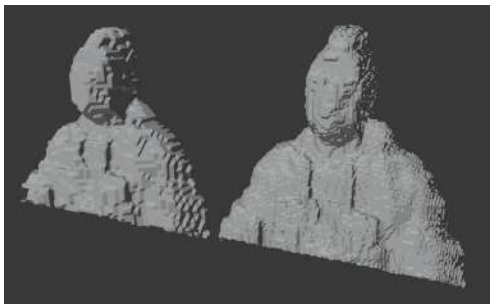

Figure 4. Mesh examples with *step* equal to 1cm and 0.7cm form left to right

For the transmission of data, the protocols TCP/IP were used. The algorithm's step and camera's resolution can be selected according to the available network's bandwidth.

D. Texture Mapping

The aforementioned procedure of geometry processing results in a textureless mesh. The image captured by the camera's left lens is then used as a texture. In order to find the appropriate UV coordinates of every vertex of the mesh, algorithm 1 was used, considering the focal length and optical center coordinates of the camera. Both parameters are defined in pixels. The exact focal length can vary depending on the camera calibration and selected resolution.

Algorithm 1 UVs calculation

Input: Mesh vertices, focal length, optical center and camera's resolution

Output: UV coordinates for each vertex

for every $vertex$ of Mesh **do**
  $U_{temp} \leftarrow (vertex.x * focalLength.x)/vertex.z + opticalCenter.x$;
  $V_{temp} \leftarrow (vertex.y * focalLength.y)/vertex.z + opticalCenter.y$;
  $U \leftarrow U_{temp}/resolution.x$;
  $V \leftarrow V_{temp}/resolution.y$;
end for

E. Display

The final rendering takes place in a virtual or augmented reality environment, depending on the available equipment of the receiver(s). We used the Meta2 augmented reality headset by Meta® for an AR experience.

IV. Results

A. Implementation statistics

The presented implementation statistics for the algorithm were extracted by taking into account the average implementation of the algorithm in twenty random frames. Three different sample tests are presented. Data size and execution time refer only to the mesh generation. The end point systems were connected via a local network. In all the experiments on the capture side we have used a PC Intel Core i5-9400F CPU, GeForce TRX 2060 OC 6G GPU, 8 GB RAM. The algorithms are written in C#.

Point Cloud Vertices ≈ 105000					
Step	**1cm**	**0.9cm**	**0.8cm**	**0.7cm**	**0.5cm**
Vertices	25K	30K	40K	50K	97K
Mesh data size (bytes)	400K	480K	640K	800K	1.552M
Execution Time (ms)	9	13	17	25	63

Point Cloud Vertices ≈ 120000					
Step	**1cm**	**0.9cm**	**0.8cm**	**0.7cm**	**0.5cm**
Vertices	39K	49.5K	62K	79K	115K
Mesh data size (bytes)	624K	792K	992K	1264K	2.4M
Execution Time (ms)	16	23	32	45	118

Point Cloud Vertices ≈ 167000					
Step	**1cm**	**0.9cm**	**0.8cm**	**0.7cm**	**0.5cm**
Vertices	40K	52K	67K	85K	155K
Mesh data size (bytes)	640K	832K	1072K	1.36M	2.67M
Execution Time (ms)	19	26	38	55	145

Table I
IMPLEMENTATION STATISTICS FOR THREE SAMPLE TESTS

Execution time indicates the frames per second rate that can be supported. For example, in order to achieve 30 FPS rate the execution time must be at most $1/30 = 33.33$ms.

B. Experimental Results

Experimental results from the proposed system are displayed in the figures 6 and 5. Fig. 6 shows two final meshes in front and side view. The step and camera resolution used were 1cm and 1280x720 respectively. Fig. 5 depicts the final rendering on the receiving side, in an AR environment.

Figure 5. Final rendering in an AR enviroment

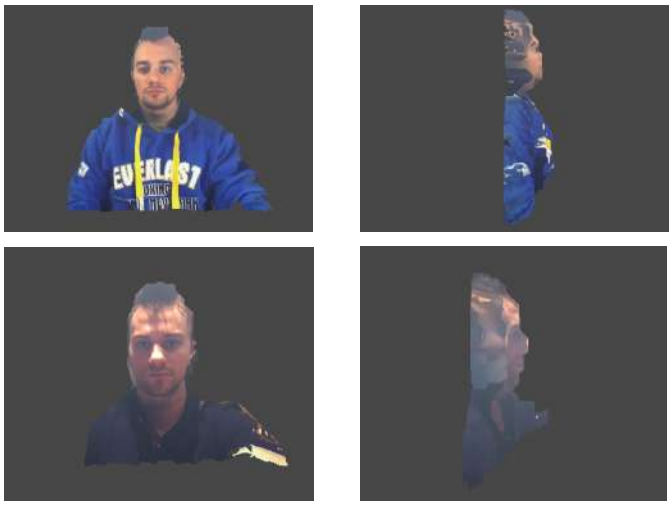

Figure 6. Mesh results with 1cm *step* and 1280x720 texture resolution, in front and side view

V. Conclusion

We have presented an end-to-end, one way, real-time 3D reconstruction system of the human body's upper part. The proposed system allows a more direct and realistic display of the user in mixed reality environments, requiring only a RGB-D camera on the capture side. With the huge development of augmented and virtual reality applications in recent years, we hope that live 3D capture will be a major way of human communication in the near future. Such tele-immersion systems prove to be valuable during periods where face-to-face communication is obstructed due to unforeseeable circumstances, as has unfortunately been the case with the Covid-19 global pandemic.

Acknowledgment

This work has been supported by the EU Horizon2020 funded project "Smart, Personalized and Adaptive ICT Solutions for Active, Healthy and Productive Ageing with enhanced Workability (Ageing@Work)" under Grant Agreement No. 826299

References

[1] H. Fuchs, G. Bishop, K. Arthur, L. McMillan, R. Bajcsy, S. Lee, H. Farid, and T. Kanade, "Virtual space teleconferencing using a sea of cameras," in *Proc. First International Conference on Medical Robotics and Computer Assisted Surgery*, vol. 26, 1994.

[2] T. Kanade, P. Rander, and P. Narayanan, "Constructing virtual worlds from real scenes," in *ACM Multimedia*, vol. 1, 1997.

[3] S. J. Gibbs, C. Arapis, and C. J. Breiteneder, "Teleport–towards immersive copresence," *Multimedia Systems*, vol. 7, no. 3, pp. 214–221, 1999.

[4] H. Towles, W.-C. Chen, R. Yang, S.-U. Kum, H. F. N. Kelshikar, J. Mulligan, K. Daniilidis, H. Fuchs, C. C. Hill, N. K. J. Mulligan *et al.*, "3d tele-collaboration over internet2," in *In: International Workshop on Immersive Telepresence, Juan Les Pins*. Citeseer, 2002.

[5] T. Peterka, R. L. Kooima, D. J. Sandin, A. Johnson, J. Leigh, and T. A. DeFanti, "Advances in the dynallax solid-state dynamic parallax barrier autostereoscopic visualization display system," *IEEE transactions on visualization and computer graphics*, vol. 14, no. 3, pp. 487–499, 2008.

[6] G. Kurillo, R. Bajcsy, K. Nahrsted, and O. Kreylos, "Immersive 3d environment for remote collaboration and training of physical activities," in *2008 IEEE Virtual Reality Conference*. IEEE, 2008, pp. 269–270.

[7] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang *et al.*, "blue-c: a spatially immersive display and 3d video portal for telepresence," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 819–827, 2003.

[8] T. Tanikawa, Y. Suzuki, K. Hirota, and M. Hirose, "Real world video avatar: real-time and real-size transmission and presentation of human figure," in *Proceedings of the 2005 international conference on Augmented tele-existence*, 2005, pp. 112–118.

[9] A. Maimone and H. Fuchs, "Real-time volumetric 3d capture of room-sized scenes for telepresence," in *2012 3DTV-conference: the true vision-capture, transmission and display of 3D video (3DTV-CON)*. IEEE, 2012, pp. 1–4.

[10] S. Beck, A. Kunert, A. Kulik, and B. Froehlich, "Immersive group-to-group telepresence," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 4, pp. 616–625, 2013.

[11] D. S. Alexiadis, D. Zarpalas, and P. Daras, "Real-time, realistic full-body 3d reconstruction and texture mapping from multiple kinects," in *IVMSP 2013*. IEEE, 2013, pp. 1–4.

[12] C. Zhang, Q. Cai, P. A. Chou, Z. Zhang, and R. Martin-Brualla, "Viewport: A distributed, immersive teleconferencing system with infrared dot pattern," *IEEE MultiMedia*, vol. 20, no. 1, pp. 17–27, 2013.

[13] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou *et al.*, "Holoportation: Virtual 3d teleportation in real-time," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 741–754.

[14] P. Stotko, S. Krumpen, M. B. Hullin, M. Weinmann, and R. Klein, "Slamcast: Large-scale, real-time 3d reconstruction and streaming for immersive multi-client live telepresence," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 5, pp. 2102–2112, 2019.

[15] "Api documentation: Api reference." [Online]. Available: https://www.stereolabs.com/docs/api/

[16] L. Ladický, O. Saurer, S. Jeong, F. Maninchedda, and M. Pollefeys, "From point clouds to mesh using regression," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 3913–3922.

[17] E. Piazza, A. Romanoni, and M. Matteucci, "Real-time cpu-based large-scale three-dimensional mesh reconstruction," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1584–1591, 2018.

[18] W. Lorensen and H. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 21, pp. 163–, 08 1987.