

Article

Real-Time 3D Reconstruction Method for Holographic Telepresence

Fazliaty Edora Fadzli ^{1,2,3,*} , Ajune Wanis Ismail ^{1,2,3}, Shafina Abd Karim Ishigaki ^{1,2,3},
Muhammad Nur Affendy Nor'a ^{1,2,3}  and Mohamad Yahya Fekri Aladin ^{1,2,3}

- ¹ School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM),
Johor Bahru 81310, Johor, Malaysia; ajune@utm.my (A.W.I.); shafina@graduate.utm.my (S.A.K.I.);
mnaffendy2@graduate.utm.my (M.N.A.N.); myfekri2@graduate.utm.my (M.Y.F.A.)
² Mixed and Virtual Environment Research Lab (Mivielab), Universiti Teknologi Malaysia (UTM),
Johor Bahru 81310, Johor, Malaysia
³ ViCubeLab, Universiti Teknologi Malaysia (UTM), Johor Bahru 81310, Johor, Malaysia
* Correspondence: fedora3@graduate.utm.my

Abstract: This paper introduces a real-time 3D reconstruction of a human captured using a depth sensor and has integrated it with a holographic telepresence application. Holographic projection is widely recognized as one of the most promising 3D display technologies, and it is expected to become more widely available in the near future. This technology may also be deployed in various ways, including holographic prisms and Z-Hologram, which this research has used to demonstrate the initial results by displaying the reconstructed 3D representation of the user. The realization of a stable and inexpensive 3D data acquisition system is a problem that has yet to be solved. When we involve multiple sensors we need to compress and optimize the data so that it can be sent to a server for a telepresence. Therefore the paper presents the processes in real-time 3D reconstruction, which consists of data acquisition, background removal, point cloud extraction, and a surface generation which applies a marching cube algorithm to finally form an isosurface from the set of points in the point cloud which later texture mapping is applied on the isosurface generated. The compression results has been presented in this paper, and the results of the integration process after sending the data over the network also have been discussed.

Keywords: 3D reconstruction; telepresence; telepresence holographic



Citation: Fadzli, F.E.; Ismail, A.W.;
Abd Karim Ishigaki, S.; Nor'a,
M.N.A.; Aladin, M.Y.F. Real-Time 3D
Reconstruction Method for
Holographic Telepresence. *Appl. Sci.*
2022, *12*, 4009. [https://doi.org/](https://doi.org/10.3390/app12084009)
10.3390/app12084009

Academic Editors: Deokwoo Lee and
YoHan Park

Received: 31 January 2022

Accepted: 6 April 2022

Published: 15 April 2022

Publisher's Note: MDPI stays neutral
with regard to jurisdictional claims in
published maps and institutional affili-
ations.



Copyright: © 2022 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the Creative Commons
Attribution (CC BY) license ([https://](https://creativecommons.org/licenses/by/4.0/)
[creativecommons.org/licenses/by/](https://creativecommons.org/licenses/by/4.0/)
4.0/).

1. Introduction

Telepresence can be defined in several ways. According to Minsky (1980) [1], telepresence was defined as remote control tools in reference to teleoperation systems. Steuer (1992) [2] has defined telepresence as the experience of feeling a presence by using the communication medium. Shih (1998) [3] understands telepresence as the degree to which a user can feel their existence in a virtual space. In our modern time, telepresence can be defined as a technology that can provide the feeling of being present in another remote location which enables both a remote and a local user to communicate. The goal of telepresence is to create the feeling of being there or physically present with a remote person [4]. It is possible to have telepresence through a natural-sized image of the person, a 3D representation of the user or their environment, or the interaction between the remote user and the local user. A visual sense can be created by several paradigms, including a remote person appearing in a local setting [5,6], a remote space that appears to be stretched beyond the local surroundings [7,8], as well as a local user who is immersed in a remote setting [9,10]. The 3D reconstruction method is significant to produce human representation. Telepresence combined with 3D reconstruction technology has the potential to alleviate many of the limits associated with standard video-based communication, such as limited natural movement in 3D space, situational awareness, gaze direction, and eye contact due to the fixed viewpoint of the remote participant's camera, as supported by Pejosa et al. [11] which has evaluated

the comparison between traditional video communication such as Skype with their 3D reconstruction integrated with the telepresence system called Room2Room. The result findings show that their systems show significant improvement in presence and completion time. Fuchs, H. et al. [12] also agrees that 3D reconstruction integrates with telepresence and can also offer the potential for remote collaboration, virtual 3D object manipulation, or exploring the remote site. Modern telepresence technologies enable persons in remote areas to virtually meet and interact with one another through the use of realistic 3D user representations.

While display technology has evolved, the ability of the display to replicate a 3D environment remains a critical component. People have been working to advance 3D display technology for years, and researchers have gathered and produced a variety of inventions in recent decades, as well as benefiting from technologies that involve the viewing of 3D components, such as 3D telepresence and virtual reality (VR). Holographic projection is regarded as one of the most promising true 3D display technologies in the near future. This technology can also be implemented in many different forms, such as holographic prisms [13], Z-Holograms [14], HMDs [15], and others.

This study, however, wants to discard the use of any specialized equipment such as head-mounted displays to enable the remote user to move freely while interacting with the other user. This study shows the initial results of displaying the human 3D reconstruction using the Z-hologram display that incorporates the Pepper's Ghost method to create the illusion as if the reconstructed user appeared as a 3D representation in the real-world environment. As demonstrated in Figure 1, holographic projection technology is based on an illusionary method known as Pepper's Ghost, which was initially utilized in Victorian theatres around London in the 1860s [16]. A brightly lighted figure below the stage, hidden from the public's view, is mirrored in a pane of glass between the performer and the audience. It appears to the audience that the ghost is present on stage. A hologram is a picture of interference and diffraction on the surface of a real 3D object captured on a specific optical film or glass from all point light sources of a projected target item [17].

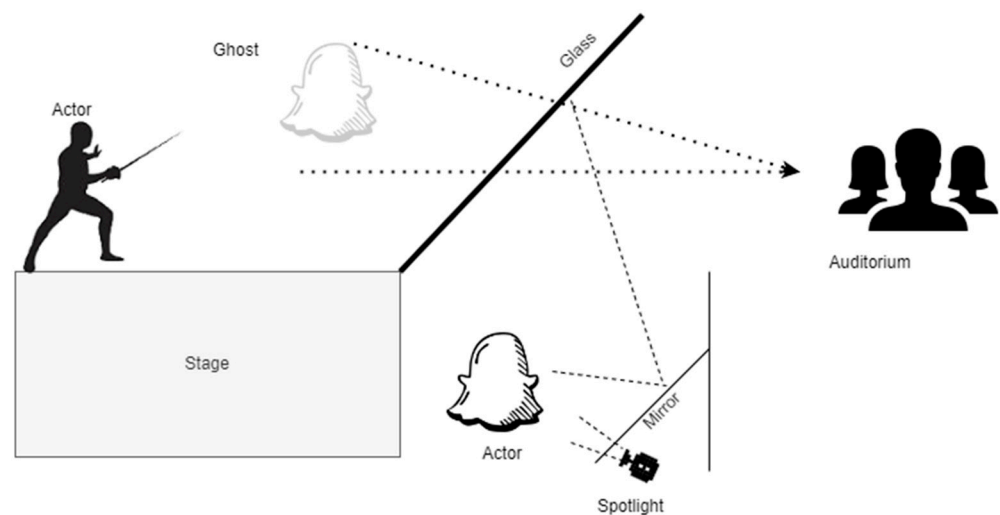


Figure 1. The schematic illustration of stage setup for Pepper's Ghost.

This research was inspired by earlier works related to 3D reconstruction for telepresence. Researchers have been intrigued by the prospect of 3D telepresence for decades, but because of technological limitations in the past, prototypes have only just begun to appear in the marketplace. Multiple cameras are deployed in prior works in order to build a 3D reconstruction of a room-scaled scene [11], and their images are constantly changed to include the moving user, utilizing a variety of stereo reconstruction [18] techniques. This enables the reconstruction of the scene into 3D representation. When inexpensive depth sensors, for instance, the Microsoft Kinect [19], became widely available at low cost

and were capable of acquiring video images as well as per-pixel depth information and were accessible, the total amount of studies and developments related to 3D reconstruction merged with telepresence systems increased substantially [20]. Therefore, this paper further discusses the 3D reconstruction process involved to reconstruct a 3D representation of the captured user using RGB-D camera implemented with holographic telepresence. The test application and the results are also presented, and this paper ends with a conclusion.

The first phase is designed to produce the real-time 3D reconstruction of the local user where the flow is as demonstrated in Figure 2. When the data is obtained using the commodity depth sensor, which has been pre-processed, the next step is to remove the background in order to get the foreground mask which contains only the captured user's body without the background. After we obtain the foreground mask of the captured data, the 3D point cloud is extracted. For the surface production of the 3D model, the marching cube algorithm is then used with point cloud data as the input. The resulting 3D model has then applied the texture mapping.

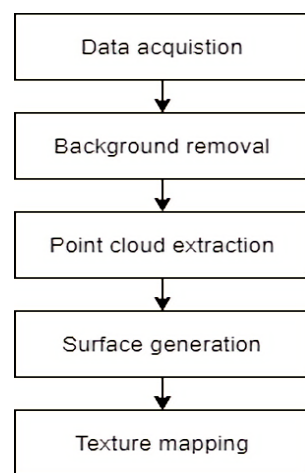


Figure 2. Diagram flow for the 3D reconstruction method.

After the textured 3D model of the local user has been reconstructed, the data, which includes mesh, texture and audio, will be transmitted to the remote location through transmission over the network. The experimental setup for the 3D telepresence will then be set up for the test application.

2. Real-Time 3D Reconstruction Method

2.1. Data Acquisition

Microsoft Kinect V2 has been utilized to capture the local user in the data acquisition process. Data captured by depth sensors can be noisy and needs to be pre-processed to recover the missing depth values at some defective areas in the depth map. The depth holes can be filled by using a hole-filling algorithm [21], after which the depth map will be smooth, using a bilateral filtering algorithm [22]:

$$D(p(x, y)) = \frac{1}{w_n} \sum_{l=0}^{m-1} w(x_l, y_l) \times D_l \quad (1)$$

The depth information of the depth hole in the depth map can be filled using the hole-filling formula shown with m be a constant, where $p(x, y)$ will be the filling target point, D_l presents the l^{th} depth value, w_n denotes the normalization parameter, and the $w(x_l, y_l)$ is the weight coefficient. The hole in depth map will be filled through this hole-filling processing step. After this process, the traditional bilateral filtering algorithm is used to smooth the depth map with holes that have been filled.

Since the RGB-D cameras have been utilized, the instrument parameters for a general colour-depth sensor for 3D reconstruction are the distance difference and depth resolution. Distances between the reconstructed target points were compared to the ground truth measurements. Using Kinect, the object was then captured from two different distances to get measurements between 0.7 m to 2 m. RGB-D cameras and the corresponding depth used for scene reconstruction, then depth resolution was measured by moving the Kinect away (0.5–15 m) from a planar target in sufficiently fine steps to record all values returned in an approximately 5° view field around the image centre. Considering the depth resolution, when the depth image has been constructed by triangulation from the IR image, the depth resolution for triangulation-based devices such as the Kinect is expected [23]. However, this paper is limited to explaining the marching cube algorithm for 3D reconstruction, therefore it does not provide the evaluation to count on instrument parameter for its accuracy.

2.2. Background Removal

An RGB-D camera has been utilized during the background removal process to capture the local user. The Kinect V2 sensors have captured the colour and depth images. From the depth image, the body index image contains a 2D grid where each coordinate gives a simple 0 to 5 integer representing which body the sensor associates with that coordinate.

The background removal from the depth image to generate the body index image is required as this research only focuses on reconstructing the human. Thus, the foreground extraction step is needed and can be done by segmenting the foreground pixels from the background pixels in depth images. In order to extract foreground pixels, the foreground or background information is converted into binary ones and zeros, respectively.

The body index image includes the instance segmentation map for each body in the depth camera capture. Each pixel maps to the corresponding pixel in the depth or IR image. The value for each pixel represents which body the pixel belongs to. It can be either background or the index of a detected body. We obtained the human body segmentation mask using this body index image mapped together with the colour image.

2.3. Point Cloud Extraction

In this research, we used two Kinects, and the multiple sensors were used simultaneously. In order to obtain the 3D points, the $[x, y, z]$ coordinates of each pixel inside the human body segmentation mask used perspective projection with $z = \text{depth}$. Then, the $[x, y, z]$ points were projected onto the colour frame using the intrinsic and extrinsic parameters of the colour camera to extract the corresponding colour of each 3D point.

The 3D-coloured points (X, Y, Z, R, G, B) of each depth pixel using perspective projection are obtained using the following equation.

$$\begin{aligned} X &= \frac{\text{depth} \times (x - cx)}{f} \\ Y &= \frac{\text{depth} \times (y - cy)}{f} \\ Z &= \text{depth} \end{aligned} \quad (2)$$

where (cx, cy) is the principal point and f is the focal length. To fuse the coloured 3D data, the extrinsic parameters of each camera, i.e., the poses between each camera and the reference will be used to transform all the point clouds into a single reference frame. The next step is to refine the initial estimate of the extracted point cloud using iterative closest points (ICP) [24] with Algorithm 1.

Algorithm 1: Iterative Closest Point (ICP)

Input: $F_1 \dots F_n$ —3D point cloud from each sensor
Data: $R_1 \dots R_n, t_1 \dots t_n$ —initialized to identity
Result: $R_1 \dots R_n, t_1 \dots t_n$ —refinement transform for each sensor
 $v = 1; \tau = 0.01; \text{firstPass} = \text{true}; e = 0; \hat{e} = 0$
Step:
while $v > \tau$ **do**
 $e = 0$
 for $i = 1$ **do**
 $K = \{1, 2, \dots, n\} \setminus i$
 /* ICP aligns the i -th shape to all the other ones, updates
 R_i, t_i and returns the average error per point */
 $e = e + \text{ICP}(F_K, F_i, R_i, t_i)$
 end
 if $\text{firstPass} == \text{true}$ **then**
 $\text{firstPass} = \text{false}$
 $v = 1$
 else
 $v = \frac{|e - e|}{e}$
 end
 $\hat{e} = e$
End

We want to construct, within each cube, a 3D model that is usually constituted by a triangular mesh that correctly represents the geometry and topology of the isosurface. The surface generation from point cloud data will later be implemented with the extended marching cube algorithm, which requires a discrete set of cubes as mentioned in Custodio et al. [25]. ICP has been used to divide the input volume into a discrete set of cubes. This is the primary reason the ICP can avoid degenerate triangles and guarantee topological correctness during the surface generation stage.

2.4. Surface Generation

Next, the surface generation from point cloud data that has been extracted is the process where the unorganized points in the point cloud were connected and generated a set of triangles that closely approximates a surface of interest. The Marching Cube algorithm takes voxel data and extracts an isosurface, a polygonal mesh surface representation. Each voxel has a value, and the marching cube algorithm attempts to create a mesh for the surface at a specified iso value. When it comes to an isosurface, the implicit function $f(x, y, z) = c$ is generally what defines it. This means that all 3D points in the volume part of a c -surface satisfy the previous criterion. When we use Marching Cubes, the 3D volume is divided into voxel squares of equal size. The predefined triangular pattern is possible. A triangular pattern is applied to voxels that intersect the isosurface to approximate the course of the isosurface within the cell.

In this research, discrete point cloud data are processed using the techniques presented in Hoppe et al. [26]. We decided to use this algorithm because it would provide insight into the fundamental concepts behind several existing surface reconstruction methods. It was possible to reduce the algorithm's complexity by using a set of points on an unknown surface with their normals. The Marching Cube algorithm has been proven by Stotko et al. [27], who claim that it is very compact and easily manages the data which has and can be a benefit for telepresence, which requires immediate transmission as well as fast and compact data structures to allow for reconstructing and providing a virtual 3D model in real time to remote users. Kowalski et al. [24] agreed that the enhancement needed to produce a fast and inexpensive 3D data acquisition system for multiple Kinect sensors.

The modification of the Marching Cube algorithm is based on [25]. In Algorithm 2, the step in number 7 is used a looping foreach voxel recalculating and generating process. The

cube vertices are insufficient to determine the correct surface triangulation. The original marching cube does a lookup into one of eight different cases as in the yellow frames in Figure 3a that shows as the 4th, 6th 7th, and 10th and 11th cubes in the modified lookup table. Modifying the marching cube to triangulate the implicit surface allows the possibility of the prescribing boundaries to increase, and it becomes sufficient. Extended marching cubes then look at each voxel's boundaries and do a lookup into one of 14 different cases in blue frames, as shown in Figure 3b. The mesh associated with the looked-up case is added in place of the voxel. After all voxels have been processed, the result is a set of mesh triangles that approximate the mesh from which the point cloud was created.

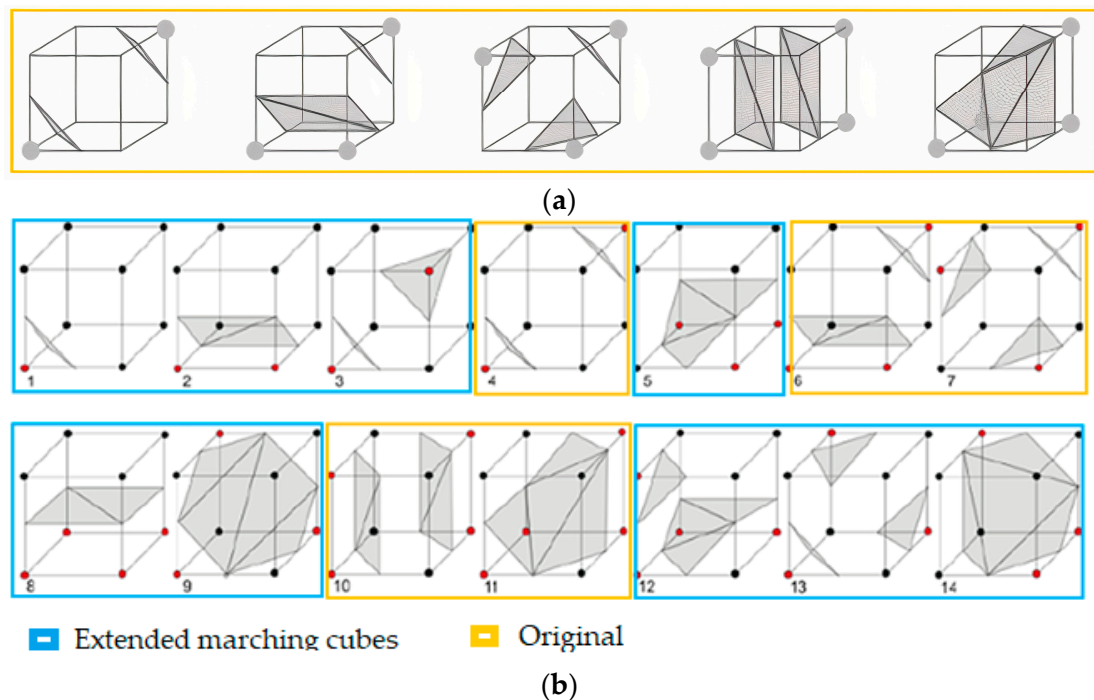


Figure 3. The modified Marching Cube triangulation cases lookup-table. (a) The original Marching Cube triangulation cases. (b) The modified Marching Cube triangulation cases.

2.5. Texture Mapping

Texture mapping can help produce a visually appealing model by applying high-quality textures on a 3D mesh with minimal geometric complexity, as claimed by [29]. Additionally, it is necessary to consider that voxel-based 3D reconstruction methods generate, on average, fewer triangles and vertices than the depth map's initial 2D resolution in pixels, depending on the number of voxels [29]. Therefore, using a colour-per-vertex approach results in colour aliasing, leading to insufficient quality. We use complete texture mapping to address this matter, which entails projecting vertex points onto colour images to get the UV coordinates. The texture is then assigned per triangle as suggested in [30].

A process involving transformation from 3D mesh vertices $v = (x, y, z)^T \in V$ in relation to an object space into a texture coordinate $t = (u, v)^T \in \Omega_T$ on a 2D texture image $T: \Omega_T \rightarrow \mathbb{R}^3$ is known as texture mapping. Using UV coordinates to define a triangle or fragment permits the storage of many texels (texture elements) $t \in \Omega_T$, each of which holds an RGB colour value. After we had the reconstructed 3D model, the model's vertices were projected onto the colour image planes to acquire colour correspondences in the neighbouring colour frame, resulting in a textured 3D model.

Algorithm 2: Marching Cube

Input: Sampling dataset $P = \{p_1 \dots p_n\}$ with $p_i \in \mathbb{R}^3$, marching cube vertex, v
Output: surface $S \subset \mathbb{R}^3$ which approximates P , a list of vertices to be rendered with their respective normals.

Step:

1. Examine for each point in the local neighbourhood, set of k nearest neighbours
2. Compute best approximating tangent plane
3. Find normal for each sample point p_i
4. Compute z as the projection of p onto $Tp(x_i)$
5. **if** the distance to the sample point is nearest to $z < P$ **do**
 return sign of the distance function, $f(p)$ as positive

6. **else**
 return sign of the distance function, $f(p)$ as negative

7. **while** $f(p) > 0$
 for each voxel (cube) c_i of the dataset **do**
 Calculate an index to the cube, by comparing the cube vertex's
 v , 8 density values to the iso value, h
 Verify the edges list from a lookup table by using the
 calculated index.

Linearly interpolate to find the surface-edge intersection based on the scalar values in each vertex of the edge.

Using of central differences method [28] to compute a unitary normal in each cube vertex. Interpolate each triangle vertex's normal

Return the vertex normal and the triangle vertices.

8. **end for**

9. **end while**

End

3. Proposed Real-Time 3D Reconstruction for Holographic Telepresence

After the real-time 3D reconstruction phase has been done, the next step is to enable the 3D telepresence. The textured 3D model of the local user produced was then transferred to a remote location through the network to be displayed at a remote location in real-time to enable the local and remote user to engage with one another through the telepresence. The overall framework of the proposed real-time 3D reconstruction for a 3D holographic telepresence is shown in Figure 4.

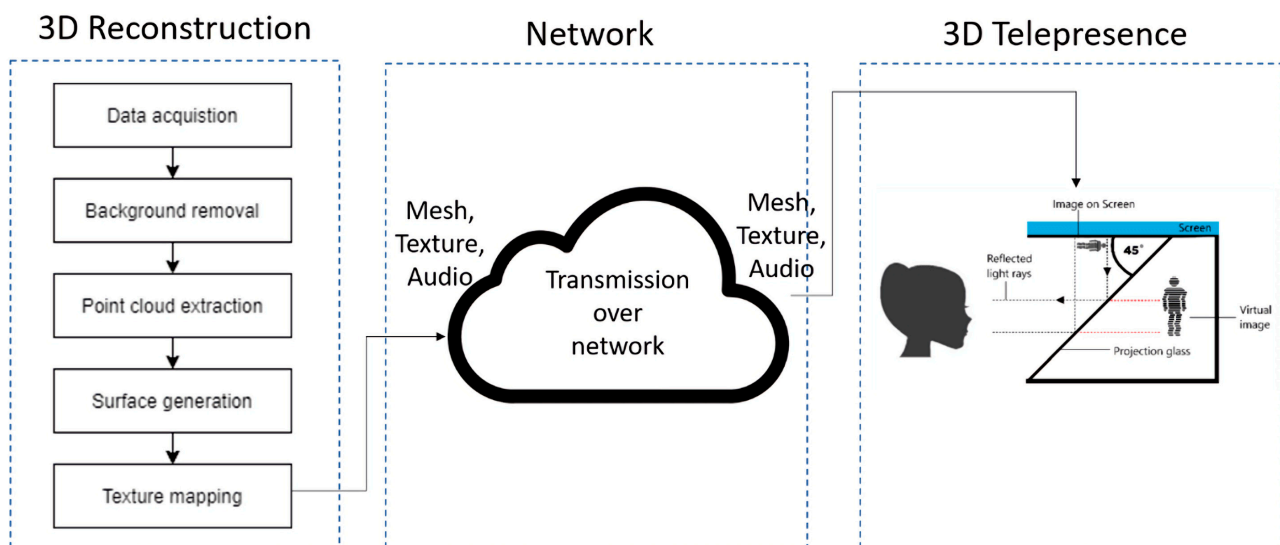


Figure 4. The framework of real-time 3D reconstruction for holographic telepresence.

The textured 3D model, along with the audio of the local user, is then transmitted over the network to the remote location, as in Figure 5. These data will first be encoded to be compressed according to the suitable bandwidth of data transmission through an internet protocol. After the data was sent and received at the remote location, the data was decompressed using a lossless compression method on the remote client laptop and ready to be displayed in the reconstructed textured 3D model of the local user. The lossless compression method used for this research is the Lempel-Ziv-Free (LZF) compression, as recommended by Waldispühl et al. [31]. This compression algorithm requires a minimal code space and working memory. The mesh data with audio input of remote users was also sent over to the local location through the network.



Figure 5. The data transmission over the network for the telepresence system.

Once the data sent from the local site were received on the remote site, the mesh data was rendered and displayed using holographic projection. As illustrated in Figure 6, Z-Hologram was chosen as the display device to project the virtual reconstructed local user onto a real-world environment for remote users to experience and interact with it.

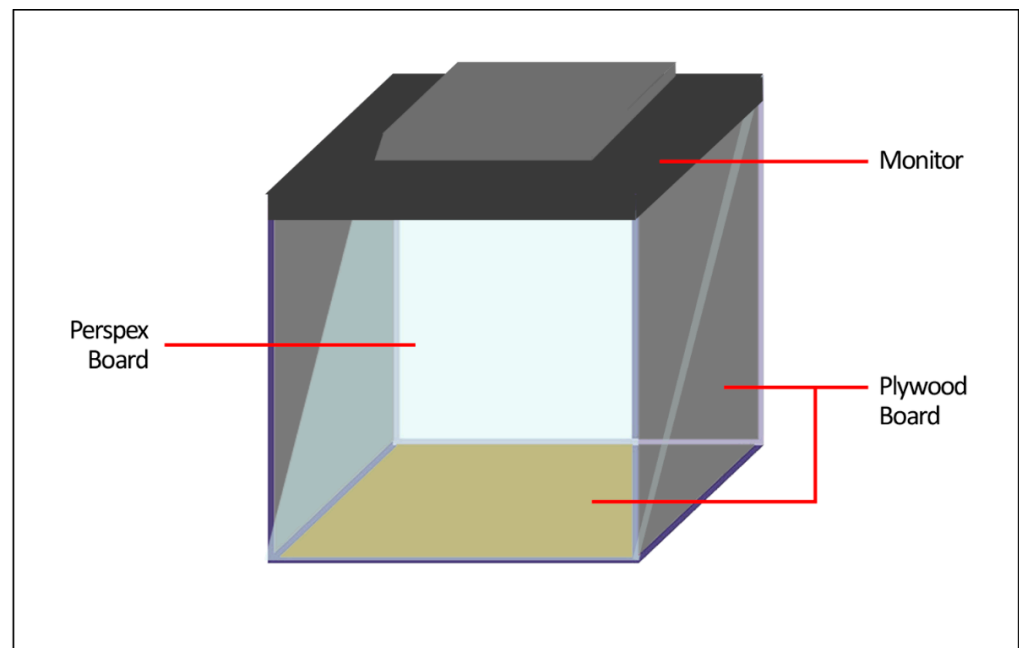


Figure 6. The Z-Hologram setup structure.

For Z-Hologram, the effect of floating virtual 3D images for the viewers is formed when the reference beam and the object or image source beam are incident on opposite sides of the reflective transparent surface, as illustrated in Figure 7. The beams interfere and record an image called a holographic image. To reconstruct the image, a point source of white light illuminates the hologram from the proper angle, and the viewer looks at it from the same side as the light source.

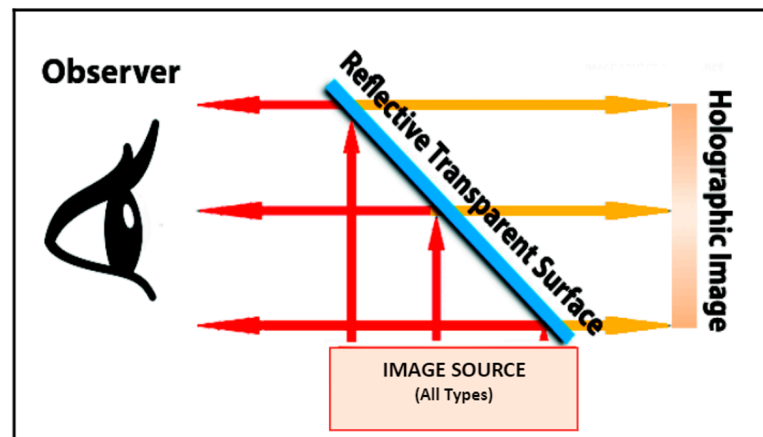


Figure 7. The reflective ray principle of the hologram.

As agreed with by Oh and Kwon [32], the hologram technique can enhance the realism and the immersion to provide the realizable 3D stereoscopic vision when the floating hologram projecting the 2D image to represent the 3D image in the air uses the glass panel. Yang et al. [33] also claimed that, using the Pepper's Ghost principle, a 2D image satisfying certain psychological depth cues such as occlusion can be displayed, thereby providing the viewers with 3D feelings. Thus, the 3D data that has been generated and displayed using the Z-hologram can be perceived by the user in 3D and helps provide a more absorbing experience of viewing the 3D data for the user. Using Z-hologram can benefit from 3D data as it able to provide the viewer with 3D impressions without the use of wearable hardware. Besides, a VR headset does not support a holographic display because any object inside VR is considered as a virtual object. Holographic projection is when the object is projected onto the real-world. This article also mentions that the use of hardware can restrict user movement and cause discomfort to the user.

4. Results and Discussion

This section explains the experimental setup for real-time 3D reconstruction for 3D telepresence. The test application of our work will be further explained in this section as well.

The experimental setup is presented in Figure 8a, which shows equipment such as desktop computers, a Kinect and a tripod, as well as its placement used for data acquisition of the local user in Figure 8b. An RGB-D camera was placed with appropriate height, h at the distance, d , for enabling data acquisition using the Kinect V2 to gather the depth image and colour image of the depth used for life-sized 3D reconstruction. The depth camera field of view (FOV), α° will be set to a suitable angle to enable the life-size capture of the local user's full body.

The Z-hologram is built with several components: a monitor screen, Perspex board, PVC pipe, and plywood. Firstly, the hologram is built with the Perspex board according to the measurement of the monitor screen such that the holographic content fits onto the Perspex board. The monitor then projects the reconstructed user onto the Perspex board. The reconstructed user appeared in the real environment when the projection from the monitor was reflected on the Perspex. The Plywood board is used to support the monitor placed on the board such that the monitor is stable on the Perspex and acts as the wall structure for the hologram. The result is shown in Figure 9.

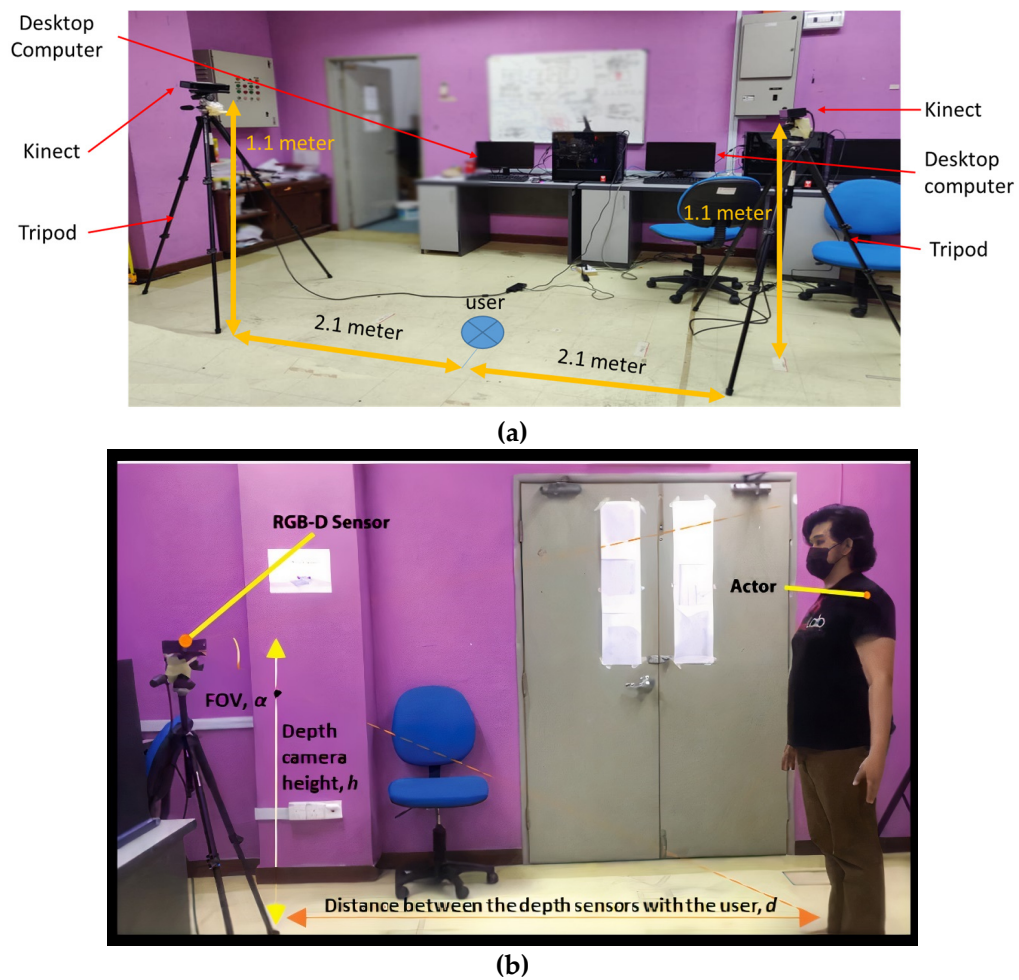


Figure 8. The experimental setup. (a) The experimental setup using two Kinect. (b) The setup with user.

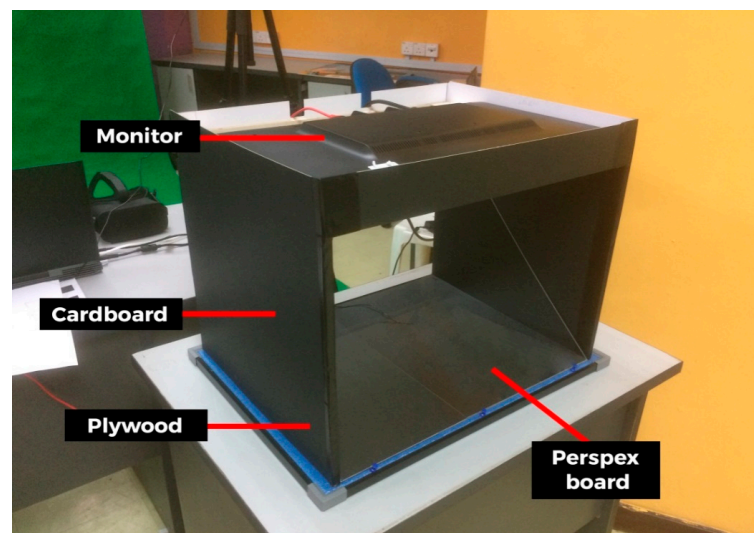


Figure 9. The Z-Hologram that was built for this research.

A test application has been prepared to integrate the real-time 3D reconstruction with 3D telepresence. First, both local and remote users are required to set roles as either sender or receiver using the simple user interface (UI) of the application.

As explained in the previous section, the research framework has a few phases, including 3D reconstruction and 3D telepresence. The results for each process of the 3D reconstruction using the marching cube algorithm to a produced textured 3D model of the local user are as illustrated in Figure 10. The first stage generates depth streams during data acquisition from the RGB-D sensor. Data acquisition was capturing and processing data input from each of the RGB-D cameras on the client site. The captured data will later be transmitted to the server using a stable network with background removal. During point cloud extraction, the data is captured from multiple sensors. After projecting from depth field to point cloud, a virtual scene is reconstructed from a sequence of point clouds. This process is called point cloud registration, where the algorithm has been used to estimate the rigid transformation between the two datasets of point clouds, which can then be used to merge the two sources of point clouds. This registration process can then extract objects and transform the point cloud data into surface information to recreate a 3D mesh. Up to the surface generation stage, we have confirmed that the application was still running in real-time.

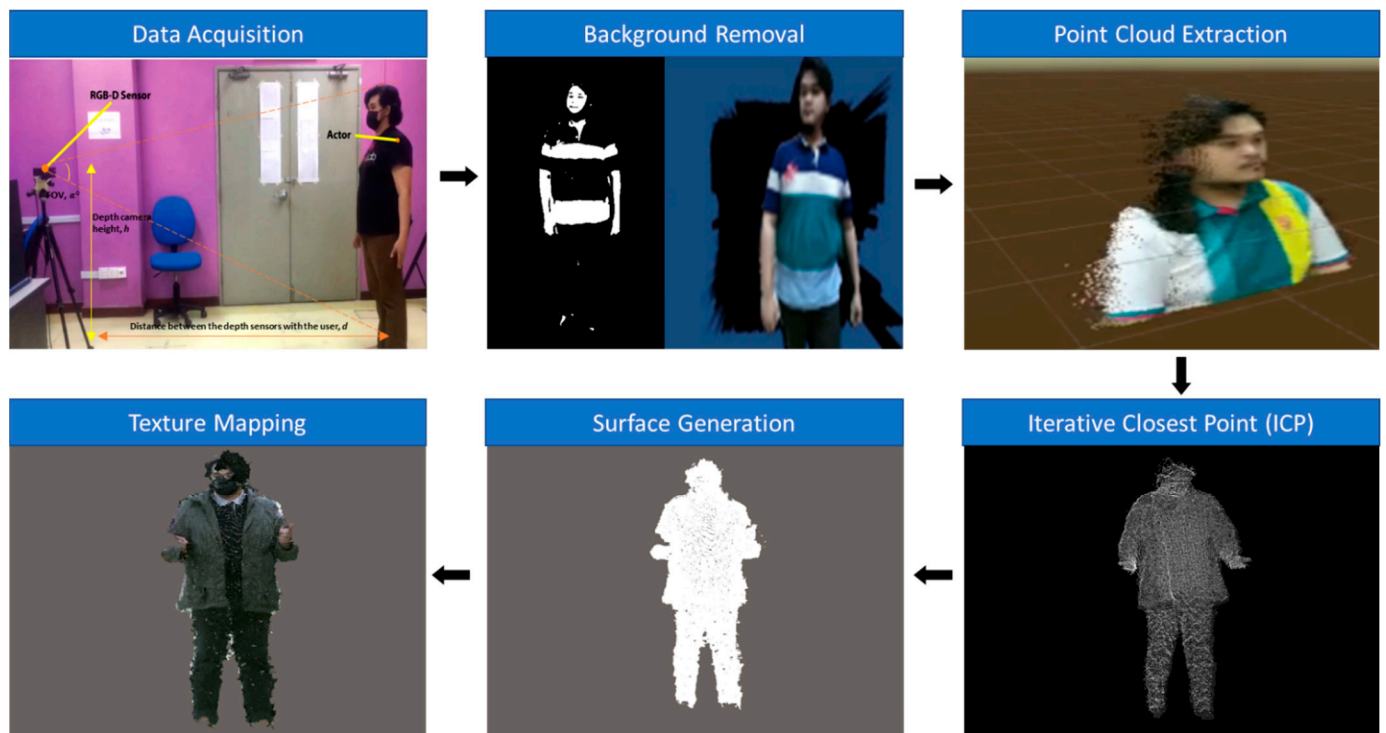


Figure 10. The real-time 3D reconstruction result from each stage.

Based on Figure 10, the background removal has been using a depth map, and the foreground pixels were extracted when the foreground or background information was converted into binary ones and zeros, respectively. The corresponding point set estimation algorithm fused the point cloud projected using the filtered depth map captured from the RGB-D sensor. During surface generation using the Marching Cube algorithm, the voxel grid used was set to the size of the depth frame height and width before re-evaluating the grid, calculating each square to find acceptable triangles to generate the surface. Then, the attempt to create the calculated texture map was initiated for the texture mapping process. In Table 1 we have presented the processing time measurement for the real-time process except for UV mapping. Table 1 shows the results we have compared to previous research. Our method used two Kinects, although we aim to reduce the number of Kinects; later, we measured the processing times, fps and the number of triangles. The fps rate is highest and the processing time is faster. Next, we performed the integration process by sending the data over the network.

Table 1. The processing time records, the results in a comparison with previous work.

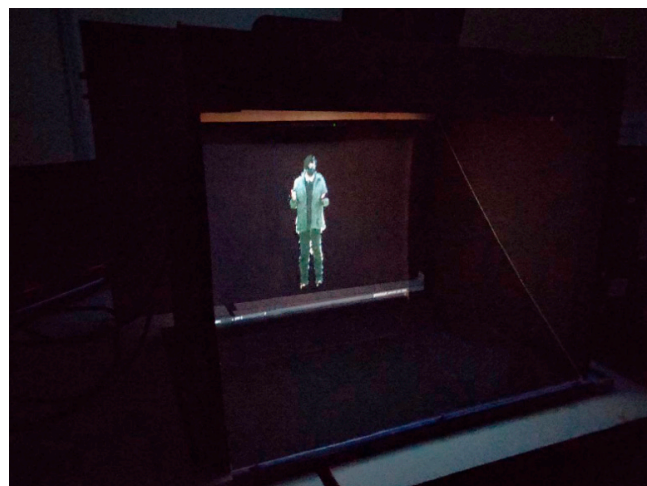
Research	Number of Kinects	Processing Time (ms)	Processing Rate (fps)	Average Number of Vertices	Average Number of Triangles
This research	2	27	78	2274	12,303
Alexiadis et al. [30]	5	78	13	228,581	445,447

Based on Table 1, our method used two Kinects compared to five Kinects, and the average numbers of triangles is significantly different. Our processing rate in frames per second produced higher than 78 fps compared to Alexiadis et al.'s [30] findings. After the 3D-textured model of the local user has been reconstructed, the 3D data such as mesh and texture along with the audio of the local user will be compressed and transmitted across the network to the remote site after being assembled in a UDP packet. When the packet is received in the remote client site, the packet will be disassembled and the data will be decompressed before being displayed on the remote user's PC. The experiment will not use green screen settings as it did in the present case [34] in order to ensure that the device is used in practical situations of natural environments.

Figure 11 shows the results of the 3D telepresence where a remote user can view and interact with a reconstructed 3D textured model of a local user sent to the remote location over the network. We have tested displaying the reconstructed 3D textured model using a small-scale Z-hologram. Our method has been loaded into the telepresence system to present the 3D presence of user representation in real-time, and we compare our results with Cordova-Esparza [13], whose work has performed real-time 3D reconstruction and was loaded into telepresence, and who also recorded their findings after the compression process in a color image, UV and compression data from client to server. We also performed the same process and recorded our findings. Table 2 shows the comparison; based on compression from raw data to final data, there are significant differences, especially when we requested that the server perform the telepresence and communicate with client user. However, the texture mapping required an improvement in real-time. With UV mapping compression we cannot compare with Cordova-Esparza [13], as they have not covered the UV compression in their method. However, we still present our UV compression results in Table 2.



(a)



(b)

Figure 11. The initial result of displaying the (a) reconstructed 3D model of the user (b) in Z-hologram.

Table 2. Our proposed real-time 3D reconstruction compression result when sending to the server in Z-hologram.

Research	Color Image		UV		Client to Server	
	Raw	Compression	Raw	Compression	Raw	Compression
This research	868,352	433,613	37,620	10,790	1,062,478	506,692
Cordova-Esparza [13]	6,220,800	76,800	-	-	6,654,976	111,534

5. Conclusions

Over the years, people have tried to advance the development of 3D reconstruction with different innovations and technologies such as telepresence. This paper presents the marching cube algorithm in a real-time 3D reconstruction and integrates it with 3D telepresence technology. Firstly, this paper introduced 3D reconstruction for the telepresence system and discussed a few related works. Each process in the 3D reconstruction, which includes the data acquisition, the removing of the background, and the extracting of the point cloud, which was then used later to generate the surface and applied texture mapping of the local user in real-time, has been explained in this paper. The study also presented an application to implement the proposed method and framework. The application used to test our 3D reconstruction to run real-time in holographic display. The process explained in this article has shown that the 3D reconstruction of a user can be generated using commodity RGB-D cameras integrated with a telepresence system which can benefit the remote user when the 3D representation of the local user is displayed in the Z-hologram, which can be perceived in 3D.

In conclusion, in this paper we have examined the flow of the 3D reconstruction method. The method involved several processes, which include the capturing and data acquisition from the depth sensor, the background removal, and the point cloud extraction. The surface generation process using the Marching Cube algorithm has been explained in this paper before the process ended with texture mapping. Based on the results, the real-time 3D reconstruction that employed the marching cube algorithm to generate 3D human representation has been successfully merged with the telepresence system and displayed at the remote location. However, the communication cues and flow have not yet been covered in this paper. For future works we plan on extending our work using holographic projection using a projector-based device to display the resulting 3D reconstruction output to emphasize the 3D telepresence more. We also plan to perform an evaluation to measure the accuracy and the quality of point cloud data. The results regarding the network requirements such as the delay would be areas of our future investigation. Therefore, we conclude that this paper has presented 3D telepresence with communication features by using a holographic projection as a display. As highlighted in this paper, the processes in producing the real-time 3D reconstruction consists of data acquisition, background removal, point cloud extraction, and surface generation which applies a Marching Cube algorithm as a final output.

Author Contributions: Conceptualization, F.E.F. and A.W.I.; methodology F.E.F. and A.W.I.; software, M.N.A.N., F.E.F. and S.A.K.I.; validation, A.W.I., M.Y.F.A. and M.N.A.N.; formal analysis, F.E.F., A.W.I., S.A.K.I. and M.N.A.N.; investigation F.E.F., M.N.A.N., M.Y.F.A. and S.A.K.I.; resources, F.E.F., A.W.I. and S.A.K.I.; data curation, F.E.F. and S.A.K.I.; writing—original draft preparation, F.E.F.; writing—review and editing, F.E.F. and A.W.I.; visualization F.E.F., M.N.A.N. and S.A.K.I.; supervision, A.W.I.; project administration, A.W.I.; funding acquisition, A.W.I. and F.E.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Fundamental Research Grant Scheme by Ministry of Higher Education Malaysia VOT R.J130000.7851.5F401.

Informed Consent Statement: Informed consent was obtained from the subject involved in the study who is one of this article authors.

Data Availability Statement: Not applicable.

Acknowledgments: We are most grateful to the Mixed and Virtual Reality Laboratory (Mivielab) and the ViCubeLab at Universiti Teknologi Malaysia (UTM) for providing the equipment and technical support throughout the research was being carried out.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Minsky, M. Telepresence. *OMNI Magazine*, June 1980; pp. 44–52.
2. Steuer, J. Defining Virtual Reality: Dimensions Determining Telepresence. *J. Commun.* **1992**, *42*, 73–93. [CrossRef]
3. Shih, C.F. Telepresence and bricolage: A conceptual model of consumer experiences in virtual environments. In *1998 Winter Society for Consumer Psychology Conference Proceedings*; Society for Consumer Psychology: Columbus, OH, USA, 1998; Volume 231.
4. Tuhovčák, J.; Hejčík, J.; Jícha, M. A Review of Mixed Reality Telepresence. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; Volume 864, p. 012081. [CrossRef]
5. Yu, K.; Gorbachev, G.; Eck, U.; Pankratz, F.; Navab, N.; Roth, D. Avatars for Teleconsultation: Effects of Avatar Embodiment Techniques on User Perception in 3D Asymmetric Telepresence. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 4129–4139. [CrossRef] [PubMed]
6. Kolkmeier, J.; Harmsen, E.; Giesselink, S.; Reidsma, D.; Theune, M.; Heylen, D. With a little help from a holographic friend: The openimpress mixed reality telepresence toolkit for remote collaboration systems. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, Tokyo, Japan, 28 November–1 December 2018; pp. 1–11.
7. Zhang, Y.; Yang, J.; Liu, Z.; Wang, R.; Chen, G.; Tong, X.; Guo, B. VirtualCube: An Immersive 3D Video Communication System. *arXiv* **2021**, arXiv:2112.06730. [CrossRef]
8. Tonchev, K.; Bozhilov, I.; Petkova, R.; Poulkov, V.; Manolova, A.; Lindgren, P. Implementation Requirements and System Architecture for Mixed Reality Telepresence Application Scenario. In *Proceedings of the 24th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Okayama, Japan, 14–16 December 2021; pp. 1–6. [CrossRef]
9. Yoon, L.; Yang, D.; Chung, C.; Lee, S.H. A Full Body Avatar-Based Telepresence System for Dissimilar Spaces. *arXiv* **2021**, arXiv:2103.04380.
10. Fadzli, F.E.; Ismail, A.W. A Robust Real-Time 3D Reconstruction Method for Mixed Reality Telepresence. *Int. J. Innov. Comput.* **2020**, *10*, 15–20. [CrossRef]
11. Pejša, T.; Kantor, J.; Benko, H.; Ofek, E.; Wilson, A. Room2room: Enabling life-size telepresence in a projected augmented reality environment. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, San Francisco, CA, USA, 27 February–2 March 2016; pp. 1716–1725.
12. Fuchs, H.; State, A.; Bazin, J.-C. Immersive 3D Telepresence. *Computer* **2014**, *47*, 46–52. [CrossRef]
13. Córdova-Esparza, D.-M.; Terven, J.R.; Jiménez-Hernández, H.; Herrera-Navarro, A.; Vázquez-Cervantes, A.; García-Huerta, J.-M. Low-bandwidth 3D visual telepresence system. *arXiv* **2018**, arXiv:1804.02343. [CrossRef]
14. Ali, A.Z.M.; Ramlie, M.K. Examining the user experience of learning with a hologram tutor in the form of a 3D cartoon character. *Educ. Inf. Technol.* **2021**, *26*, 6123–6141. [CrossRef]
15. Yu, K.; Eck, U.; Pankratz, F.; Lazarovici, M.; Wilhelm, D.; Navab, N. Duplicated Reality for Co-located Augmented Reality Collaboration. *IEEE Trans. Vis. Comput. Graph.* **2022**, *1*. [CrossRef] [PubMed]
16. Haleem, W.M.A.; Arous, S.; Amer, T. Potentials Benefits of Applying Three Dimensional Hologram Technology (3DHT) in The Hotel Industry. *J. Fac. Tour. Hotels-Univ. Sadat City* **2021**, *18*.
17. Andrade, M.A.R. Holographic Reality: Enhancing the Artificial Reality Experience through Interactive 3D Holography. Ph.D. Thesis, Universidade da Madeira, Funchal, Portugal, 2021.
18. Orts-Escalano, S.; Rhemann, C.; Fanello, S.; Chang, W.; Kowdle, A.; Degtyarev, Y.; Izadi, S. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, Tokyo, Japan, 16 October 2016; pp. 741–754.
19. Kinect for Windows. Kinect-Windows App Development. Available online: <https://developer.microsoft.com/en-us/windows/kinect/> (accessed on 22 February 2022).
20. Zollhöfer, M.; Stotko, P.; Görlitz, A.; Theobalt, C.; Nießner, M.; Klein, R.; Kolb, A. State of the art on 3D reconstruction with RGB-D cameras. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2018; Volume 37, pp. 625–652.
21. Du, H.Y.; Miao, Z.J. Kinect depth maps preprocessing based on RGB-D data clustering and bilateral filtering. In *Proceedings of the Chinese Automation Congress (CAC)*, Wuhan, China, 27–29 November 2015; pp. 732–736.
22. Yang, Q. Recursive bilateral filtering. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 399–413. [CrossRef]
23. Khoshelham, K. Accuracy analysis of kinect depth data. In *Proceedings of the ISPRS Workshop Laser Scanning*, Calgary, AB, Canada, 29–31 August 2011; Volume 38. [CrossRef]
24. Kowalski, M.; Naruniec, J.; Daniluk, M. Livescan3d: A fast and inexpensive 3d data acquisition system for multiple kinect v2 sensors. In *Proceedings of the 2015 International Conference on 3D Vision*, Lyon, France, 19–22 October 2015; pp. 318–325.

25. Custodio, L.; Pesco, S.; Silva, C. An extended triangulation to the Marching Cubes 33 algorithm. *J. Braz. Comput. Soc.* **2019**, *25*, 6. [\[CrossRef\]](#)
26. Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W. Surface reconstruction from unorganized points. In Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, Chicago, IL, USA, 1 July 1992; pp. 71–78. [\[CrossRef\]](#)
27. Stotko, P.; Krumpen, S.; Hullin, M.B.; Weinmann, M.; Klein, R. SLAMCast: Large-scale, real-time 3D reconstruction and streaming for immersive multi-client live telepresence. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 2102–2112. [\[CrossRef\]](#) [\[PubMed\]](#)
28. A Jeffreys, H.; Jeffreys, B. Central differences formula. In *Methods of Mathematical Physics*; Cambridge University Press: Cambridge, UK, 1988; pp. 284–286.
29. Wiemann, T.; Annuth, H.; Lingemann, K.; Hertzberg, J. An Extended Evaluation of Open Source Surface Reconstruction Software for Robotic Applications. *J. Intell. Robot. Syst.* **2014**, *77*, 149–170. [\[CrossRef\]](#)
30. Alexiadis, D.S.; Zarpalas, D.; Daras, P. Real-time, realistic full-body 3D reconstruction and texture mapping from multiple Kinects. In Proceedings of the IVMS 2013, Seoul, Korea, 10–12 June 2013; pp. 1–4. [\[CrossRef\]](#)
31. Waldspühl, J.; Zhang, E.; Butyaev, A.; Nazarova, E.; Cyr, Y. Storage, visualization, and navigation of 3D genomics data. *Methods* **2018**, *142*, 74–80. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Oh, K.J.; Kwon, S.K. Real-time Implementation of Character Movement by Floating Hologram based on Depth Video. *J. Multimed. Inf. Syst.* **2017**, *4*, 289–294.
33. Yang, L.; Dong, H.; Alelaiwi, A.; El Saddik, A. See in 3D: State of the art of 3D display technologies. *Multimed. Tools Appl.* **2015**, *75*, 17121–17155. [\[CrossRef\]](#)
34. Wu, C. Inverse Rendering for Scene Reconstruction in General Environments. Ph.D. Thesis, Universität des Saarlandes, Saarbrücken, Germany, 2014. [\[CrossRef\]](#)