International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2023)
J.-M. Normand, M. Sugimoto, and V. Sundstedt (Editors)

3D Reconstruction for Tele-Immersion in 360° Live Stream

Clément Dluzniewski[1,2], Hakim Chekirou[1], Jérémie Le Garrec[1], Claude Andriot[1], and Frédéric Noël[2]

[1]Université Paris-Saclay, CEA, List, F-91120, Palaiseau
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000, Grenoble, France

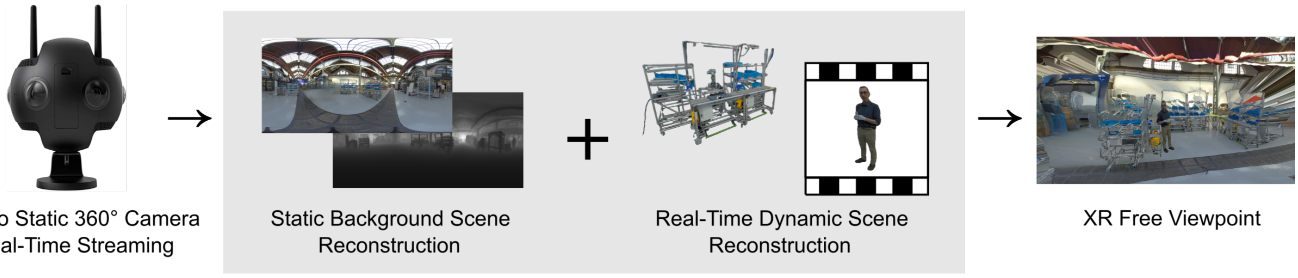

Figure 1: *Proposed approach to reconstruct a dynamic scene from a single static 360° camera. Three type of elements are considered: environment, object of interest and people. The environment, static, is reconstructed in an initialization step (section 3). The dynamics elements, object of interest and people, are reconstructed in real-time with different 3D representation (section 4). Combining these static and dynamic reconstructions, XR user can freely navigate in the 360° video.*

Abstract

Nowadays, most volumetric tele-immersion systems are based on a multi-camera system to capture a dynamic 3D place. With these acquisition devices, the development of a mobile tele-immersion system seems compromised, as a lot of equipment would have to be moved. One promising way to achieve a mobile system would be to use a single 360° camera and develop ways of reconstructing in 3D a dynamic scene in real time from a single point of view. Therefore, we propose an approach to freely navigate into a 360° video captured with a static camera. The approach considers three types of elements in the scene, the environment, the object of interest and the people, and relies on a different 3D representation for each type of element. Distinguishing the scene elements enables a real-time method to be adopted, by reconstructing static elements once and using fast-computable 3D representations for dynamic elements. As the method is real-time, we develop a streaming pipeline to enable XR users to move live within the camera stream.

CCS Concepts

• ***Computing methodologies*** → ***Virtual reality;*** *Reconstruction;*

1. Introduction

Tele-immersion aims to immerse one or more people, at different locations, in a new place either virtual or real [Ohl18]. While a virtual place is completely modeled by computer, 3D reconstruction methods along with acquisition devices must be used to virtualize a real place. These acquisition devices, mostly based on several cameras, are classified in two categories: outside-in and inside-out [RTW20, LTT15]. In both systems, camera positions can be approximated as belonging to a circle, but in the outside-in configuration, the cameras face the center of the circle, while in the inside-out configuration, the cameras face the outside of the circle. As noted in [RTW20], this difference implies that outside-in devices are better suited for capturing objects, while inside-out devices are better suited for capturing environments. Then, outside-in devices offer the possibility to tele-immerse objects of interest of the scene in 3D, such as human avatars [OERF*16], but they are generally not mobile because the cameras of the device must be placed all around the room. On the other hand, inside-out devices are easily transportable because they are composed of a single device such as a 360° camera, but they do not enable the complete reconstruction of independent 3D elements. This disadvantage of the 360° camera is detrimental for the creation of virtual reality scenes because it implies that one cannot have a rendering of an arbitrary point of view of the scene, making free navigation impos-

DOI: 10.2312/egve.20231325

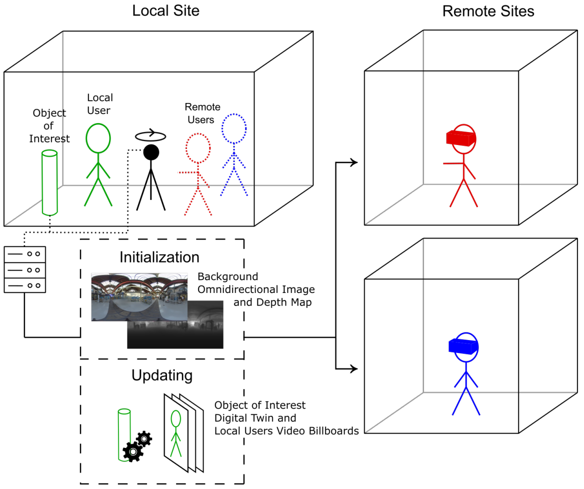
Local Site
Remote Sites
Object of Interest
Local User
Remote Users
Initialization
Background Omnidirectional Image and Depth Map
Updating
Object of Interest Digital Twin and Local Users Video Billboards

Figure 2: *Overview of our approach. Remote users connect to the server to freely navigate in the streamed omnidirectional video.*

sible. An interesting research problem is then to determine how to create virtual reality scenes with free navigation from an inside-out acquisition device.

In this paper, we propose an approach to create a tele-immersion application based on a single static 360° camera for immersing remote users live into a dynamic 3D reconstruction. The single-camera setup provides an easy-to-use mobile tele-immersion system without the need for preprocessing for use cases such as remote learning, immersive meetings or remote assistance. To achieve this goal, we developped a method to reconstruct a free-viewpoint virtual environment from an omnidirectional stream, relying on the assumption that the scene is composed of a static background and dynamic foreground elements (Figure 1). In our context, the foreground represents an object of interest, which is the central element of the scene, with surrounding people. An overview of the approach is shown Figure 2. Using parallax generation, registration techniques and video tracking, we achieve a representation in which users can move freely around the camera position. The strength of our approach is that the 3D representation is obtained in real time and transmitted over the network with limited bandwidth, enabling users to move around inside the 360° video in live.

2. Related Works

2.1. 360° based Tele-Immersion

To implement a mobile tele-immersion system, we propose to use a single static omnidirectional camera. An existing proposition based on a static omnidirectional camera is AVT [RTM*20]. AVT is a system in which a host user brings a traveler user into his environment captured by a single static omnidirectional camera. The traveler receives a stream of the video filmed by the camera in real-time and can interact with the host. Because no additional 3D information is captured, the traveler is fixed to the camera position, so there can only be one remote user at the same time. To enable a remote user to move around the environment, some authors have proposed to mount the omnidirectional camera on a mobile base, as in VROOM [JZWR20] or Holobot [KKK*23]. In this case, the camera position is controlled by the remote user, who can choose its position in the environment by physically moving the camera. The problem with this approach is that it requires one omnidirectional camera per remote user, which limits the number of participants. A solution for bringing several remote users into the scene is to introduce 3D in addition to video. [TLL*19] proposed to combine an omnidirectional video stream with a 3D reconstruction. A user can switch from a real-time 360° stream recorded in first-person view to a 3D reconstruction in which they can move freely. However, they assume that the scene is static, which makes their system unusable if there are dynamic elements. In addition, the use of a single moving camera does not allow for a fully dynamic scene (regions not directly visible to the camera cannot be updated).

We propose a system inspired by AVT, based on static omnidirectional camera, with the difference that we want to generate a large motion parallax in real-time so that several remote users can move freely in the environment.

2.2. Parallax Generation

Achieving a 6-DoF representation from a single static omnidirectional camera is still a challenging problem. Most of the proposed methods and representations aims to create 3-DoF+ experience for producing head-motion parallax effects when viewing the 360° video with a HMD, but not a full 6-DoF scene. The basic principle for creating a 6-DoF experience from a single static camera is to recover the geometry to render the scene at any arbitrary point of view. An omnidirectional image being the projection of the inner surface of a textured three-dimensional sphere seen with a camera virtually placed at its center, the addition of geometry enables to deform this sphere into a more complex shape closer to reality. A commonly studied geometric representation added to the omnidirectional image is the omnidirectional depth map. An omnidirectional depth map is a grayscale image where one pixel represents the distance from that particular point on the sphere surface to the sphere center. The development of spherical networks, taking as input a 360° image, allows estimating the geometry thanks to two types of approaches: direct estimation of the depth map and layout estimation [dSPMLJ22]. The first type of method estimates the depth map directly from the 360° image. The authors proposed neural networks producing an omnidirectional depth map, taking as input the equirectangular projection of the image [ZKZD18, SSC21], a perspective images slicing of the 360° image [RAYR22, LGY*22] to exploit already existing monocular depth estimation architecture, or both equirectangular and perspectives projections [WYS*20, JSZ*21]. On the other hand, layout estimation methods recover the captured room layout which can be seen as a simplified depth map approximating the floor, ceiling and walls by planes. Many layout estimation methods are based on the Manhattan hypothesis [SHSC19, YWP*19, WYS*21], which assumes that the scene is a room with a flat floor, ceiling, and walls, and that the walls intersect at right angles. The disadvantage of these approaches is that they produce depth maps where objects are flattened against walls, floors and ceilings, giving less realistic virtual reality scenes. Alternative approaches have been proposed based on more complex inputs than just a 360° image to exploit the properties of multi-camera devices omnidirectional cameras. [MJJK21] used a 4 fisheyes omnidirectional camera and proposed to compute the depth map in real time with fisheye stereo. A widely used input type is omnidirectional stereo (ODS) [PBEP01]. The ODS is a 360° left-right stereo pair that can be conveniently obtained with a multi-camera device. In [LXLL19] a network is proposed taking as input an ODS image and outputting the depth map in real-time. With this additional geometric information, the 360° image is rendered by creating a mesh from its associated depth map and projecting the image onto the mesh. However, this rendering approach is not the most suitable when it comes to create full 6-DoF scene. Indeed, an omnidirectional depth map only encodes a particular type of sphere deformations following a topology named star [MS19]. Representing a scene with an omnidirectional depth map will be acceptable if the scene can be viewed as a single connected component without independent elements like a simple empty room. If this is not the case, phantom surfaces will appear between non-connected component, in particular between foreground and background elements. Even if the presence of these artifacts may not be disturbing for 6-DoF users [DDDP18], more complex topologies may be modeled with layered representations. A layered representation is a set of semi-transparent images of the scene from the same point of view, with or without explicit geometry [RTW20]. Because the layered representation contains information about what is hidden behind the occlusions, it enables the modeling of more complex topologies than a star domain. When the geometry is modeled implicitly by the layer, such that if one layer is in front of another, it implies that this layer is closer to the camera, the representation corresponds to Multi-Plane Image (MPI) [TS20, HWY22]. Some authors have adapted the MPI representation to work with omnidirectional image. The Multi-Sphere Image (MSI) is introduced in [ALG*20]. The MSI representation corresponds to a set of concentric semi-transparent spherical layers centered on the camera position. They rely on a neural network inferring the MSI from a pair of ODS images in real-time. Similarly, [WGD*22] proposed Multi-Cylinder Image (MCI), a set of cylindrical layers but computed from a single panoramic image. Even if this representation enables a large motion parallax, layer artifacts quickly appear when the user moves away from the center of the MSI, making this representation impractical for tele-immersion with free navigation. A strategy to achieve a better 6-DoF with large motion is to explicitly model the geometry with a depth map per layer, leading to the Layered Depth Image (LDI) representation [SGHS98, SSKH20, KMA*20]. Contrary to MPI, the different layers do not implicitly represent depth information but rather topological information: if two elements are on two different layers, then they are independent. A common approach is to use a spherical LDI with two layers to model the foreground and background elements [HASK17]. In [SKC*19, MKK*20] a 3 layers structure is proposed : a dynamic foreground layer, an extrapolated layer for static background regions occluded by moving objects and an inpainted layer to fill-in regions occluded by static objects. [LXM*20] proposed Multi Depth Panorama (MDP) a panoramic LDI with a arbitrary number of layers, thanks to a 3D CNN predicting an MPI.

To generate a 6-DoF scene, we propose a representation inspired by a two-layers omnidirectional LDI, but instead of using a foreground layer, foreground elements are modeled with registered 3D models and billboards.

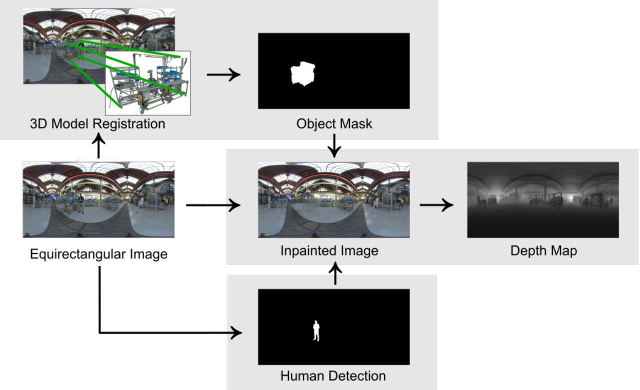

Figure 3: *Static scene reconstruction. Information about the object of interest (upper box) and about the people (lower box) are required to remove foreground from the environment (middle box).*

3. Static Scene Reconstruction

Since we want a system in which users can navigate freely in real time, no existing approach is completely satisfying. To achieve this goal, one solution is to reconstruct separately the different type of elements that compose the scene. The first element is the environment, corresponding to the background of the input image. The environment is modeled by an omnidirectional image associated with a depth map to add relief. To keep the interactive times, we assume the environment is static, thus avoiding the reconstruction of the depth map at each frame. The second element is the object of interest, corresponding to the central element of the scene, for example a machine to repair in a remote assistance scenario. The last elements are the people co-located with the camera whose avatars need to be represented. To initialize this scene, a static reconstruction is performed on the first frame of the video stream, consisting in two main steps : registration of the object of interest and creation of the environment. The static scene reconstruction is summarized Figure 3.

3.1. Object of Interest Registration

The representation of the geometry of a scene by a depth map implies that its topology is a star-domain [MS19]. Under this topology, only one connected component is possible to model, which makes the creation of a scene with independent 3D elements impossible. To solve this issue, our proposal consists of representing the foreground elements with a 3D model, automatically aligned in the 360° video.

[ZLM*22] proposed an algorithm to align a 3D model on an equirectangular image, based on the analysis of the silhouette. However, their method is not generic and a detector needs to be trained to segment the element of interest in the video. We propose

Proceedings published by Eurographics - The European Association for Computer Graphics.

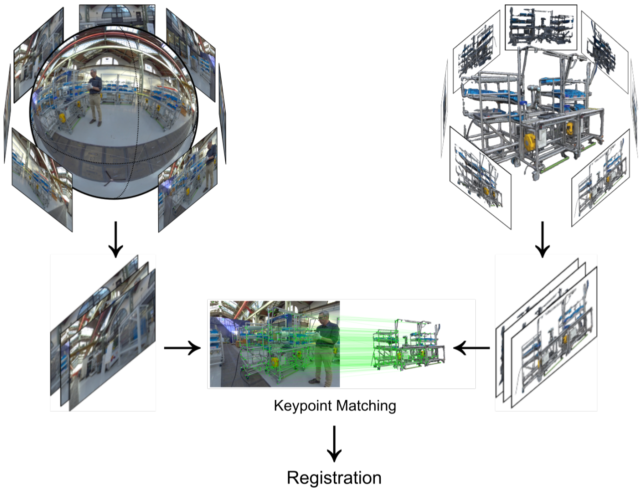

Figure 4: *Object registration. Perspectives images of the 360° image and the 3D model are compared to find the best match. Given the pair that maximizes the match, the pose of the camera is estimated and the position of the 3D model is derived.*

to use a marker based and model based approach based on superpoint [DMR18] and superglue [SDMR20] to find the alignment of a textured object in a equirectangular image. The algorithm is summarized Figure 4. Estimating the position of the object is equivalent to solving a camera localization problem using Perspective-n-Point (PnP). First, two perspective cameras C_{360} and C_{model} must be determined, the first covering the object of interest in the 360 image and the second covering the same elements of the 3D model as C_{360}. Note that the approach supposes only one instance of the object of interest is present in the scene. N perspective images in the 360° image are sampled, and M images are rendered around the 3D model. The superpoint-superglue is run on the $N \times M$ pairs of images, and the pair that maximizes the matches is kept with their camera positions. Superglue is used as it was more robust to texture alteration between the rendering of the 3D model and the 360° image than other methods in our experiments To get the 3D points, the 2D markers produced by superpoint-superglue from the rendered image are projected onto the 3D model by ray-tracing. The PnP problem is solved using RANSAC between the 2D points from C_{360} and the 3D points. The PnP algorithm is initialized with the position of C_{model} and the intrinsic parameters are set to K_{360}. The resulting camera C^*_{model} is the camera that covers the model in the same way as C_{360} covers the object of interest. Lastly, the model is positioned by computing the relative translation and rotation $[R,t]$ from C^*_{model} to C_{360}. The model is then translated by t and rotated around C_{360} by R.

3.2. Environment Creation

The environment is a particular 3D object that defines the limits of the scene in which the user navigates. Omnidirectional images are suitable for modeling the environment, but in our case the image also contains foreground elements. Therefore, the first step to create the environment is to remove the foreground elements, the object of interest and the people, from the 360° image. A solution is to obtain a mask of these elements and use an inpainting algorithm. With the registration of the 3D model, a segmentation of the object of interest in the equirectangular image is easily available. To get a mask of the people, an object detection method was used. Detecting objects in omnidirectional image is studied [YQC*18] but [Fas19] observed that YOLO detector is robust to equirectangular distortions if the elements are not near to either the poles or to the boundaries. With a static camera properly aligned, humans in the equirectangular image respect this constraint. Then, YOLO was used to generate the people masks. Given the first frame of the video with these masks, the foreground elements were removed with the inpainting algorithm. The most convincing results were obtained with LaMa [SLM*22].

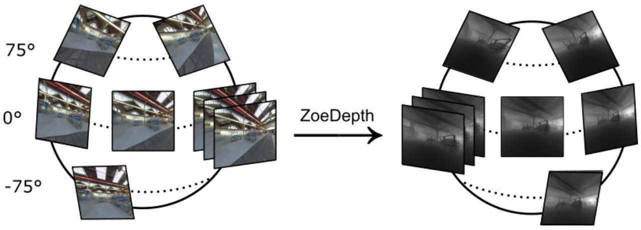

Figure 5: *Omnidirectional depth estimation. The input equirectangular image is decomposed into perspective images. Monocular depth estimation is performed on each perspective image. The resulting depth maps are projected back onto an equirectangular image.*

To add the geometry, an omnidirectional depth map was associated to the inpainted image. The depth map was computed by decomposing the omnidirectional image into a set of perspective images to exploit the existing perspective monocular depth estimation network. The existing depth estimation methods did not produce depth maps of a suitable quality for virtual reality, the depths were not detailed enough with too many artifacts. To address these issues, a similar approach to the 360MonoDepth [RAYR22] was adopted, but the perspective sampling, depth estimation model, and depth fusion methods were modified. In particular, Midas [RLHK20] was replaced by ZoeDepth [BBW*23], a monocular perspective depth estimation model that predicts metric depth instead of inverse relative depth. Consistency between different perspectives and depth fusion are greatly improved, as problems of scale between different depth maps are avoided.

The proposed algorithm is illustrated Figure 5. First, overlapping tangent perspective images were extracted in a dense manner to cover the entirety of the sphere. We use gnomonic projection [LGY*22] to project 60, 15 and 15 images at latitudes of 0°, +75° and -75° respectively. The field of view is the same for all images and is set to 90° to maximize coverage and reduce distortion. In the second step, ZoeDepth was run on all images and the depth maps were projected into a global equirectangular depth map using the inverse gnomonic transformation. Overlapping depth values are averaged across all the depth maps. Since every pixel is covered by multiple images, the resulting equirectangular depth map is consistent with low distortions and artifacts. One of the advantages of

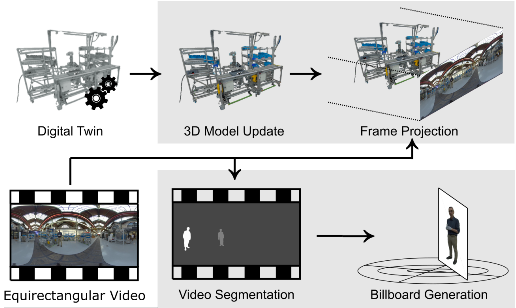

Figure 6: *Dynamic scene reconstruction. The object of interest (upper box) and the people (lower box) are intialized with information from the static scene reconstruction.*

this algorithm is that it seems sufficiently robust to make coherent depth estimation in inpainted regions. The background image with the corresponding depth map are transmitted to the clients at the initialization to render the environment.

4. Dynamic Scene Reconstruction

To have a complete 3D scene reconstruction, dynamic elements must be added to the static representation of the scene. Dynamic elements must be tracked and positioned in the scene to achieve live reconstruction. The live constraint has guided us towards representations that can be obtained fast. The first step in the dynamic reconstruction is to animate the object of interest by enhancing it with a digital twin. The second step is to create a billboard for each person in the video to represent their avatar. The dynamic scene reconstruction is illustrated Figure 6.

4.1. Object of Interest Updating

Because the described registration algorithm is too long to run in real-time, it can not be used at each frame for dynamic object. A solution may be to track in 6-DoF the dynamic object of interest along the video, but this approach relies on heavy computations. Our proposition to avoid tracking is to animate the object of interest by enhancing the 3D model with a digital twin. The digital twin captures the kinematics of the real object of interest and sends them to the user, changing the pose of the 3D model in real time. For example, if the object of interest is a cobot, the joints articular positions and velocities are transmitted to animate the 3D model along the video.

An issue is that users may easily notice a 3D model is integrated in the scene since it may lack of photorealism (simple texture, no lighting model, ...). To have a coherent rendering of the model, the current frame of the video is used to texture the object of interest. The pixels of the video aligned on the 3D model are projected on it and update a global texture of the object of interest. As the object of interest moves, thanks to the digital twin, new regions initially occulted become visible, the global texture of the 3D model is then updated. If a region is never seen by the camera, the corresponding pixels in the global texture is set to a default color.

4.2. Human Avatars

Reconstruction of a 3D avatar from a single point of view without geometric information is an ambiguous problem. To our best knowledge, only Monoport [LOX*20] proposes to create a 3D human avatar in real-time from a single RGB stream. Their algorithm can recover regions of the avatar that are not directly seen by the camera, but is based on PIFu [SHN*19] which is prone to reconstruction errors that may degrade the user sense of presence. Our proposal is to represent people with billboards, 2D plans positioned in 3D space on which videos are projected. Experiments have shown this representation was more comfortable for users than a noisy reconstruction [DMC*22], or equivalent to a 3D avatar if the user remained facing to the billboard [CKL*20]. Using billboards with the background depth map is equivalent to using a layered depth image with a flat depth map for the foreground. We chose to not calculate a depth map on the foreground because we assume that adding relief to the billboard is marginal for avatar realism. The goal of our pipeline is then to generate flat billboards of dynamic elements only from the omnidirectional video stream. The use of billboards enables the generation of 3D representations of objects in the video generically (the representation can be created for any objects as long as they can be detected at initialization) and in real time. No particular human keypoints needs to be tracked, a 360° video segmentation is enough to cut out a person from the video and insert the corresponding billboard in the 3D space.

The first step is to perform the video segmentation of the whole equirectangular video to isolate people from other elements. The advantages of processing the whole equirectangular image is that with the video segmentation all the human billboards are created in one pass, instead of segmenting individual patches for each person. The video segmentation is initialized with the resulting mask of the human detection performed in the static scene reconstruction. The video segmentation deform these masks throughout the video to fit the boundaries of the tracked elements. Since video segmentation can be time-consuming, the mask updating in our pipeline is done on a reduced resolution frame to be processed in real time. After being predicted on the reduced image, the mask is rescaled to the original image resolution with bicubic interpolation. To pass from the equirectangular mask to a perspective image centered on the tracked person, a gnomonic projection is performed for each tracked person on the frame of the video. The parameters of an individual projection are simply retrieved thanks to a bounding box containing the mask. The center of projection (λ, ϕ) and the field of view f are then computed with the following equations :

$$\lambda = 2\pi(\frac{x_{bb}}{w} - \frac{1}{2})$$
$$\phi = \pi(\frac{y_{bb}}{h} - \frac{1}{2})$$
$$f = \max(2\pi\frac{w_{bb}}{w}, \pi\frac{h_{bb}}{h})$$

where (x_{bb}, y_{bb}) is the center of the bounding box of the mask and (w_{bb}, h_{bb}) is the size of the bounding box. The projection is applied by using the same field of view horizontally and vertically, resulting in a square perspective image. Therefore, f is the maximum between the horizontal field of view f_h and the vertical field of view f_v. Before performing the gnomonic projection, the input equirectangular image is masked using the predicted person segmentation, and the pixels outside the mask are replaced by a default color.

The last step to render a person as a billboard is to get their pose in the 3D space. Since our billboards are world-aligned (always facing the center of the scene), only the position and the scale are needed. The center of the billboard (x,y,z) and the scale s are estimated with the following formulas :

$$\begin{aligned} x &= \rho \sin\phi \cos\lambda \\ y &= \rho \tan\frac{f_v}{2} \\ z &= \rho \sin\phi \sin\lambda \\ s &= \rho \tan\frac{f}{2} \end{aligned}$$

where ρ is the distance between the billboard and the camera. This distance is computed with the formula proposed by [MLPALC21]. The formula assumes that the object whose distance is being estimated touches the ground, the floor is flat, and the camera height is known, assumptions valid in our case. To apply this formula, we suppose the contact point between a person and the floor is the pixel of the equirectangular mask with the lowest value in ordinate. Video billboards with calculated parameters are streamed to clients to render people in the environment. The billboards are rendered as a 3D object with chroma keying to make the background transparent.

5. Technical Evaluation

The goal of the evaluation is to search for a configuration usable for practical tasks of the system, meaning interactive times and great visual quality. To ensure that our system is able of interactive times, the frame rates were measured according to the resolution of the video and the number of people tracked. An important factor for the final rendering is the quality of the people contouring, the quality of the video segmentation was also evaluated according to the resolution of the video and the distance to the camera.

For the implementation, we used YOLOv8 to detect the people in the first frame of the omnidirectional video. We used XMem [CS22] to segment the video, which we modified to take as input an image of reduced resolution to speeding up the segmentation, and output a mask of the initial resolution with a bicubic interpolation. We also used CuPy to perform all possible image computation on GPU. The computations were performed on a server with a NVIDIA GeForce RTX 3090 GPU. The clients, connected to the server, render the scene in real-time using Unity.

5.1. Real-Time Evaluation

To check if the dynamic elements can be rendered in real time, the duration of the entire billboard generation pipeline was measured, from the video segmentation to determining the 3D position. Three video input resolution were tested, low (400 × 200), medium (800 × 400) and high (1600 × 800) to determine the maximum achievable resolution in real time. The change in resolution have no effect on the image quality but affect the quality of the person contouring. Also, one to three people were tracked to determine if the system is fast enough to process several people. The resulting frame rates are provided in Table 1.

Table 1: *Average frame rates based on equirectangular input resolution and number of tracked people.*

Resolution		Frame rate
Low (400 × 200)	1 person	38.4 fps
	2 people	23.9 fps
	3 people	17.6 fps
Medium (800 × 400)	1 person	30.5 fps
	2 people	18.9 fps
	3 people	13.8 fps
High (1600 × 800)	1 person	11.2 fps
	2 people	7.8 fps
	3 people	5.9 fps

From these results, we observe that the system interactivity is suitable only in low resolution and medium with a single person. The additional cost when tracking several people is mostly due to the fact the billboards positioning in the 3D space is performed sequentially for each individual person. Efforts to implement parallelization could result in reducing the frame rates when the number of people is higher than 1. Also, no effect of the tested output billboard resolution, from low-resolution to high-resolution, have been observed on the frame rate.

To ensure the proposed representation can be streamed over the network, we also evaluate the used bandwidth of our approach. Most of the scene elements were sent to a client on the local network, the environment was transmitted in HTTP, the billboard videos in RTSP and other useful information in UDP. The single-person billboard video resolution was 800 × 800 pixels and if more than one person was present in the scene, their billboard images were stacked into a larger common image to use only one RTSP stream. For comparison, the original 360° video used to generate the scene was also streamed using RTSP with an equirectangular projection of resolution 2048 × 1024 pixels. We found the bandwidth usage of our approach was close to the 360° video except at the start where our approach receives a larger amount of data to obtain the static 3D reconstruction. The observed latency was around 1.5 seconds, which seems acceptable for interaction between local and remote people. These results confirm our assumption that our approach enables a live free navigation with a network usage equivalent to videoconferencing.

5.2. Billboard Quality Evaluation

This experiment aims to determine the evolution of the quality of the contouring of the people depending on the distance to the camera to provide information on the good practices with this system. We hypothesize that the higher is the resolution, the higher is the

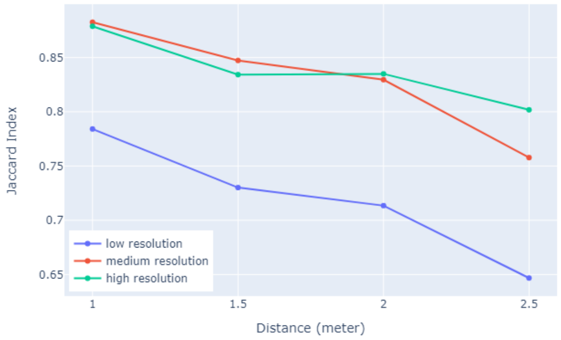

Figure 7: *Quality of a person contour depending on the distance to the camera and the input resolution.*

quality of the contouring, and that the quality of the contouring decreases when the distance to the camera increases.

To evaluate the contouring quality, we first shot some videos with a person at different distances from the camera (1 m, 1.5 m, 2 m and 2.5 m). This person performs a number of different poses that are the same in all the videos. In these videos, some key frames were selected and used to manually create reference masks. These reference masks, used as ground truth, were obtained with an interactive segmentation tool [SPK22] by selecting the person on the equirectangular image. Then, the video segmentation was performed on the videos, taking as input the low, medium and high resolutions. The result of the video segmentation with the different resolutions at these key frames were saved and compared to the reference masks using the Jaccard index (Intersection over Union). The average of the Jaccard indexes of the key frames for each distance is computed to obtain the graph Figure 7. Based on these results, we observe that the contouring quality degrades as the distance from the camera increases, thus confirming our hypothesis. Note that the values of the Jaccard indexes are high because the image has a resolution of 2048 × 1024 while the segmentation of the person occupies only a small part of this image. However, the metric seems to indicate that the medium and high resolutions have roughly the same contour quality while that of the low resolution is well below. This observation suggests that a medium resolution can be used for performance reasons (Table 1) without loss of quality. We also notice, from the resulting mask, that all resolutions tend to lose track of arms and hands after 1.5 m, which can be detrimental to non-verbal communication. Experimenting with users to assess the impact of billboard quality on subjective metrics would be interesting.

From these experiments, we can conclude that to have an interactive use of our system with a pleasant billboard representation, the best configuration is to use a medium resolution (able to reach real time performances) with a single person positioned between 1 m and 1.5 m from the camera.

6. Limitations and Openings

Today the system is limited and potential improvements are identified. Because the technologies we use are designed to work on perspective image, our system shows limitations when spherical properties of equirectangular images needs to be exploited. For example, a person going from the left border of the image to the right (or vice versa) may not be properly tracked. The lack of 3D for the avatar will be an issue if the user is not facing the on-site user. A solution can be to use 4D reconstruction to build the avatar along the video Another issue is to propagate the texture of the aligned 3D model from visible area to occluded area to have a fully textured by the 360° image. Solving this issue would help to transparently integrate the 3D model to the picture and let the user freely moves around the object of interest without discomfort. Finally, using layered approaches to create a mesh from a single depth map without stretched surfaces will be beneficial to create more realistic background.

7. Conclusion

We presented an approach to capture a 3D scene in real time, from a single static omnidirectional camera, in which users can move. The approach considers three different types of elements (environment, object of interest and people) and proposes a different 3D representation for each of these elements. The static environment is reconstructed from a 360° image on which foreground elements are removed, combined with a depth map. The object of interest is obtained by aligning a 3D model, known beforehand, on the 360° video and animating it with a digital twin. Finally, people are represented by billboards, positioned in the 3D space.

This proposition is an interesting technical foundation for a mobile tele-immersion system because of the mobility of the acquisition device and the used bandwidth for streaming, comparable to 2D video. For the moment, the system lacks interaction between on-site and remote people, or between different remote people. The perspective would be to integrate remote people avatars into the 3D reconstruction, so that different users are aware of each other presence. Future user evaluations will be conducted to review the suitability of the approach for concrete tasks implied in collaborative tele-immersion.

References

[ALG*20] Attal B., Ling S., Gokaslan A., Richardt C., Tompkin J.: MatryODShka: Real-time 6DoF Video View Synthesis using Multi-Sphere Images. In *Computer Vision – ECCV 2020* (2020), Lecture Notes in Computer Science, Springer International Publishing, p. 441–459. doi:10.1007/978-3-030-58452-8_26. 3

[BBW*23] Bhat S. F., Birkl R., Wofk D., Wonka P., Müller M.: ZoeDepth: Zero-shot Transfer by combining Relative and Metric Depth. *arXiv* (2023). doi:10.48550/ARXIV.2302.12288. 4

[CKL*20] Cho S., Kim S.-w., Lee J., Ahn J., Han J.: Effects of volumetric capture avatars on social presence in immersive virtual environments. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2020), p. 26–34. doi:10.1109/VR46266.2020.00020. 5

[CS22] Cheng H. K., Schwing A. G.: XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model. In *Computer Vision – ECCV 2022* (2022), Springer Nature Switzerland, p. 640–658. 6

Proceedings published by Eurographics - The European Association for Computer Graphics.

[DDDP18] Dupont De Dinechin G., Paljic A.: Cinematic Virtual Reality With Motion Parallax From a Single Monoscopic Omnidirectional Image. In *2018 3rd Digital Heritage International Congress (DigitalHERITAGE) held jointly with 2018 24th International Conference on Virtual Systems Multimedia (VSMM 2018)* (2018), p. 1–8. doi:10.1109/DigitalHeritage.2018.8810116. 3

[DMC*22] Debarba H. G., Montagud M., Chagué S., Herrero J. G.-L., Lacosta I., Langa S. F., Charbonnier C.: Content format and quality of experience in virtual reality. *Multimedia Tools and Applications* (2022). doi:10.1007/s11042-022-12176-9. 5

[DMR18] DeTone D., Malisiewicz T., Rabinovich A.: SuperPoint: Self-Supervised Interest Point Detection and Description. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), IEEE, p. 337–33712. doi:10.1109/CVPRW.2018.00060. 4

[dSPMLJ22] da Silveira T. L. T., Pinto P. G. L., Murrugarra-Llerena J., Jung C. R.: 3D Scene Geometry Estimation from 360° Imagery: A Survey. *ACM Computing Surveys* (2022). doi:10.1145/3519021. 2

[Fas19] Fassold H.: Adapting Computer Vision Algorithms for Omnidirectional Video. *ACM Multimedia* (2019). doi:10.1145/3343031.3350579. 4

[HASK17] Hedman P., Alsisan S., Szeliski R., Kopf J.: Casual 3D photography. *ACM Trans. Graph.* (2017). doi:10.1145/3130800.3130828. 3

[HWY22] Han Y., Wang R., Yang J.: Single-View View Synthesis in the Wild with Learned Adaptive Multiplane Images. In *ACM SIGGRAPH 2022 Conference Proceedings* (2022), SIGGRAPH '22, Association for Computing Machinery, p. 1–8. doi:10.1145/3528233.3530755. 3

[JSZ*21] Jiang H., Sheng Z., Zhu S., Dong Z., Huang R.: UniFuse: Unidirectional Fusion for 360° Panorama Depth Estimation. *IEEE Robotics and Automation Letters 6* (2021), 1519–1526. doi:10.1109/LRA.2021.3058957. 2

[JZWR20] Jones B., Zhang Y., Wong P., Rintel S.: VROOM: Virtual Robot Overlay for Online Meetings. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (2020). 2

[KKK*23] Kim J., Kim D., Kim B., Kim H., Lee J.: Holobot: Hologram based extended reality telepresence robot. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction* (2023), HRI '23, p. 60–64. doi:10.1145/3568294.3580043. 2

[KMA*20] Kopf J., Matzen K., Alsisan S., Quigley O., Ge F., Chong Y., Patterson J., Frahm J.-M., Wu S., Yu M., Zhang P., He Z., Vajda P., Saraf A., Cohen M.: One shot 3D photography. *ACM Transactions on Graphics 39*, 4 (2020), 76:76:1–76:76:13. doi:10.1145/3386569.3392420. 3

[LGY*22] Li Y., Guo Y., Yan Z., Huang X., Duan Y., Ren L.: OmniFusion: 360 Monocular Depth Estimation via Geometry-Aware Fusion. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), IEEE Computer Society, p. 2791–2800. doi:10.1109/CVPR52688.2022.00282. 2, 4

[LOX*20] Li R., Olszewski K., Xiu Y., Saito S., Huang Z., Li H.: Volumetric Human Teleportation. In *ACM SIGGRAPH 2020 Real-Time Live* (2020), SIGGRAPH 2020, Association for Computing Machinery. doi:10.1145/3407662.3407756. 5

[LTT15] Lee C.-C., Tabatabai A., Tashiro K.: Free viewpoint video (FVV) survey and future research direction. *APSIPA Transactions on Signal and Information Processing 4* (2015). doi:10.1017/ATSIP.2015.18. 1

[LXLL19] Lai P. K., Xie S., Lang J., Laganière R.: Real-Time Panoramic Depth Maps from Omni-directional Stereo Images for 6 DoF Videos in Virtual Reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2019), p. 405–412. doi:10.1109/VR.2019.8798016. 3

[LXM*20] Lin K.-E., Xu Z., Mildenhall B., Srinivasan P. P., Hold-Geoffroy Y., DiVerdi S., Sun Q., Sunkavalli K., Ramamoorthi R.: Deep Multi Depth Panoramas for View Synthesis. In *Computer Vision – ECCV 2020* (Berlin, Heidelberg, 2020), Springer-Verlag, p. 328–344. doi:10.1007/978-3-030-58601-0_20. 3

[MJJK21] Meuleman A., Jang H., Jeon D. S., Kim M. H.: Real-Time Sphere Sweeping Stereo from Multiview Fisheye Images. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN, USA, 2021), IEEE, p. 11418–11427. doi:10.1109/CVPR46437.2021.01126. 3

[MKK*20] Mühlhausen M., Kappel M., Kassubeck M., Bittner P., Castillo S., Magnor M.: Temporal Consistent Motion Parallax for Omnidirectional Stereo Panorama Video. In *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology* (2020), Association for Computing Machinery, p. 1–9. doi:10.1145/3385956.3418965. 3

[MLPALC21] Mazzola G., Lo Presti L., Ardizzone E., La Cascia M.: A Dataset of Annotated Omnidirectional Videos for Distancing Applications. *Journal of Imaging 7*, 8 (2021), 158. doi:10.3390/jimaging7080158. 6

[MS19] Minja A., Senk V.: Quasi-Analytical Simulation Method for Estimating the Error Probability of Star domain Decoders. *IEEE Transactions on Communications 67* (2019), 3101–3113. doi:10.1109/TCOMM.2019.2895829. 3

[OERF*16] Orts-Escolano S., Rhemann C., Fanello S., Chang W., Kowdle A., Degtyarev Y., Kim D., Davidson P. L., Khamis S., Dou M., Tankovich V., Loop C., Cai Q., Chou P. A., Mennicken S., Valentin J., Pradeep V., Wang S., Kang S. B., Kohli P., Lutchyn Y., Keskin C., Izadi S.: Holoportation: Virtual 3D Teleportation in Real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (2016), Association for Computing Machinery, p. 741–754. doi:10.1145/2984511.2984517. 1

[Ohl18] Ohl S.: Tele-immersion concepts. *IEEE Transactions on Visualization and Computer Graphics 24*, 10 (2018), 2827–2842. doi:10.1109/TVCG.2017.2767590. 1

[PBEP01] Peleg S., Ben-Ezra M., Pritch Y.: Omnistereo: Panoramic stereo imaging. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23* (2001), 279–290. doi:10.1109/34.910880. 3

[RAYR22] Rey-Area M., Yuan M., Richardt C.: 360MonoDepth: High-Resolution 360° Monocular Depth Estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), IEEE Computer Society, p. 3752–3762. doi:10.1109/CVPR52688.2022.00374. 2, 4

[RLHK20] Ranftl R., Lasinger K., Hafner D., Koltun V.: Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–1. doi:10.1109/TPAMI.2020.3019967. 4

[RTM*20] Rhee T., Thompson S., Medeiros D., Anjos R., Chalmers A.: Augmented Virtual Teleportation for High-Fidelity Telecollaboration. *IEEE Transactions on Visualization and Computer Graphics PP* (2020), 1–1. doi:10.1109/TVCG.2020.2973065. 2

[RTW20] Richardt C., Tompkin J., Wetzstein G.: *Capture, Reconstruction, and Representation of the Visual Real World for Virtual Reality*. Lecture Notes in Computer Science. Springer International Publishing, 2020, p. 3–32. doi:10.1007/978-3-030-41816-8_1. 1, 3

[SDMR20] Sarlin P.-E., DeTone D., Malisiewicz T., Rabinovich A.: SuperGlue: Learning Feature Matching with Graph Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), IEEE Computer Society. doi:10.1109/CVPR42600.2020.00499. 4

[SGHS98] Shade J., Gortler S., He L.-w., Szeliski R.: Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), SIGGRAPH '98, Association for Computing Machinery, p. 231–242. doi:10.1145/280814.280882. 3

[SHN*19] Saito S., Huang Z., Natsume R., Morishima S., Kanazawa A., Li H.: PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In *The IEEE International Conference on Computer Vision (ICCV)* (2019), p. 2304–2314. 5

[SHSC19] Sun C., Hsiao C.-W., Sun M., Chen H.-T.: HorizonNet: Learning Room Layout with 1D Representation and Pano Stretch Data Augmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), IEEE Computer Society, p. 1047–1056. doi:10.1109/CVPR.2019.00114. 2

[SKC*19] Serrano A., Kim I., Chen Z., DiVerdi S., Gutierrez D., Hertzmann A., Masiá B.: Motion parallax for 360° RGBD video. *IEEE Transactions on Visualization and Computer Graphics* (2019). doi:10.1109/TVCG.2019.2898757. 3

[SLM*22] Suvorov R., Logacheva E., Mashikhin A., Remizova A., Ashukha A., Silvestrov A., Kong N., Goka H., Park K., Lempitsky V.: Resolution-robust Large Mask Inpainting with Fourier Convolutions. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2022), IEEE, p. 3172–3182. doi:10.1109/WACV51458.2022.00323. 4

[SPK22] Sofiiuk K., Petrov I. A., Konushin A.: Reviving Iterative Training with Mask Guidance for Interactive Segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)* (2022), p. 3141–3145. doi:10.1109/ICIP46576.2022.9897365. 7

[SSC21] Sun C., Sun M., Chen H.-T.: HoHoNet: 360 Indoor Holistic Understanding with Latent Horizontal Features. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), IEEE Computer Society. doi:10.1109/CVPR46437.2021.00260. 2

[SSKH20] Shih M.-L., Su S.-Y., Kopf J., Huang J.-B.: 3D Photography using Context-aware Layered Depth Inpainting. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020). doi:10.1109/CVPR42600.2020.00805. 3

[TLL*19] Teo T., Lawrence L., Lee G. A., Billinghurst M., Adcock M.: Mixed reality remote collaboration combining 360 video and 3d reconstruction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), CHI '19, Association for Computing Machinery, p. 1–14. doi:10.1145/3290605.3300431. 2

[TS20] Tucker R., Snavely N.: Single-View View Synthesis With Multiplane Images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), IEEE, p. 548–557. doi:10.1109/CVPR42600.2020.00063. 3

[WGD*22] Waidhofer J., Gadgil R., Dickson A., Zollmann S., Ventura J.: PanoSynthVR: View Synthesis From A Single Input Panorama with Multi-Cylinder Images. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2022), p. 584–592. doi:10.1109/ISMAR55827.2022.00075. 3

[WYS*20] Wang F.-E., Yeh Y.-H., Sun M., Chiu W.-C., Tsai Y.-H.: BiFuse: Monocular 360 Depth Estimation via Bi-Projection Fusion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), p. 459–468. doi:10.1109/CVPR42600.2020.00054. 2

[WYS*21] Wang F.-E., Yeh Y.-H., Sun M., Chiu W.-C., Tsai Y.-H.: LED2-Net: Monocular 360 Layout Estimation via Differentiable Depth Rendering. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), IEEE Computer Society, p. 12951–12960. doi:10.1109/CVPR46437.2021.01276. 2

[YQC*18] Yang W., Qian Y., Cricri F., Fan L., Kamarainen J.-K.: Object Detection in Equirectangular Panorama. In *2018 24th International Conference on Pattern Recognition (ICPR)* (2018), IEEE Computer Society, p. 2190–2195. doi:10.1109/ICPR.2018.8546070. 4

[YWP*19] Yang S.-T., Wang F.-E., Peng C.-H., Wonka P., Sun M., Chu H.-K.: DuLa-Net: A Dual-Projection Network for Estimating Room Layouts from a Single RGB Panorama. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), IEEE Computer Society, p. 3358–3367. doi:10.1109/CVPR.2019.00348. 2

[ZKZD18] Zioulis N., Karakottas A., Zarpalas D., Daras P.: OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas. In *Computer Vision – ECCV 2018* (2018), Lecture Notes in Computer Science, Springer International Publishing, p. 453–471. doi:10.1007/978-3-030-01231-1_28. 2

[ZLM*22] Zanetti M., Luchetti A., Maheshwari S., Kalkofen D., Ortega M. L., De Cecco M.: Object Pose Detection to Enable 3D Interaction from 2D Equirectangular Images in Mixed Reality Educational Settings. *Applied Sciences 12*, 1111 (2022), 5309. doi:10.3390/app12115309. 3