University of Brasília

Exact Sciences Institute
Computer Science Department

Volumetric video capture, object reconstruction and telecommunication methods for Mixed Reality experiences

Rafael Diniz

Thesis submitted in partial fullfilment of
the requirements to qualify for Doctoral Degree in Informatics

Advisor
Prof.a Dr.a Mylène Christine Queiroz de Farias

Brasília
2018

University of Brasília

Exact Sciences Institute
Computer Science Department

Volumetric video capture, object reconstruction and telecommunication methods for Mixed Reality experiences

Rafael Diniz

Thesis submitted in partial fullfilment of
the requirements to qualify for Doctoral Degree in Informatics

Prof.a Dr.a Mylène Christine Queiroz de Farias (Advisor)
UnB

Prof. Dr. Prof. Eduardo Peixoto Fernandes da Silva
UnB

Prof. Dr. Camilo Chang Dorea
UnB

Prof. Dr. Guido Lemos
Federal University of Paraíba

Prof. Dr. Bruno L. Macchiavello Espinoza
Coordinator of Graduate Program in Informatics

Brasília, December 7, 2018

Abstract

Volumetric video is a core underlying technology for emerging Mixed Reality systems. What was previously available for a glimpse only in science fiction movies and futuristic predictions, now with the ongoing research on volumetric video capturing, coding and presentation, realistic mixed reality experiences are close to become a reality. At the same time, computer generated holography and other digital 3D projection techniques start to became more common and affordable. The emergence of solutions for capturing volumetric video and devices which can display volumetric video mixed with the real world are paving the way to a new media, where a real object and its volumetric virtual image are indistinguishable. In this text we make an analysis of the state of the art in volumetric video technologies related to creating mixed reality experiences, and propose some technical novelties in order to allow a more broad development and adoption of volumetric video applications. Considering the current challenges of capturing volumetric objects or scenes, it is proposed a system for capturing volumetric content from a single RGB-D capture device. Motion estimation, a key element for allowing compression of dynamic elements, is also addressed with a global motion estimation method for volumetric video. An adaptive volumetric video transmission method for packet-switched networks is also covered and, finally, a volumetric video metric for quality evaluation initially made for mesh is being adapted to work with voxelized point-clouds. The methodology of application of the metric to evaluate Mixed Reality volumetric content is also included. The ultimate goal of this work is to provide elements for the dawn of realistic mixed reality applications, be it a tele-conference, a distance learning class, or a volumetric talk-show TV broadcast.

Keywords: RBG-D, volumetric video, point cloud, mesh, virtual reality, mixed reality

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Volumetric video is the technology behind all new immersive Mixed Reality (MR) systems which are being developed or are already available. Different from 2D video, volumetric video represents the 3D surface of objects, adding geometric coordinates to each color component present of a scene. In order to enable Mixed Reality (MR) applications, a new set of algorithms and techniques are being developed and standardized, along with efforts to launch new hardware for capturing, processing and presenting volumetric content in a MR experience. It is expected that a MR system is able to blend real world and virtual computer-generated graphics in a realistic way.

The concept of Mixed Reality was introduced by Paul Milgram and Fumio Kishino [3], and its defined in terms of the Virtuality Continuum, where a Mixed Reality environment lies anywhere between the extremes of the Virtuality Continuum, as shown in Figure 1.1.

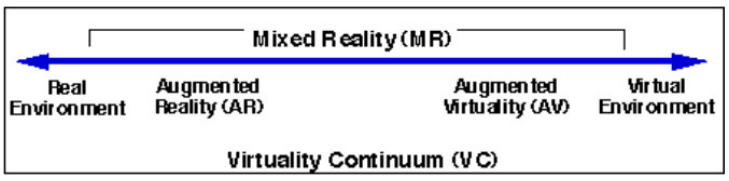

Figure 1.1: Milgram's simplified Mixed Reality definition in terms of the Virtuality Continuum.

Volumetric video, the media representation of current MR experiences, cannot be easily captured as it relies on 3D reconstruction of each scene, instead of just a single view of it, like in 2D video. In traditional video, a CCD[1] or CMOS[2] based light sensor produces a 2D array of pixels after an analog to digital conversion process, whereas for volumetric videos there is a need for a reconstruction process, which makes use of range

[1]CCD: Charge-Coupled Device.
[2]CMOS: Complementary Metal-Oxide semiconductor.

sensors and is based on the structured light or the time-of-flight sensing technology. An illustration of this reconstruction process is shown in Figure 1.2. In the future, instead of capturing the world as color and geometric shapes, the light field itself will be able to be recorded with all its amplitude and phase information, and reproduced, for example, through near-eye [4] or retinal [5] light field holographic projectors. Instead of volumetric elements (voxels), holographic elements (hogels) will be used.

Volumetric visual content can be experienced in head-mounted displays (HMD), 3D flat screens or through holographic projectors. Considering their general availability and the possibility to create mixed reality experiences, the focus in this text is on head-mounted displays, especially the ones which allow mixed reality experiences. Typically, HMDs are classified in two types depending on the experience provided: Virtual Reality (VR) or Augmented/Mixed Reality. The VR type presents the volumetric graphics in a screen inside an opaque device, where there is no blending of real and virtual. The Augmented or Mixed Reality types allow for the users to visualize both real world and computer generated content, as shown in Figure 1.3.

Figure 1.2: User with a Oculus Rift VR glass interacting with another user, while being captured by a setup of three (one hidden) RGB + Depth Microsoft Kinect sensors. A synthesized scene with the two users is shown in the TV in the back.

In order to create a mixed reality environment, interactivity is paramount, and an HMD needs to provide a 6 degrees of freedom to the user. This requires additional sensors to track the head position and the eyes of the user. Also, it is necessary to have sensors to map the external world, which allows the HMD device to know where to project a volumetric element in the field of view of a user, as illustrated in Figure 1.4. To present a smooth and realistic blend between real and virtual worlds, mixed reality HMD devices must account on user's localization, eyes and head tracking, and external world mapping, allowing a user to have infinite ways of visualize a scene, given that 6 degrees of freedom are allowed.

Figure 1.3: User viewing a remote located kid though a MR device. Top left shows the real scene and bottom left the way the scene is viewed in the Microsoft's MR HMD, called HoloLens.

Figure 1.4: Magic Leap One mixed reality head-mounted device. In the image it's possible to see some of the many sensors the device has, including depth sensor.

Volumetric video has many use cases, like for example a live multi-party volumetric video conference, where each person in the conference has it's own volumetric video captured and transmitted, while receiving volumetric streams from other participants (e.g. Microsoft's Holoportation [6]). It is worth pointing out that volumetric video allows a more accurate representation of the world, being extremely useful for industrial, medical, educational, gaming and recreational applications, among other purposes.

Currently there is a standardization effort by a joint ISO/IEC/MPEG group (See [7]) to launch more than one codec for dynamic and static volumetric imagery. Despite this, other advances in fields like volumetric video capture, production and exhibition also need to improve to allow proper real world implementations of 3D immersive experiences with six degrees of freedom interactivity.

This work has the objective to fill some of the gaps which prevents realistic Mixed Reality experiences outside expensive laboratories. Considering that most algorithms available use volumetric video created from an array of cameras, which is a much more expensive setup, the proposed algorithms can be used in practical scenarios, where users do not have access to expensive equipment. Also a new motion estimation method based on registration of solid bodies is proposed, which is designed to improve the coding rate of a volumetric video. An adaptive volumetric video transmission method is also proposed for networks without guaranteed bandwidth reservation, and, finally, in order to evaluate the quality of volumetric video content, a new objective metric for volumetric video quality assessment is proposed.

The main contribution of this work is the development of new algorithms and methods for volumetric video, with focus on enabling simple setups, like one with a single RGB-D capture device is available for volumetric reconstruction.

This document is organized in the following chapters: this introduction, a second chapter with a theoretical overview, followed by the proposal and early results of this doctoral work, then a forth chapter chapter with conclusions and a work plan schedule.

Chapter 2

Volumetric Video Theoretical Overview

This chapter contains an overview about volumetric video, a review of the RGB-D sensors used to create the volumetric video, and a discussion about quality evaluation methods.

2.1 Volumetric video overview

Real-time 3D volumetric video systems use either a mesh or a point-cloud format to represent the 3D objects. Unfortunately, given the order and the topology of the 3D-frames, the computational complexity of these systems is higher then 2D video [8]. It is worth mentioning that the data acquired from one or more RGB-D sensors does not have surface information, only color information for different 3D coordinates.

Point-clouds are composed of 3D coordinates plus color and sometimes also other attributes like surface normal and reflectance, while mesh also contains polygonal surfaces connecting the points. By being a simpler representation, point-clouds are usually preferred for real-time applications. Indeed, voxelized point-clouds are pretty much being used in state-of-the-art codecs being developed for volumetric video [1] [2]. A voxelized point-cloud is a point-cloud where its 3D points are converted to voxels, which are 3D small boxes in a 3D grid of a bounded region. A correctly voxelized point-cloud can be used to represent solid objects [9], but an incorrectly voxelized point-cloud might present empty voxels (holes) between each occupied voxel, allowing an user to see through an object, thus reducing the quality of experience. A mesh representation, obviously, does not suffer from this problem.

In 3D computer graphics field, 3D polygon meshes are being used for decades to represent 3D object volumetric data. However, point-clouds are simpler to obtain than 3D polygon meshes, as surface reconstruction does not need be computed. Point-clouds

have a more compact representation, as they do not store any connectivity among the points, so they are more computationally efficient, a relevant aspect for real-time systems.

An important data structure which is very popular to represent a point-cloud is the octree [10] [11]. Octree partitioning keep diving a volume in small sub-volumes, while there is one or more points inside the volume, as presented in Figure 2.1. Other data structures exist, like spanning trees [12], binary trees [13] or based on a graph representation [14], but by far, octrees are the most popular.

Figure 2.1: Two variations of octree, in the left a traditional octree partitioning where cubes with points are divided until a target layer is reach, and in the right an octree approach where partitioning a cube is based on split decisions, for example, based on RDO (Rate-Distortion Optimization).

Volumetric objects can also be segmented and then projected into 2D frames to be encoded as common 2D image or vídeo [15] [16]. As detailed by Schwarz [7], most of today's volumetric video coding systems still rely on standard 2D video codecs available in current hardware. As a consequence, volumetric video is obtained by projected 3D data in 2D images and the projection and geometric parameters are transported as metadata. Schwarz mentions that compression solutions for volumetric video representations are poor in both spatial and temporal dimensions, being motion estimation in 3-D space an ill-defined problem. Nevertheless, many new codecs for voxelized point-clouds are being proposed [1] [2] [17].

As mentioned earlier, capturing the world in 3D is not an easy task. Most of the currently available work suggest the use of at least 3 RGB-D sensors for a fair volumetric reconstruction [18]. The difficulty in assembling and calibrating such an array of RGB-D sensors limits their popularity in more mainstream communication applications. Previous to the availability of affordable RGB-D sensors, multi-view stereo based volumetric reconstruction was typically used. Stereo based volumetric reconstruction relies on captured views from different cameras to extract the volumetric shape from an object [19]. This area is a pretty mature field of research, but has its limitations due to inherent absence of real captured depth data.

Although there is a lot of work in the field of volumetric video reconstruction, the reconstruction of 3D objects from RGB and Depth frames still faces many challenges, like for example, noisy and missing data acquired from the sensors [20] and lighting differences

between sensors. The reconstruction methods available to obtain of a volumetric content from multiple RGB-D capture devices is examined in depth by Berger [20], which first classifies reconstruction methods in relation to point-cloud artefacts, like missing data, misalignment and non-uniform sampling, and input requirements, like presence or not of surface normals. The mentioned techniques of surface reconstruction, as pointed by Berger, has grown from methods that handle limited defects in point-clouds reconstructions, to methods that handle substantial artefacts, and he also discusses about a growing development of data-driven reconstruction algorithms, which use large priors database and allows for a method to identify classes and properties of objects. As explained, reconstruction can use the color and geometric data from sensors only, but also use prior information. Firman et at. [21], for example, proposes a structure prediction of unobserved voxels from single depth image by employing classes of 3D mesh models used as reference for the reconstruction. Alexiadis et al. [22] also proposes a reconstruction method specific for human performance reconstruction, while Boldi et al. [23] proposes a method specific for face reconstruction. On the specific case of 3D volumetric face reconstruction, some other approaches uses just 2D color images as input also, like in [24] and [25].

Concerning deep learning approaches for 3D shape generation and completion, 3D ShapeNets [26] uses deep learning to train a 3D convolutional network from a shape database, and complete or repair shapes, including broken meshes [27]. Other work which uses deep learning for object shape reconstruction includes Rock et at [28] and Song et al.[29], all with a similar approach, using deep learning knowledge acquired with volumetric objects data-sets.

There is in the literature, a few works that perform a 3D scan using a single consumer grade RGB-D. Among these, is that work of Hernandez et al. that implements a 3D face scan using a single RGB-D device [30]. Also, prior Kinect Fusion work [31], provides tools for performing a good quality 3D scanning using just one RGB-D sensor. Kinect Fusion's technique consists of rotating the camera or the object in order to capture all its faces.

An interactive mixed reality experience can have interactive synthesized 3D objects with pre-defined behavior. Interactive 3D objects can be described through a cause and effect paradigm as provided by specific domain languages, like NCL (Nested Context Language) [32], for both Augmented and Mixed reality experiences.

2.2 RBG-D sensors overview

Current RGB-D camera devices provide at least two separate streams: color and depth (also the captured IR frame is usually available). These streams come from different types of sensors inside the device. Naturally, each sensor has different accuracy and

noise levels and, typically, there is no pixel alignment between the two streams. In this work, we consider two types of RGB-D devices, which are classified by the depth ranging technology: structured light [33] and time-of-flight based [34]. The sensors of the Kinect 1 are structured light based, while the sensors of the Kinect 2 are time-of-flight based (See fig. 2.2 containing Kinects 1 and 2 on a tripod).

Figure 2.2: Kinect 2 (left) and Kinect 1 (right) assembled on a tripod, in the configuration used in the capture experiments.

It is worth mentioning that the Kinect for Xbox 360 (Kinect 1) was the first widely available RGB-D sensor. The Kinect 1 hardware, shown in Figure 2.3, comes with a RGB camera, an Infrared (IR) projector and an IR camera. The depth sensor is based on the structured light principle and is composed of an IR projector combined with an IR camera. The IR projector projects a known pattern of IR dots to a scene and the IR camera captures this projected pattern. By comparing the projected with the captured IR pattern, the sensor can obtain the depth information [35]. The outputs of the sensor are transmitted over the USB 2 interface and are composed of a chroma subsampled (bayer-pattern) of 8 bit/pixel RGB and 11 bit/pixel depth streams. For a typical USB transfer mode, both streams have 640x480 pixels at 30fps.

Kinect for Xbox One, or just Kinect 2, is a time-of-flight based device and is currently the most widely used RGB-D capture device for volumetric video production. Kinect 2, shown in Figure 2.4, has a 1920x1080 HD RGB camera and uses a different ranging technology when compared to the Kinect 1. It has an IR light source which emits a

modulated square wave. With the receiver's captured phase shift information, the sensor can compute the depth. Kinect 2's time-of-flight range sensor outputs a 512x424 depth frame, which together with the RGB information, is transmitted over USB 3 bus to a host computer. The depth sensor has a better accuracy and less noisy output than the structured light based Kinect 1 [36].

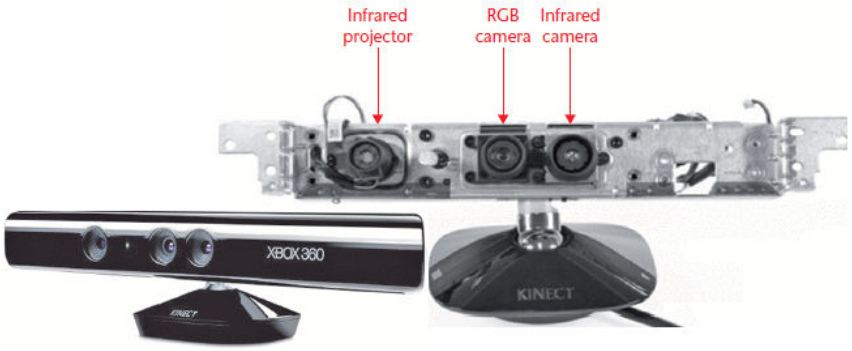

Figure 2.3: Kinect 1 hardware, with it's Infrared projector, RGB camera and Infrared camera.

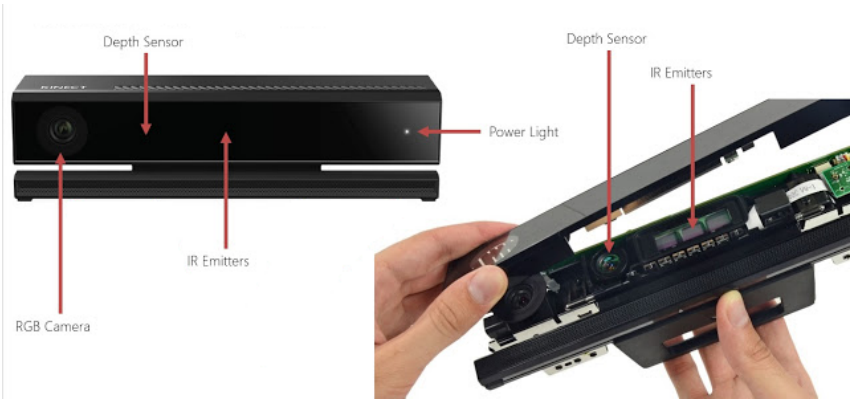

Figure 2.4: Kinect 2 hardware, with it's IR emitters, depth sensor and RGB camera.

Also relevant in the RGB-D sensors ecosystem are the ASUS' Xtion PRO live, Occipital's Structure Sensor and the Intel's RealSense, which are structured light based sensors [37]. Other time-of-flight based sensors exist, but (unlike Kinect 2) are more expensive and do not always come with an embedded RGB camera. One example is the Fotonic or SwissRanger sensors [38]. An interesting research project by Carnegie Mellon and University of Toronto is the EpiToF, which consists of a time-of-flight sensor that can block ambient light, allowing it to operate in an outdoors environment [39]. Other simpler sensors, which claim depth estimation have stereo cameras and use software to perform photogrammetric stereo image to depth estimate.

The accuracy and depth working range of consumer grade 3D sensors are evaluated by Guidi [40] and Schöning [37]. Kinect 2 device seems to be have a clear advantage over the others in terms of depth working range, which ranges from 0.5 to 4.5m, field-of-view of 70.6° by 60° (HxV) and a good depth accuracy in the millimeter scale. Typically to perform live volumetric video capture, a set of Kinect 2 devices are used (see [18] and [41]), being very common in both VR and MR experiences[1].

2.3 Volumetric video quality evaluation

The quality of the experience of volumetric video was addressed by Desai et al. [42]. Their method is based on (1) global parameters that take into consideration holes and missing parts of the human body and (2) local factors that consider the details of the represented face, as shown in Figure 2.5. Doumanoglou [43], on the other hand, compared the relative importance of the quality of the geometric representation and the texture resolution. They also studied the impact of the network, studying effects of parameters, like latency. Both Desai and Doumanoglou's metrics are non-reference metrics and make use of deep learning based prediction model to obtain quality scores which derive from a trained network.

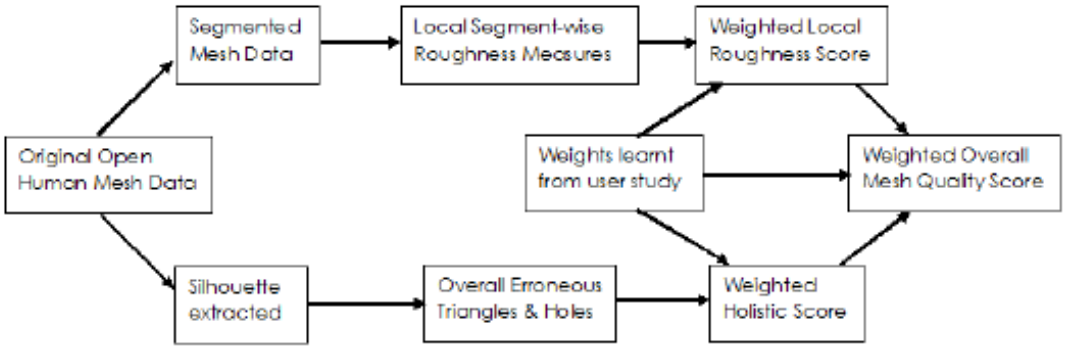

Figure 2.5: Desai's learning based objective evaluation for 3D human meshes schematic.

A more traditional full-reference quality metric for geometric distortion is given by Tian [44], which proposes a point-to-plane based PSNR. For evaluation of color distortion, Queiroz [45] proposes to project the 3D object in the faces of a cube and use 2D metrics to calculate the distortions. A follow up work of Queiroz's full-reference metric was Torlig [46]'s one, improving the evaluation of known 2D image metrics applied to 6 orthographic projections of a 3D object. Torlig tested PSNR (and variations), SSIM, MSSIM and VIFP metrics, and related that MSSIM and VIFP performed best. Also Alexiou et

[1]In this classification of VR and MR experiences, which are typically requires real-time constraints, we exclude setups that rely on LiDAR for capture, since they operate with one depth value per sample.

al. [47] uses 2D metrics to evaluate octree-based compression artifacts of point-clouds, rendered as mesh objects, and concluded that the subjective results are affected by the usage of a surface reconstruction algorithm, and that state-of-art objective point-cloud metrics can not sufficiently predict the quality of every content in the test conducted.

Zhang [48] discuss the methodology of a subjective quality analysis, how to use current ITU recommendations for subjective quality assessment of volumetric content, and presents his results, concluding geometric noise can affect more the quality than color noise among other conclusions. Alexiou et al [49] discusses a methodology on how to subjectively evaluate the geometric quality of point-cloud, although using very simple synthetic objects. Other work includes Javaheri et al. [50], which who subjective quality evaluation of denoising algorithm, and concluded that point-to-plane metrics better represents human vision quality perception than RMSE the Hausdorff distance.

Chapter 3

Proposals and Early Results

Considering the challenges for implementation of a good quality live tele-immersion system based on volumetric video, the focus of this work is to enable volumetric video use cases in currently available consumer hardware. For example, a live volumetric tele-presence communication system or a live volumetric broadcast application.

To address the challenge of a more broad adoption for volumetric video systems, in this Ph.D. work, we propose:

- An improved algorithm for depth hole filling which uses the color differences to infer the depth;
- A method for reconstruction of volumetric video objects from a single sensor;
- One method for global motion estimation which uses the transformation matrix output of the registration of frames in a volumetric video;
- An adaptive point cloud transmission method for packet-switched based networks which downsamples the point cloud to match the available bitrate in the network;
- A framework for open meshes quality evaluation will be adapted to work with voxelized point-clouds.

The proposals already implemented were written in C and C++, and the code contains operations to perform all the basic point cloud and mesh operations, like geometric transformations, point to point distance, crop and merge, apart of the algorithms proposed in this text. The following libraries were used for the development of this work: OpenKinect's project libfreenect and libfreenect2 [51], for Kinects support, and Open3D [52], for the registration implementation among other useful volumetric data structures Open3D library provides. The proposals were tested in both a high-end computer and a notebook

computer. The high-end computer is a dual eight-core (32 SMT[1]) Intel Xeon E5-2620, with 80GB of RAM memory and two video cards, a NVidia Quadro P6000 and a NVidia GeForce GTX 1080. The notebook computer is a Lenovo ThinkPad T430 with a dual-core (4 SMT) Intel Core i5-3320M with 8GB of RAM with an external NVidia GTX 1080 GPU (see fig. 3.1). The notebook setup is also used when capturing outside the laboratory.

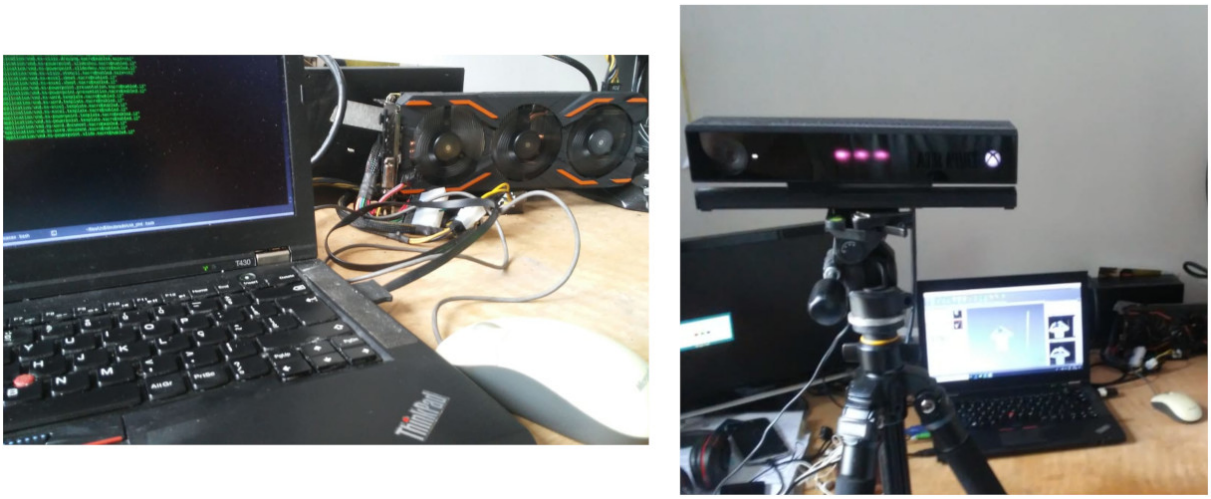

Figure 3.1: Partial view of the notebook with an external GPU attached (left) and the Kinect 2 connected to the notebook computer used for capture outside the laboratory (right).

In this chapter we present the proposals and the state of its implementation and results already obtained. The first section contains the description of framework of volumetric video system, followed by the proposed contributions of this work. Section two contains the proposed hole filling algorithm and, in section three, the volumetric reconstruction method which takes as input the RGB and Depth streams from a single capture device. Section four describes an innovative motion estimation method, while section five contains an adaptive point-cloud sampling for communication networks. Finally, section six has a discussion and a proposal of a metric for objective quality evaluation of volumetric content.

3.1 Volumetric Video Framework

In order to achieve a realistic mixed reality presentation of a remote object or scene, a complete volumetric representation must be captured. To acquire a volumetric video stream of an object, it is also necessary to perform some data processing and volumetric

[1]SMT: Simultaneous multithreading permits current CPUs to share CPU resources among 2 independent threads, improving the overall performance.

reconstruction steps. But, currently there are some challenges for making these algorithms work in real-time given the processing power and network constraints available today. This is particularly true when we take into account hardware and mobile network connections.

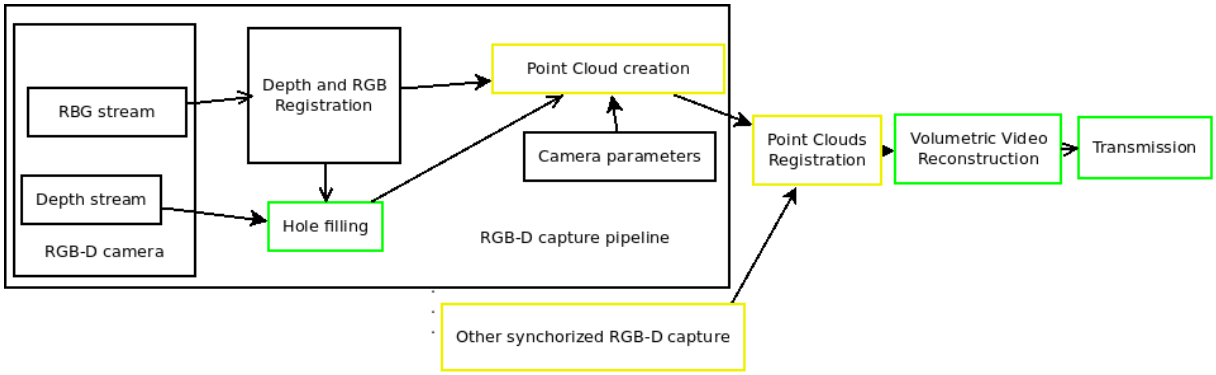

Figure 3.2: Diagram of software and hardware framework proposed in this work. The framework includes the RGB-D capture (left) and the volumetric video reconstruction (right). The block in green represents that contributions were are already implemented, while the blocks in yellow represents work in progress.

Our goal in this work is to implement a real-scenario tele-immersion application, using currently available customer hardware. The proposed hardware and software framework includes contributions on key aspects of the volumetric video workflow. Figure 3.2 shows the proposed framework architecture. In the figure, the parts of the framework that were already implemented are marked in green, while the parts that still need to be implemented are in yellow. In the next sections, we describe each of these parts of the framework, discussing the challenges and the approach that will be used in this work.

3.2 Improved hole filling algorithm

Typically, a fast hole filling algorithm (with parallel execution capabilities) is used to perform an initial hole filling on the depth map, which reduces the number of introduced errors. One algorithm that is commonly used, for example in KinectFusion [31], is the Inverse Distance Weighted (IDW) interpolation algorithm that is able to estimate an absent Z value by computing a weighted average of the surrounding depth values [53]. Another example of hole filling algorithm is the Natural Neighbor Interpolation (NNI), which computes the weights using a Voronoi-diagram of the depth values [54].

We propose a gap filling algorithm for the depth map that takes into account the color information to compute the missing depth values. To maintain the parallel execution and the independence of the spatial data, the proposed gap filling algorithm runs locally

in spatial neighborhoods of the RGB and Depth 2D frames. The pseudo code of the proposed algorithm is in Algorithm 1. In the proposed algorithm, if a depth value is missing in a given X and Y coordinate of the depth map, the 8-neighborhood pixels of the X,Y pixel in the RGB frame are compared to the color of the pixel with missing depth value. The pixel in the neighborhood with more similar color to the pixel with missing depth is then used as a prediction of the value used to fill the depth hole. The color frame is not modified, but some pixel color values which were lost due to its depth value being invalid, can now appear in the reconstructed RGB-D frame when the proposed algorithm fills a missing depth value. As a result, the noise produced during capture is attenuated and small holes are closed.

```
Data: RGB, Depth Map
Result: Depth Map after depth hole filling
1 rgb_frame ← registered_rgb;
2 depth_frame ← registered_depth;
3 for i = 0; i < width*height; i++ do
4 |   if depth_frame[i] = nil then
5 |   |   8neighborhood ← get_8_neighborhood(rgb_frame[i]);
6 |   |   sel_neighbor ← get_near_color(rgb_frame[i], 8neighborhood);
7 |   |   depth_frame[i] ← depth_frame[sel_neighbor];
8 |   end
9 end
```

Algorithm 1: Description of the proposed gap filling algorithm.

The algorithm properly closes small holes and also completes small occluded regions to the depth sensor. In Figure 3.6, the algorithm is able complete an occluded part of nose, improving subjective quality perception.

Some simulations were carried to evaluate the subjective quality of the algorithm. By artificially creating holes in three different positions of an example depth map of a captured RGB-D frame, we obtained results shown in Figure 3.3, fig. 3.4 and fig. 3.5. In Figure 3.3 and 3.5 a 5x5 depth map hole was introduced in different parts, while Figure 3.4 had a 5x10 hole added to the depth map.

This algorithm is particularly good compared to other interpolators because it uses the color information as guidance for recreation of the depth value. The efficacy of the proposed algorithm relies on the fact that current RGB-D sensors have a better accuracy in the video camera than in the depth sensor. The image frames contain no holes and have low noise and the depth frames have holes and higher noise.

In related work, for example, like in Queiroz's MCIC [1] or Mekuria's point-cloud codec [2], there can be seem artifacts in the point-clouds given as example, like shown in Figure 3.7. These artifacts can be attenuated with the proposed filter.

Figure 3.3: Point-cloud created from the original RGB-D frame (left), a point-cloud created with a hole in the depth map in the region of the t-shirt (center) and the view of the point-cloud (right) after the application of the proposed algorithm to the frame with hole (right).

Figure 3.4: Point-cloud created from the original RGB-D frame (left), a point-cloud created with a hole in the depth map in the region of the beard (center) and the view of the point-cloud (right) after the application of the proposed algorithm to the frame with hole.

3.3 Single view RGB-D volumetric reconstruction

The proposed framework for volumetric reconstruction first creates a complete volumetric representation of each person which will join a volumetric video session. For this, we use a methodology based on the Truncated Signed Distance Function [55] and Kinect Fu-

Figure 3.5: Point-cloud created from the original RGB-D frame (left), a point-cloud created with a hole in the depth map in the region of the eye (center) and the view of the point-cloud (right) after the application of the proposed algorithm to the frame with hole.

Figure 3.6: A point-cloud created from a single RGB-D frame (left), and the same frame after applied the proposed hole filling algorithm (right).

sion [31], which assembles the volumetric 3D object representation by moving the capture device around the object (in this case, a person). Then, the volumetric model of the person is stored, in a point-cloud format.

Volumetric scene completion from single view RGB-D capture is also proposed by [28], which uses large categorized 3D models for object shape retrieval. Song [29] proposed a semantic scene completion, which uses a 3D convolutional neural network trained with large 3D scene datasets (E.g. 45,622 houses with 775,574 rooms). Yang [56] uses a generative adversarial network to reconstruct objects from a single view capture and uses large databases to train the network, and claims to be the state of art in its class. Instead of

Figure 3.7: Point-cloud examples given by Queiroz [1] (left) and Mekuria [2] (right) used as source media for encoding experiments. Artifacts can be seem in both point-cloud examples.

using a deep learning approach, our work relies only on fast geometric transformations, so its fast and does not necessarily require a powerful GPU, like the deep learning implementations require. Yang proposes the highest resolution among the deep learning based volumetric reconstruction methods of 256^3 voxel space, while we propose to support at least a 1024^3 voxel space and more than one million of voxels, taking as reference the volumetric video test material sent ISO/IEC/MPEG Point-Cloud Compression group by 8i[57], which has frames of up to 1024^3 voxel resolution and typically more than one million occupied voxels.

The big advantage of the proposed reconstruction system is its simplicity in the capture setup. The proposed system uses for volumetric capture just one RGB-D capture device to reconstruct a 3D representation of a human figure (speaker), greatly simplifying the acquisition procedure in applications like mixed-reality volumetric video teleconferences.

The proposed solution can be used not only in live two-way communication, but also in volumetric video broadcasting and Internet volumetric video services where the a person is giving a speech, giving a class, presenting the news or playing an online game in volumetric video format. The system prioritizes the data corresponding to the participants' head, therefore preserving important information from speaker's face.

After the model is captured, the proposed volumetric object is reconstructed reconstruction from a single RGB-D sensor as follows. First, an initial gap filling stage is performed. Then, a fast global registration algorithm [58] is used to align the pre-recorded 3D model to a single view point-cloud (see Figure 3.8). The registration process returns a transformation matrix which is then used to align the head model to the frame captured by the RGB-D camera. Finally, the aligned model is used as a basis for reconstructing the different head areas in the single view RGB-D capture. In this process, occluded parts

are filled and the rest of the head is reconstructed.

Figure 3.8: Point-cloud view of a RGB-D frame (left) and volumetric model of human head (right).

The systems assumes that (1) the back of the head of the person is non-deformable; (2) the speaker is looking ahead during most of the time, allowing the RGB-D camera to capture the mouth and eyes of the speaker; and (3) self-occlusions do not occur often, which allows the head of the speaker to be completely reconstructed. Using this framework, the higher dynamics of the object (changes in the mouth, nose, and eyes regions) can be fully represented in the reconstructed volumetric video stream. The proposed system aims to generate volumetric video frames with a minimum number of occluded areas and missing body parts.

The psedocode in Algorithm 2 describes the algorithm proposed for reconstruction process from single RGB-D capture device.

Input: volumetric pre-captured model, RGB input stream, Depth input stream
Output: recontructed volumetric point-cloud frames

$model\_pc \leftarrow pre\_captured\_model$
$model\_face\_pc \leftarrow get\_face(model\_pc)$
$model\_no\_face \leftarrow model\_pc - model\_face\_pc$
while $rgb\_d\_camera\_state = capturing$ **do**
    $color \leftarrow input\_rgb$
    $depth \leftarrow input\_depth$
    $color \leftarrow registration\_get\_color(depth, color)$
    $depth \leftarrow registration\_get\_depth(depth, color)$
    $pc \leftarrow create\_point\_cloud\_from\_rgb\_d(depth, color)$
    $face\_pc \leftarrow get\_face(pc)$
    $transform\_matrix \leftarrow fast\_pc\_registration(model\_face\_pc, face\_pc)$
    $local\_model \leftarrow transform(transform\_matrix, model\_no\_face)$
    $reconstructed\_pc \leftarrow merge\_pc(local\_model, pc)$
end

Algorithm 2: Description of the proposed volumetric reconstruction algorithm for single RGB-D capture.

The get_face() function is a fast simple algorithm which gets the point with maximum Z, which in most of the cases will be the edge of nose, and then returns the neighbors of the nose point, which includes the eyes and mouth. This function is used to further optimize the computation efficiency of the registration between the model and the input live volumetric stream. If the nose is not found using the described heuristic, the global registration with the complete model and input frame is performed, without loss of accuracy.

The registration steps in lines 7 and 8 of algorithm 2 makes the alignment between the captured RGB and Depth frames. It's interesting to note that the capture rate of the color and depth frames of the Kinect 2 device are the same, but the timestamps of the color and depth frames differ in the range of 10ms to 20ms, which is less then a frame period which is approximate 33ms at 30fps, but still makes the registration of the RGB and Depth frames important, especially for high speed movements.

The global registration method used to align the model to the input frame (line 11 of algorithm 2) is the one proposed by Zhou et at [58].

Figure 3.9, in the left, shows the extracted face of a human model (line 2 in algorithm 2), a model without the face in the center (line 3) and a single input RGB-D frame's face in the right (line 10 in algorithm 2).

Figure 3.9: Face segmented from the model (left), model without face (center) and face from single RGB-D frame.

Finally Figure 3.10 shows the reconstructed point-cloud (line 13 in algorithm 2). It's possible to notice that in the picture there is a small color temperature difference between the model (more sun was coming through the window at the moment of the capture) and the selected RGB-D frame. Also, there are scaling and blending artifacts, which can be clearly seem in figure 3.11. A blending issue is found where the hair starts in the head, caused by the under sampling near the edges of an object, and a small scaling issue is noted, caused by distances changes between the user and the camera.

Figure 3.10: Reconstructed point-cloud using the proposed reconstruction from single view RGB-D input.

Figure 3.11: Reconstructed volumetric video frame from single RGB-D sensor where it's possible to notice scaling misalignment and blending artifacts.

More specifically, our system is a computationally fast volumetric video system that reconstructs 3D representations of the speakers in real-time using a consumer CPU, with room left for optimization like GPU processing offload. The proposed method allows a better mixed reality experience when compared to incomplete and open object volumetric representation produced by using just one RGB-D capture device. Our proposal aims to recreate a complete volumetric representation of each person in the teleconference session.

Our tests show that each captured frame is processed, on average, under 33ms (the CPU used was a 16-core Intel Xeon E5-2620 at 2.1GHz). This acquisition and processing time guarantees that a 30fps input can be processed in realtime.

Similarly to methods that represent a complete human body (eg. [22]), the proposed algorithm can be extended to capture different types of objects, for which the changes occur mostly in one side/face of the object.

Currently, we are working to fix illumination, scaling and blending issues. Scaling issues are being addressed by simple object to camera distance measurement and model re-scaling, while illumination issues will be compensated by using the information from lighting changes during the communication. Finally, the blending artifacts will be removed

by a voxelization technique in development, which gives more priority to voxels from the live captured frame than a model voxel in the voxelization process. Important to note that this work was done without any explicit voxelization processing. Up to now, we assume that the density of points and the point-cloud point size representation are well balanced in order we cannot see holes in a displayed object.

One objective is to also evolve the system to create the model on-the-fly, as the user changes the pose, an internal volumetric representation of the user is constructed. Also, we intent to perform the reconstruction of the full body, instead of just the head, by using body joints information geometry.

Concerning the objective quality evaluation of our proposal, in a reconstruction process, as we don't have a reference frame, so a full-reference objective metric has no use the evaluate the proposed system. We are working on a no-reference quality evaluation metric for point-clouds to better evaluate the quality properties of this proposal.

3.4 Global Motion Estimation

Among the currently available codecs for volumetric video, there is the work proposed by Queiroz and Chou's MCIC (Motion-Compensated Intra-frame Coder) [1] which based on previous RAHT (Region-Adaptive Hierarchical Transform) [59] and Mekuria's PCC (Point-Cloud Codec) [2], both proposals submitted to ISO/IEC/MPEG point-cloud codec standardization group. Another codec is Google's Draco [60], which supports both mesh and point-cloud data structures, but uses a kD-tree structure for encoding the geometry, instead of an octree-based encoder [10], like used by the other mentioned codecs), and does not support motion compensation. Dricot et al. [11] further improve octree based geometry compression by employing octree split decisions based on a Rate-Distortion Optimization process. Our proposal consists of a motion estimation process which uses the transformation matrix result of the registration between a reference frame and a frame to be encoded. Then, the transformation matrix is encoded as a volumetric video motion frame.

It was made an evaluation of the suitability of the available codecs for use in live mixed reality experiences. We analyzed the computational efficiency and the picture quality of the compressed volumetric media, and selected an octree-based compression for the geometry and the RAHT for color coding. For an estimate of the size of an independent point-cloud frame, we used the PointCloudLibrary [61] octree-based compression for the geometry and the RAHT public implementation[2] for color coder. Octree is the structure used by most works for geometric encoding and RAHT is a state of the art point-cloud

[2]RAHT public implementation: https://github.com/digitalivp/RAHT/

color coder. As example, we selected a point-cloud with 72383 points and voxel size of $0.002^3 m^3$ (very high resolution 8i samples uses $0.001^3 m^3$) to perform an encoding and decoding process, as shown in fig. 3.12. The geometry of the point-cloud compressed with an octree-based point-cloud compression [10] presented an 4.636 bits/point (bpp) rate, and total size of 41946 bytes. The color compression of the point-cloud using RAHT provided 3.942 bpp and total size of 35672 bytes. Then the total frame of the point-cloud is 77618 bytes (41946 bytes of geometry plus 35672 bytes color), and the coding rate is 8.57 bpp.

Figure 3.12: Point-cloud before compression (left) and the resulting point-cloud after compression and decompression (right).

Considering state of the art of point-cloud inter-predictive frame coding, like [2] and [1], both proposes voxel motion estimation, and [2] also proposes to use rigid transformation (like our proposal) as optional motion estimate predictor, but our proposal applies a fast global registration method [58], instead of a local one, like the much used ICP (Interactive Closest Point) algorithm [62], and provides higher compression than voxel (or macroblocks of voxels) based motion estimation.

Our initial proposal and implementation is based in Algorithm 3, which describes a volumetric video encoding procedure where our proposal for global motion estimation is used. The encoding process consists of loop where every time a new frame is to be encoded, an objective metric (P-PSNR [45]) is used to decide if the frame will be encoded with the octree/RAHT coder or if our motion estimation should be used to encode the volumetric frame.

Our proposal performs motion estimate of whole objects, instead of individual voxels, by using a transformation matrix. We carried simulations using the 8i data-set [57], with the "soldier" volumetric video sample in order to define the P-PSNR threshold, present in Algorithm 3, line 6. We used 1 second of "soldier" 8i volumetric video, samples 730 to 759 and 36db was chosen as the P-PSRN threshold, which presents good quality balance between intra (octree + RAHT) and motion frames, while introducing minor artifacts to a human perception. The experimental data of P-PSNR calculation in the 1s volumetric video frame is shown in Table 3.1

Input: input_frame
Output: encoded_frame

```
reference_frame ← current_frame
first_encoded_frame ← octree_raht_encoder(current_frame)
while encoder_status = running do
    current_frame ← input_frame
    P_PSNR ← P_PSNR_CALC(reference_frame, current_frame)
    if P_PSNR > P_PSNR_threshold then
        encoded_frame ←
         get_transformation(reference_frame, current_frame)
    end
    else
        encoded_frame ← octree_raht_encoder(current_frame)
        reference_frame ← current_frame
    end
end
```

Algorithm 3: Proposed global motion estimation algorithm pseudo-code using P-PSNR objective metric for decision taking if a frame is encoded as a motion estimated frame or stardard octree/raht encoded frame, without other frame dependency.

Figure 3.13: A view of the volumeric video frame 750 of the "soldier" 8i data-set.

The "soldier" example shown in Figure 3.13 has 1,090,105 points and at an encoding rate of 8.57bpp (prior explained in this section), each frame will have an average of 1,168 kB. Our proposed motion estimation is very small, composed by the 4x4 transformation matrix, where each element has 4 bytes, which without compression leads to a 64 bytes of size. The 1s volumetric video example when encoded with octree/RAHT have bitrate

Table 3.1: Experimental measures of P-PSRN in the context of a video encoding pipeline. Ref. means the frames selected as reference frames (or Intra frame)

frame number	P-PSNR(reference,current) in db	encode type
730	reference frame	Intra
731	38.0498	Motion
732	35.2792	Intra (ref)
733	40.0237	Motion
734	37.3394	Motion
735	35.4315	Intra (ref)
736	40.6584	Motion
737	37.8292	Motion
738	35.9750	Intra (ref)
739	39.3950	Motion
740	36.5353	Motion
741	34.8736	Intra (ref)
742	39.6583	Motion
743	36.5525	Motion
744	34.5020	Intra (ref)
745	39.3783	Motion
746	36.1884	Motion
747	34.2878	Intra (ref)
748	38.8505	Motion
749	35.9079	Intra (ref)
750	38.9596	Motion
751	36.1311	Intra (ref)
752	40.6864	Motion
753	39.7609	Motion
754	35.3871	Intra (ref)
755	40.6864	Motion
756	38.4042	Motion
757	36.9784	Motion
758	36.0616	Motion
759	35.3823	Intra (ref)

of $1,168 * 8 * 30 = 280 Mbit/s$. In our proposal, for the same 1s, it is encoded 11 of the 30 frames as octree/RATH, and the 19 other frames are encoded as motion estimated, which results in a bitrate of $(1,168 * 8 * 11) + (64 * 19) = 104 Mbit/s$, and the rate of the bits per point is 3.18bpp. The bitrate reduction of more than 60%.

The proposed global motion estimation encoding has the limitation that it uses global motion registration, so deformable bodies are not well represented, leading to a small decrease of the smoothness of the volumetric video playback. So, the coding of residuals after the registration is being evaluated to increase the quality of important areas of each frame (like the face in a human). The coding of residuals, which are the voxels not represented (or represented in a wrong location) by the transformation, may also increase the number of motion frames, as we expect a higher P-PSNR value with the addition of the residual coding, which would lead to an improvement of quality. We are working on the segmentation of an object based of the object's moving joints to better represent movements of deformable bodies. Mekuria mentions in his point-cloud codec text [2] that rigid transform based motion estimation "is important to reduce the data volume at high capture rates" and "can typically save up to 30% of bitrate". The savings of the proposed algorithm in the example exceeds 60%.

We are working also on a methodology to evaluate the quality of the proposed motion estimation encoding in comparison to other proposals.

3.5 Adaptive point-cloud transmission method

Hosseini [60] proposes a pruning approach to sub-sample a 3D point-cloud in order to reduce the bandwidth needed to transmit this data. Park and Chou [63] proposed to subdivide a volumetric object in 3D tiles and then adjust the level of detail of each tile according to the user's distance and the region of interest. Moreno and Chen [64] adopted the traditional octree structure to compress the object geometry, which is a technique that was also used in other mentioned publications in this section, while adjusting the compression ratio of the octree in response to the network throughput changes.

We explore the possibility to do a voxel downsample, by increasing, for example, a $1^3 mm^3$ resolution to $2^3 mm^3$, $3^3 mm^3$ and so on. The downsampling is performed as a reaction to an improving or worsening condition of the network. This way it is possible to encode (or pre-encode) point-clouds using different resolutions, but without introducing holes by using a voxelization process, and in the same time prevent the a disruption to a mixed reality experience. The proposal is described in Algorithm 4. This solution is similar in essence to adaptive video streaming, like MPEG-DASH (Dynamic Adaptive

Streaming over HTTP) [65] where the player selects among different video resolutions to adapt to network conditions.

```
Data: Player status
Result: Downsampling decision taking
while transmission = occurring do
    receiving_status ← get_current_network_status();
    if playback_status() = risk_of_empty_buffer then
        decreate_pc_resolution();
    end
    else
        increate_pc_resolution();
    end
end
```

Algorithm 4: Description of the proposed network adaptive point-cloud transmission system.

In this proposal, volumetric frames are encoded or pre-encoded in different resolutions. The player informs the emitter to reduce or increase the point-cloud resolution, based on internal receiving status. If the player decides a better resolution can be tried, the emitter increases the resolution, and otherwise, if the player notice the receiver buffer is reducing with a risk of playback cuts, the resolution is decreased, like described in Algorithm 4. The player must have an internal long term network predictor to prevent the switch between two resolutions near the limit of the capacity of a network. Also, after the player reaches the highest or lower resolution, further calls to increate_pc_resolution() and decreate_pc_resolution(), respectively, have no effect.

Figure 3.14 shows the 8i sample, frame 1124 of "longdress", in resolutions of 1^3mm^3 (809,783 voxels), 2^3mm^3 (224,821 voxels), 4^3mm^3 (58,772 voxels), 8^3mm^3 (14,880 voxels) and 16^3mm^3 (3,629 voxels). The bitrate of each resolution, considering a 8.57bpp rate and 30fps, are respectively: 208 Mbit/s, 57 Mbit/s, 15 Mbit/s, 3.8 Mbit/s and 933 kbit/s.

The implementation of this proposal is in advanced stages, but still without simulation results yet. We plan to implement this algorithm integrated to the MPEG-DASH [65] streaming framework to validate our proposal. DASH is the technology used by video providers like YouTube and Netflix, and until now, there is no proposal in the literature of using DASH framework to transport volumetric point-cloud data.

We are evaluating the possibility to separately encode regions of a point-cloud, allowing DASH requests, for example, to download only regions which are in the field of view of an user. Also, a methodology to compare our proposal to other systems is being developed.

Figure 3.14: Frame 1124 of the 8i data-set labeled "longdress" in 4 different resolutions (left to the right): 1^3mm^3, 2^3mm^3, 4^3mm^3, 8^3mm^3 and 16^3mm^3.

3.6 Metric for Volumetric video quality evaluation

Albeit being a relatively new area, there are already some objective quality metrics that have been proposed to assess the quality of volumetric videos. Doumanoglou et al [43] provides a fairly complete evaluation methodology which considers the importance of the object geometry, the texture (color) resolution, and also the latency. Desai et al. [42] proposes a quality evaluation method for human bodies volumetric representations that are based on global parameters, like holes and missing parts of the human body and local factors, such the details of the face. Both work uses deep learning techniques to obtain a final quality score prediction and don't use reference frame. Torlig [46], on the other hand, proposed the use of already known 2D image metrics to be applied to 6 orthographic projections of a 3D object. Torlig's method was developed for full-reference objective quality evaluation, and among the tested PSNR (and variations), SSIM, MSSIM and VIFP metrics, MSSIM and VIFP performed best.

In order evaluate the quality achieved by the proposals in this work, we are adapting Desai's quality evaluation method to work with voxelized point-clouds, considering it was initially created for evaluating open human meshes. Desai's approach is able identify holes and artifacts annoying to a human in an open human mesh, without a reference frame. When capturing the world in 3D its difficult to have a reference volumetric frame due to the imperfections of the capture system, so an objective metric which uses a reference

(full-reference quality metric) to compute the distortion or quality level cannot be applied to evaluate a volumetric reconstruction method. Nevertheless, full-reference metrics can have usefulness in evaluating the encoding/decoding process of the volumetric content. The need to use a voxelized point-cloud, instead of a raw one, is that it does not make sense to look for a hole in a continuous 3D surface of a point-cloud, if a point has an infinitesimal dimension, while a voxel has three dimensions with an specified size. We carried some initial tests with Desai method for meshes, and now we are working to adapt it to run on voxelized point-clouds. This metric uses deep learning, and, if needed, we will re-train the network values, otherwise, we'll use the already trained neural network obtained by Desai if they prove to be good enough and matches with good confidence the subjective perception of a human.

Chapter 4

Conclusions and Future Work

In this chapter we present the conclusions already taken from the work, and further work to be done.

4.1 Current conclusions and further work

We realized that apart of many developments in the field of volumetric video and point clouds, many work needs to be done. As an example, no point-cloud format contain native support for storing the voxel dimensions, and volumetric image visualizers also have support for voxelized point cloud presentation. As symptom of the absence of such tools, the 8i document sent to the ISO/IEC/MPEG [57] point-cloud working group with examples of voxelized point-cloud bodies, showed a volumetric frame in a tool (MeshLab) that cannot correctly display a voxelized point-cloud, as shown in Figure 4.1. As part of this work, we will make an evaluation with different voxelization techniques and implement a proper player for voxelized volumetric video, so subjective analysis can be carried without undesired artifacts.

To be able to reconstruct a volume, the RGB-D capture device needs to have filters applied if we want to have a good quality with the available consumer grade equipment. The missing values for a captured depth map from a RGB device was addressed in Section 3.2, but other capture artifacts also need a solution, for example, in a scene with movement, there is an artifact presented in Figure 4.2. It's clearly possible to see colors not belonging to the object in the borders. We will work on a filter to attenuate this kind of artifact also.

In the case of our implementation of volumetric reconstruction using a single RGB-D capture device, we also realized that, apart of the registration step, an improvement related to lighting changes, scaling issues and the blending of the model with single view frame needs to be addressed. The first two problems are being addressed through an

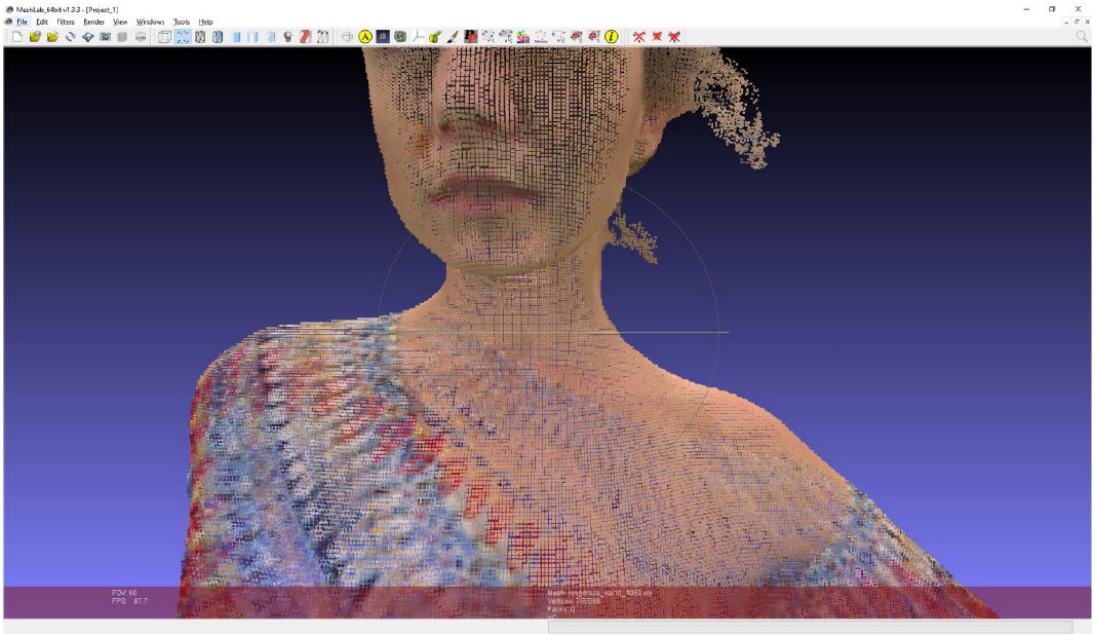

Figure 4.1: Visualization example given by 8i of a voxelized point-cloud, but clearly presenting holes.

Figure 4.2: RGB-D frame viewed as a point-cloud where a border artifact caused by Kinect's RGB and depth misalignment.

algorithm to monitor and compensate light changes, and scaling issues are being solved through to use of the distance information between an user and the sensor. The blending problems will be addressed with the development of a smart voxelization procedure which will correctly blend point clouds into one voxelized point cloud. The blending will be able

to prioritize selecting voxels from one point-cloud over another, which is important for our reconstruction method from single RGB-D input and also for any future work with multiple RGB-D capture devices working in parallel.

Concerning the global motion estimation proposal, it showed great initial results, with up to 60% of bitrate gains over the individual compression of point-cloud frames using Octree for geometry coding and RAHT for color coding, without any motion estimation. Our method for global estimation also has challenges we are working on, for example, how to represent and correctly break articulated objects in different rigid motion transformations. We will also propose a methodology on how to evaluate the quality of the created point-cloud in comparison other voxel and macro-voxel based motion estimation proposals. We will perform quality analysis in relation to same bitrate point-clouds created using state of the art motion estimation algorithms. Another important evaluation is to check the possibility to run other motion estimation implementations in real-time in consumer grade hardware.

Concerning the network adaptive point cloud streaming method, we proposed an adaptation layer which can react to network worsening or improving conditions, by decreasing or increasing the resolution of the point-cloud, respectively. This algorithm tries to provide always the best possible resolution to the user, while at the same time avoiding playback cuts. Our proposal is being integrated to a MPEG-DASH [65] transmission infrastructure, but other solutions like RTP (Real-time Transport Protocol) are not ruled out, if lowering latency becomes critical. We are also working on evaluating the overall performance and quality of the point-clouds created by our system. Finally, other types of down-sampling are being considered, which could take in consideration regions of interests or users' most common view ports.

The development of a reliable methodology to evaluate the proposed 3D reconstruction method and the motion estimation algorithm is an important part of this work. At the moment we are working on implementing and adapting the no-reference metric of Desai's [42] to run on point-cloud instead of mesh. Testing Queiroz's full-reference objective metric for point-clouds, which uses 2D image quality evaluation metrics in projected images extracted from a point-cloud or mesh, are also in consideration. The goal of testing different 2D image metrics together with the projection method is to evaluate if best performance metrics for 2D image are also good for evaluating point-clouds through some projected images. We plan to carry a subjective point-cloud quality assessment experiment with the tools which are being developed for visualization. Also a methodology for evaluating a mixed reality volumetric video presentation will be evaluated.

Among other problems to be addressed with current consumer grade hardware for properly capturing a 3D object are:

- Customer-grade RGB-D sensors cannot be controlled by an external master clock, which can cause capture misalignment and drifts;
- Internally automatically adjusted RGB camera parameters differ among the capture devices (for example, brightness can be different between cameras);
- There can be interference between sensors, since these sensors obtain the depth from a process that analyzes the feedback of an IR emission, which can be interfered by emissions from other sensors.

We consider this work will contribute to a broader adoption of volumetric video and its use in Mixed Reality experiences.

4.2 Work plan schedule

The work plan schedule is based on dead-lines, as many of the work is occuring in parallel. We base our schedule in the tasks outlined in Figure 4.3.

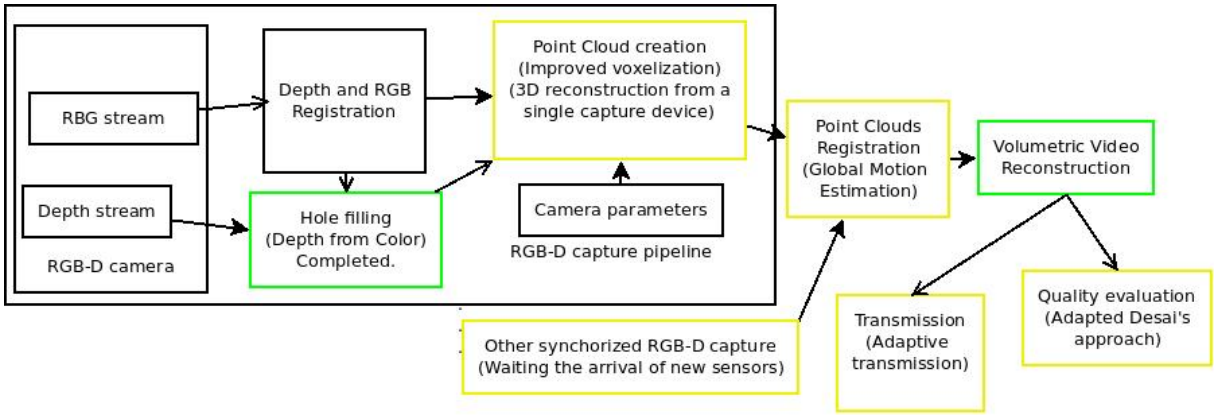

Figure 4.3: Diagram of a volumetric pipeline with the contributions proposed by this doctoral study between parenthesis.

The following deadline dates and activities to be accomplished explain our proposed work plan.

- November 2018: - Article: “Real-Time 3D volumetric human body reconstruction from single view RGB-D capture device” accepted in International Symposium on Electronic Imaging 2019;
- December 2018: Qualification exam;

- January 2019: - Initial version of the global motion estimation fully working and submission of an article about it to IEEE International Conference on Image Processing 2019;
- March 2019: Finalization of the development of voxelization techniques for voxelizing point-clouds and for merging point-clouds. An improved version of the 3D reconstruction from single view method is evaluated with the use of an initial version of the adapted metric for voxelized point-clouds. The use of the developed system in a volumetric video teleconference use case will be submitted to a journal publication;
- August 2019: Publication of an article containing all the filters developed during this work, including the already implemented method for hole filling which uses color information to recover missing depth values;
- October 2019: Tests conducted with new tools, including multi-sensors capture and Mixed Reality HMD devices for an evaluation and understanding of the outcomes of this work and improvements to the state of the art this work is contributing;
- December 2019: Finalization of the development of all the contributions, including the adaptive point-cloud transmission method for packet-switched networks;
- January 2020: Submission of the thesis;
- February 2020: Doctoral degree final exam.

References

[1] Queiroz, Ricardo L de and Philip A Chou: *Motion-compensated compression of dynamic voxelized point clouds.* IEEE Transactions on Image Processing, 26(8):3886–3895, 2017. vi, 5, 6, 15, 18, 22, 23

[2] Mekuria, Rufael, Kees Blom, and Pablo Cesar: *Design, implementation, and evaluation of a point cloud codec for tele-immersive video.* IEEE Transactions on Circuits and Systems for Video Technology, 27(4):828–842, 2017. vi, 5, 6, 15, 18, 22, 23, 26

[3] Milgram, Paul and Fumio Kishino: *A taxonomy of mixed reality visual displays.* IEICE TRANSACTIONS on Information and Systems, 77(12):1321–1329, 1994. 1

[4] Shi, Liang, Fu Chung Huang, Ward Lopes, Wojciech Matusik, and David Luebke: *Near-eye light field holographic rendering with spherical waves for wide field of view interactive 3d computer graphics.* ACM Transactions on Graphics (TOG), 36(6):236, 2017. 2

[5] Jang, Changwon, Kiseung Bang, Seokil Moon, Jonghyun Kim, Seungjae Lee, and Byoungho Lee: *Retinal 3d: augmented reality near-eye display via pupil-tracked light field projection on retina.* ACM Transactions on Graphics (TOG), 36(6):190, 2017. 2

[6] Orts-Escolano, Sergio, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, *et al.*: *Holoportation: Virtual 3d teleportation in real-time.* In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 741–754. ACM, 2016. 3

[7] Schwarz, Sebastian, Miska M Hannuksela, Vida Fakour-Sevom, and Nahid Sheikhi-Pour: *2d video coding of volumetric video data.* In *2018 Picture Coding Symposium (PCS)*, pages 61–65. IEEE, 2018. 4, 6

[8] Sugimoto, Kazuo, Robert A Cohen, Dong Tian, and Anthony Vetro: *Trends in efficient representation of 3d point clouds.* In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017*, pages 364–369. IEEE, 2017. 5

[9] Hinks, Tommy, Hamish Carr, Linh Truong-Hong, and Debra F Laefer: *Point cloud data conversion into solid models via point-based voxelization.* Journal of Surveying Engineering, 139(2):72–83, 2012. 5

[10] Schnabel, Ruwen and Reinhard Klein: *Octree-based point-cloud compression.* Spbg, 6:111–120, 2006. 6, 22, 23

[11] Dricot, Antoine, Fernando Pereira, and João Ascenso: *Rate-distortion driven adaptive partitioning for octree-based point cloud geometry coding.* In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2969–2973. IEEE, 2018. 6, 22

[12] Merry, Bruce, Patrick Marais, and James Gain: *Compression of dense and regular point clouds.* In *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 15–20. ACM, 2006. 6

[13] Shao, Yiting, Zhaobin Zhang, Zhu Li, Kui Fan, and Ge Li: *Attribute compression of 3d point clouds using laplacian sparsity optimized graph transform.* arXiv preprint arXiv:1710.03532, 2017. 6

[14] Cohen, Robert A, Dong Tian, and Anthony Vetro: *Attribute compression for sparse point clouds using graph transforms.* In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1374–1378. IEEE, 2016. 6

[15] Golla, Tim and Reinhard Klein: *Real-time point cloud compression.* In *IROS*, pages 5087–5092, 2015. 6

[16] MPEG-3DG: *Pcc test model category 2 v1.* ISO/IEC/JTC1/SC29/WG11 MPEG N17348, page n17348, 2018. 6

[17] MPEG-3DG: *Pcc test model category 3 v1.* ISO/IEC/JTC1/SC29/WG11 MPEG N17349, page n17348, 2018. 6

[18] Kowalski, Marek, Jacek Naruniec, and Michal Daniluk: *Live scan3d: A fast and inexpensive 3d data acquisition system for multiple kinect v2 sensors.* In *3D Vision (3DV), 2015 International Conference on*, pages 318–325. IEEE, 2015. 6, 10

[19] Seitz, Steven M, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski: *A comparison and evaluation of multi-view stereo reconstruction algorithms.* In *null*, pages 519–528. IEEE, 2006. 6

[20] Berger, Matthew, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva: *A survey of surface reconstruction from point clouds.* In *Computer Graphics Forum*, volume 36, pages 301–329. Wiley Online Library, 2017. 6, 7

[21] Firman, Michael, Oisin Mac Aodha, Simon Julier, and Gabriel J. Brostow: *Structured prediction of unobserved voxels from a single depth image.* In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 7

[22] Alexiadis, Dimitrios S, Nikolaos Zioulis, Dimitrios Zarpalas, and Petros Daras: *Fast deformable model-based human performance capture and fvv using consumer-grade rgb-d sensors.* Pattern Recognition, 79:260–278, 2018. 7, 21

[23] Bondi, Enrico, Pietro Pala, Stefano Berretti, and Alberto Del Bimbo: *Reconstructing high-resolution face models from kinect depth sequences.* IEEE Transactions on Information Forensics and Security, 11(12):2843–2853, 2016. 7

[24] Choi, Jongmoo, Gerard Medioni, Yuping Lin, Luciano Silva, Olga Regina, Mauricio Pamplona, and Timothy C Faltemier: *3d face reconstruction using a single or multiple views.* In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3959–3962. IEEE, 2010. 7

[25] Jiang, Luo, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu: *3d face reconstruction with geometry details from a single image.* IEEE Transactions on Image Processing, 27(10):4756–4770, 2018. 7

[26] Chang, Angel X, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, *et al.*: *Shapenet: An information-rich 3d model repository.* arXiv preprint arXiv:1512.03012, 2015. 7

[27] Thanh Nguyen, Duc, Binh Son Hua, Khoi Tran, Quang Hieu Pham, and Sai Kit Yeung: *A field model for repairing 3d shapes.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5676–5684, 2016. 7

[28] Rock, Jason, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem: *Completing 3d object shape from one depth image.* In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2484–2493, 2015. 7, 17

[29] Song, Shuran, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser: *Semantic scene completion from a single depth image.* In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 190–198. IEEE, 2017. 7, 17

[30] Hernandez, Matthias, Jongmoo Choi, and Gérard Medioni: *Near laser-scan quality 3-d face reconstruction from a low-quality depth stream.* Image and Vision Computing, 36:61–69, 2015. 7

[31] Newcombe, Richard A, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon: *Kinectfusion: Real-time dense surface mapping and tracking.* In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 7, 14, 17

[32] Mendes, Paulo Renato Conceição, Roberto Gerson de Albuquerque Azevedo, Ruy Guilherme Silva Gomes de Oliveira, and Carlos de Salles Soares Neto: *Exploring an ar-based user interface for authoring multimedia presentations.* In *Proceedings of the ACM Symposium on Document Engineering 2018*, page 9. ACM, 2018. 7

[33] Salvi, Joaquim, Sergio Fernandez, Tomislav Pribanic, and Xavier Llado: *A state of the art in structured light patterns for surface profilometry.* Pattern recognition, 43(8):2666–2680, 2010. 8

[34] Lange, Robert and Peter Seitz: *Solid-state time-of-flight range camera.* IEEE Journal of quantum electronics, 37(3):390–397, 2001. 8

[35] Zhang, Zhengyou: *Microsoft kinect sensor and its effect.* IEEE multimedia, 19(2):4–10, 2012. 8

[36] Pagliari, Diana and Livio Pinto: *Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors.* Sensors, 15(11):27569–27589, 2015. 9

[37] Schöning, Julius and Gunther Heidemann: *Taxonomy of 3d sensors.* Argos, 3:P100, 2016. 9, 10

[38] Stoyanov, Todor, Athanasia Louloudi, Henrik Andreasson, and Achim J Lilienthal: *Comparative evaluation of range sensor accuracy in indoor environments.* In *5th European Conference on Mobile Robots, ECMR 2011, September 7-9, 2011, Örebro, Sweden*, pages 19–24, 2011. 9

[39] Achar, Supreeth, Joseph R Bartels, William L Whittaker, Kiriakos N Kutulakos, and Srinivasa G Narasimhan: *Epipolar time-of-flight imaging.* ACM Transactions on Graphics (ToG), 36(4):37, 2017. 9

[40] Guidi, G, S Gonizzi, L Micoli, *et al.*: *3d capturing performances of low-cost range sensors for mass-market applications.* INTERNATIONAL ARCHIVES OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES, pages 33–40, 2016. 10

[41] Lan, Gongjin, Ziyun Luo, and Qi Hao: *Development of a virtual reality teleconference system using distributed depth sensors.* In *Computer and Communications (ICCC), 2016 2nd IEEE International Conference on*, pages 975–978. IEEE, 2016. 10

[42] Desai, Kevin, Kanchan Bahirat, and Balakrishnan Prabhakaran: *Learning-based objective evaluation of 3d human open meshes.* In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 733–738. IEEE, 2017. 10, 28, 32

[43] Doumanoglou, Alexandros, David Griffin, Javier Serrano, Nikolaos Zioulis, Truong Khoa Phan, David Jiménez, Dimitrios Zarpalas, Federico Alvarez, Miguel Rio, and Petros Daras: *Quality of experience for 3-d immersive media streaming.* IEEE Transactions on Broadcasting, 64(2):379–391, 2018. 10, 28

[44] Tian, Dong, Hideaki Ochimizu, Chen Feng, Robert Cohen, and Anthony Vetro: *Geometric distortion metrics for point cloud compression.* In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3460–3464. IEEE, 2017. 10

[45] De Queiroz, RL, E Torlig, and TA Fonseca: *Objective metrics and subjective tests for quality evaluation of point clouds.* ISO/IEC JTC1/SC29/WG1 input document M78030, 2018. 10, 23

[46] Torlig, Eric M, Evangelos Alexiou, Tiago A Fonseca, Ricardo L de Queiroz, and Touradj Ebrahimi: *A novel methodology for quality assessment of voxelized point clouds*. In *Applications of Digital Image Processing XLI*, volume 10752, page 107520I. International Society for Optics and Photonics, 2018. 10, 28

[47] Alexiou, Evangelos, Marco V Bernardo, Luis A da Silva Cruz, Lovorka Gotal Dmitrovic, Carlos Duarte, Emil Dumic, Touradj Ebrahimi, Dragan Matkovic, Manuela Pereira, Antonio Pinheiro, *et al.*: *Point cloud subjective evaluation methodology based on 2d rendering*. In *10th International Conference on Quality of Multimedia Experience (QoMEX)*, number CONF, 2018. 11

[48] Zhang, Juan, Wenbin Huang, Xiaoqiang Zhu, and Jenq Neng Hwang: *A subjective quality evaluation for 3d point cloud models*. In *Audio, Language and Image Processing (ICALIP), 2014 International Conference on*, pages 827–831. IEEE, 2014. 11

[49] Alexiou, Evangelos and Touradj Ebrahimi: *On the performance of metrics to predict quality in point cloud representations*. In *Applications of Digital Image Processing XL*, volume 10396, page 103961H. International Society for Optics and Photonics, 2017. 11

[50] Javaheri, Alireza, Catarina Brites, Fernando Pereira, and João Ascenso: *Subjective and objective quality evaluation of 3d point cloud denoising algorithms*. In *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017. 11

[51] Blake, J, F Echtler, and C Kerl: *Openkinect: Open source drivers for the kinect for windows v2 device*, 2015. 12

[52] Zhou, Qian Yi, Jaesik Park, and Vladlen Koltun: *Open3D: A modern library for 3D data processing*. arXiv:1801.09847, 2018. 12

[53] Rao, Rajendran, Abraham Konda, David Opitz, and Stuart Blundell: *Ground surface extraction from side-scan (vehicular) lidar*. In *Proc. MAPPS/ASPRS Fall Conference, San Antonio, USA*, 2006. 14

[54] Beck, Stephan and Bernd Froehlich: *Volumetric calibration and registration of multiple rgbd-sensors into a joint coordinate system*. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*, pages 89–96. IEEE, 2015. 14

[55] Curless, Brian and Marc Levoy: *A volumetric method for building complex models from range images*. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 16

[56] Yang, Bo, Stefano Rosa, Andrew Markham, Niki Trigoni, and Hongkai Wen: *3d object dense reconstruction from a single depth view*. arXiv preprint arXiv:1802.00411, 2018. 17

[57] d'Eon, E, B Harrison, T Myers, and PA Chou: *8i voxelized full bodies, version 2–a voxelized point cloud dataset*. document MPEG, page m74006, 2017. 18, 23, 30

[58] Zhou, Qian Yi, Jaesik Park, and Vladlen Koltun: *Fast global registration.* In *European Conference on Computer Vision*, pages 766–782. Springer, 2016. 18, 20, 23

[59] Queiroz, Ricardo L de and Philip A Chou: *Compression of 3d point clouds using a region-adaptive hierarchical transform.* IEEE Transactions on Image Processing, 25(8):3947–3956, 2016. 22

[60] Hosseini, Mohammad and Christian Timmerer: *Dynamic adaptive point cloud streaming.* arXiv preprint arXiv:1804.10878, 2018. 22, 26

[61] Rusu, Radu Bogdan and Steve Cousins: *3d is here: Point cloud library (pcl).* In *Robotics and automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE, 2011. 22

[62] Rusinkiewicz, Szymon and Marc Levoy: *Efficient variants of the icp algorithm.* In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001. 23

[63] Park, Jounsup, Philip A Chou, and Jenq Neng Hwang: *Rate-utility optimized streaming of volumetric media for augmented reality.* arXiv preprint arXiv:1804.09864, 2018. 26

[64] Moreno, Carlos, Yilin Chen, and Ming Li: *A dynamic compression technique for streaming kinect-based point cloud data.* In *Computing, Networking and Communications (ICNC), 2017 International Conference on*, pages 550–555. IEEE, 2017. 26

[65] Sodagar, Iraj: *The mpeg-dash standard for multimedia streaming over the internet.* IEEE MultiMedia, (4):62–67, 2011. 27, 32

Supplement I

Article Reconstruction

Abstract of the accepted article in the International Symposium on Electronic Imaging 2019.

Recently, volumetric video based communications have gained a lot of attention, especially due to the emergence of devices that can capture scenes with 3D spatial information and display mixed reality environments. Nevertheless, capturing the world in 3D is not an easy task, with capture systems being usually composed by arrays of image sensors, sometimes paired with depth sensors. Unfortunately, these arrays are not easy assembly and calibrate to non-specialists for use, making their use in volumetric video applications a challenge. Additionally, the cost of these systems is still high, which limits their popularity in more mainstream communication applications. This work proposes a system that provides a way to reconstruct the upper body of a human speaker from single view frames captured using a single RGB-D camera (a Microsoft's Kinect device). The proposed system generates volumetric video frames with a minimum number of occluded areas and missing body parts. To achieve a good quality, the system prioritizes the data corresponding to the participants' head, therefore preserving important information from speaker's face. We plan to compare the quality of the proposed system with previous systems. Our ultimate goal is to design an inexpensive system that can be used in 3D telepresence conference applications and even on volumetric video talk-show broadcasting application.