Université de Nantes

Rapport annuel d'avancement des travaux en vue de l'obtention

d'autorisation de réinscription en thèse dans

L'école Doctorale : Math Sciences et technologies de l'information et de la communication

Discipline : Informatique et Applications
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes (LS2N)

Efficient Representation of Discrete Dynamic Contents for Mixed reality

By

Amira Filali

Comité de suivi :	M. Olivier Deforges
	M. David Coeurjolly
Directeur de thèse :	M. Nicolas Normand
Co-encadrant de thèse :	M. Vincent Ricordel

Contents

List of Figures

1 Introduction

This first chapter introduces the main topic of this thesis. The chapter is organized in two different sections: Section 1.1 establishes the general context for the research activity in this thesis. Section 1.2 introduces the key problem that is addressed in the research project.

1.1 General Context

Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) are announced as the next digital technologies that will revolutionize our modes of communication [3, 10]. With the increased proliferation of applications using 3D technologies and high definition objects such as VR, AR, 360 videos, real-time 3D immersive tele-presence, representing these kinds of data has been the focus of computer vision for decades and a popular challenge for processing, storing and conveying the data independently of how it was captured. This is therefore at the heart of several researches, and in particular those in video visualization and compression due to the large amounts of data generated from the 3D measurement systems to improve the data processing, transmission and archiving.

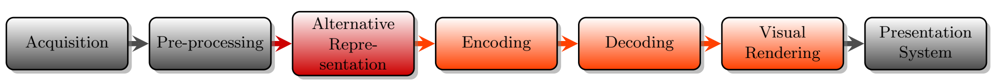

Figure 1.1: 3D Data Conceptual Workflow

Indeed, if on one side of the communication chain, the contents emerge with 360 acquisition sensors which are in continuous evolution (e.g. Nokia Ozo camera, Facebook Surround, studio capture). It is considered here also the case where the data is computer generated and not acquired from a real physical scene. On the other side of the chain, extracting the appropriate data for the best quality of experience achievable with visual rendering systems which are numbered with the Head Mounted Display HMD (e.g. Oculus Rift, HTC Vive, Playstation VR, HoloLens). Post-processing step to improve visual quality and aesthetics, e.g. filtering, contrast enhancement, etc. could be always applied but were not mandatory.

At the heart of the chain remains the need to find an efficient representation of content. The acquired/created data format may not be inevitably the best format for the data representation when considering the overall application requirements, e.g. compression, user interaction, etc. Thus the conversion of the sensed data into a more appropriate representation model/format, e.g. multiple video views and depth data, or point clouds with RGB attributes, is needed in order to better compress them while preserving their characteristics and the immersion experiences [16].

Then, the encoder targets the efficient coding of the data at hand referred to as compression efficiency. Compression efficiency means usage of the smallest number of bits for the target quality while fulfilling an additional set of identified requirements. It is often considered as the critical functionality. This encoding process may be either lossless or lossy although it is more commonly lossy. If lossy, it is typically important to consider the human perception characteristics. The decoder is a natural counterpart of the encoder and should recover the data in the adopted representation format after storage or transmission. Depending on the specific lossy encoder, the data itself and the rate used, the decoded data will have a certain amount of distortion with regard to the corresponding original data. Finally, it is important to highlight the importance of 'quality metrics' already in this context to assess the rate-distortion performance of each encoder and also compare performance. The overview of this dissertation is given in Figure 1.1.

1.2 Problem Statement

Advances in technology, 360-degree video and virtual/augmented reality field, provide many fast and reliable cameras, laser scanners, etc. which are rapidly spreading as devices for capturing the 3D data in different ways providing high definition 3D content. For instance the Velodyne HDL-64E that can capture 3D data using point data at high rates of up to 1.5 million points per second. The increased quality and quantity of the captured data with additional reflectance and color produces very dense points.

On one hand, from the source data to the diverse platform (mobile, computer, etc.), multiple levels of definition are required to allow lower processing requirement and lower storage needs. For many applications, it is desirable to transmit this data via restricted bandwidth networks. Hence, reducing this data as far as possible is necessary. Multiple methods of compression have been developed and used in research and commercial projects. Each compression process has the best utility depending on the source data and the usage (rendering, animation, performance).

On the other hand, in computer graphics field, generated/acquired real-world scene are generally defined or represented by geometric objects using different modeling methods; grids, polygons meshes, curves, point cloud, etc providing intricate details. Besides, almost of compression techniques are based on modeling first the 3D data to yield an efficient "alternative geometric representation" of the discrete content. While geometry processing, or discrete data processing is an area of research that uses concepts from applied mathematics and computer science to design efficient algorithms for the acquisition, reconstruction, analysis, compression, rendering and visualization of complex 3D models.

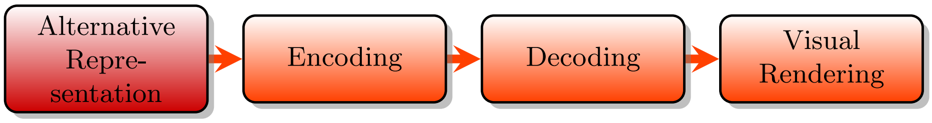

Figure 1.2: Steps from Fig.1.1 that address this thesis objectives

We will be focused in this PhD particularly by the approaches where the data of the 3D kinematic contents are volumetric. In our future work, we will be concerned by their alternative representations which are typically done by dynamic mesh, or by dynamic point cloud, or by graphs. These volumetric representations are heavy and characterized by geometry informations, topology and other attributes such as normals, colors, transparency etc., rather than those of the classic 3D images and videos. So that MPEG compression standards cannot be applied, new compression approaches (based on R-Tree, OcTree, BSP Tree, Graph-Transform, etc.) are therefore necessary, and it is necessary too to adapt the processing models, for example to estimate/compensate the motion, to optimize the rate/distortion compromise, to evaluate the distortions, or to define scalable flows. Our research may also focus on redefining and using of conventional signal processing operations (wavelet transform, frequency analysis, filtering, interpolation, etc.) to apply them to these dynamic volumetric representations where information can be modeled as meshes, graphs or point clouds. There is therefore a wide scope for discrete geometry. The substantial steps that deal with this PhD objectives are shown in Figure 1.2.

2 State of The Art

This chapter elaborates in Section 2.1 an overview of existing representation models traditionally applied on 3D data. Section 2.2 largely introduces the state-of-the-art approaches proposed by the community of signal analysis and visualization researchers.

2.1 Towards Efficient Discrete Representations

Advanced 3D representations of the world are enabling more immersive forms of interaction and communication, and also allow machines to understand, interpret and navigate our world.

2.1.1 Trends in 3D representation

3D modeling is a technique in computer graphics for producing a 3D digital representation of any object or surface. These 3D objects can be either generated automatically or created manually. In this section different data representation are presented.

2.1.1.1 Light Field

A light field is a representation of all light rays in space. It is the totality of photons (light particles) traveling in all directions throughout all points in space in a given area. Specifics within the light field can be described according to the plenoptic function L(x,y,z,θ, ϕ) introduced first in [1], which can identify any given point where light arrives in a scene through five dimensions (5D) as shown in 2.1: three to determine the position of the point and two more to identify the angle by which the light arrives at that point. So, this plenoptic function defines the amount of light flowing in every direction through every point.

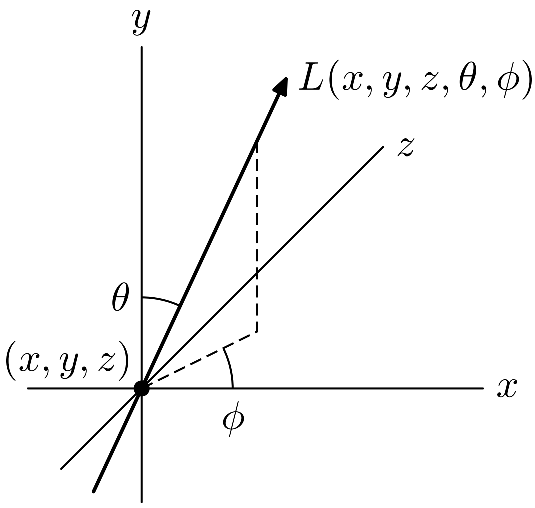

Figure 2.1: Parameterizing a ray in 3D space by position (x, y, z) and direction (θ, ϕ)[17]

Inspired by flies' eyes, light fields are designed as an array of images that captured at constant offsets or rotations of lens (see Figure 2.2). Once light fields are acquired, emitted radiance is recorded in 2D images for predefined view points and projection directions. Radiance between these view directions can be reconstructed by interpolation among neighboring images producing colored image. The reconstruction quality depends on the resolution of the images as well as the number of images.

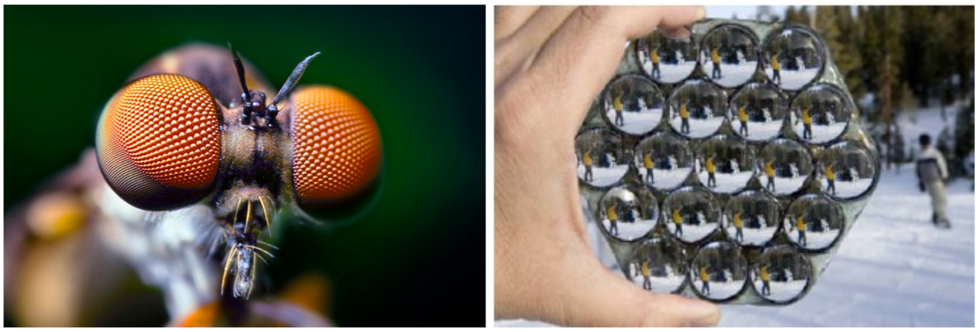

Figure 2.2: Fly eyes (by Thomas Shahan) and LightField Camera (Adobe Magic Lens)

Although the capture of a complete light field is not possible, a light field can be sampled sparsely, enabling the modeling of a virtual environment. This model can be used to reproduce views or to combine the data with computer generated data. The capturing of light with sensors typically follows similar principles of the human eyes. This increases the dimensions of information additionally. Each camera records the color information, either as RGB or in a luminance/chrominance format, associated with a particular viewpoint. The data captured by a camera may also be referred to as the texture of the scene [21]. Moreover, rather than a computation power hungry problem, rendering using light fields are more likely a bandwidth hungry problem that happens on data transmission between different computation units. This is one of the main problem that one can face to in the light field analysis and processing.

2.1.1.2 Mesh

A polygonal mesh is a piecewise linear surface approximation of a real world object in 3D computer graphics and geometric modeling. A 3D polygonal mesh is a collection of various elements, which are named edges, faces and vertices depending on reference points in x, y and z axes that defines the shape with height, width and depth of a polyhedral object [4]. In other words, a 3-D mesh is actually a hierarchical assembly of different elements represented in figure 2.3.

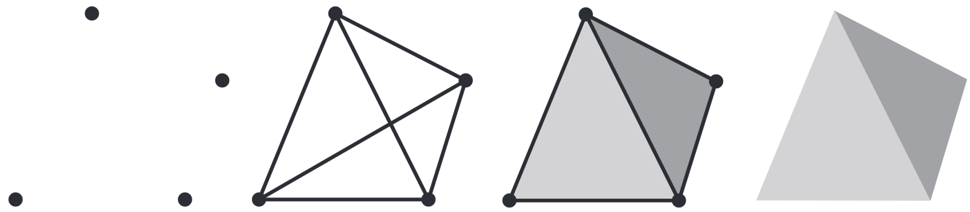

Figure 2.3: Element of a mesh: vertices, vertices connected 2 by 2 through edges, faces, the interior of the surface defined by faces[2]

The faces usually consist of triangles (triangle mesh), quadrilaterals, or other simple convex polygons, since this simplifies rendering, but may also be composed of more general concave polygons, or polygons with holes.

3D mesh provides enhanced capabilities for modeling object shapes in a more detailed way. Different representations of polygon meshes are used for different applications and goals. Most computer graphics applications (animated movies, games, augmented reality, etc.) use surface meshes, since the insides of an object are rarely visible, while scientific simulations may use surface or volume meshes depending on the setting.

2.1.1.3 Point Cloud

3D point cloud is a de facto standard for Computer Graphic and 3D modeling. Modern 3D laser scanning systems broaden the usability and access to 3D measurement systems and simplify the collection of large 3D point clouds. These 3D laser scanning systems are used to generate accurate and high resolution 3D point clouds. They have improved the quality and quantity of the captured point clouds. Scanners are able to acquire up to a million of points per second to represent the environment with a dense point cloud. Additionally, modern scanners are capable of capturing the reflectance and color values of the measured surfaces. This represents the captured environment with a very high degree of detail. Thus the environment is represented more realistically (see Figure 2.5).It is considered here also the case where the data is completely computer generated and not acquired from a physical scene as it is shown in 2.4.

Technically, a point cloud consists of a set of data points in a coordinate system indicating the location of each point, in 3D applications this coordinate system is designated by X, Y, and Z coordinates and shows the external surface of an object represented as individual points, along with one or more attributes such as color, normal, etc. associated with each point [32].

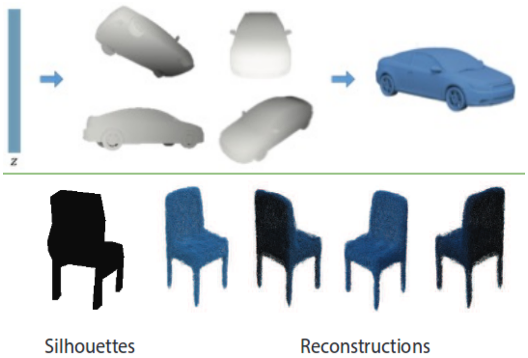

Figure 2.4: Generating or reconstructing 3D shapes from single or multi-view depth maps or silhouettes and visualizing them in dense point clouds[30]

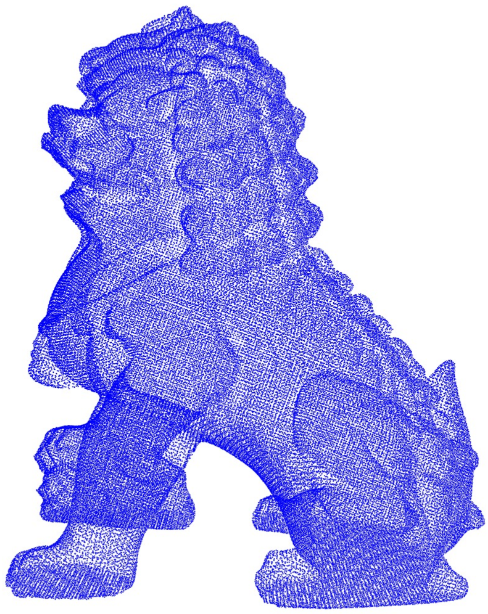

Figure 2.5: Lidar acquired point cloud[17]

3D point clouds can be captured using multiple cameras and depth sensors in various setups and may be made up of thousands up to billions of points in order to represent fully immersively reconstructed objects or scenes, thus efficient representation of point clouds are needed to store or transmit these information.

Compared with image compression, 3D point cloud compression is much more difficult, because each point is basically not related each other, e.g., no orders and no local topology exists. In addition to this, each point may have not only a 3D position information (x, y, z) but also a color information (R, G, B) and possibly other attributes such like transparency, time of acquisition, reflectance of laser or material property, etc.

What information represent the point cloud?

1. Vertices of mesh

   The points of cloud, in this case, represent the vertices of the mesh defining the shape of a 3D object. This type of data is almost computer-generated using specific software.

   There are many geometry file formats, e.g., .OBJ format where its most common elements are geometric vertices, texture coordinates, vertex normals and polygonal faces.

2. Points on surface

   The point cloud represent captured points using specific scanner defining the surface of the 3D object such as LiDAR laser sensor. It is saved in form of a very large number of points that cover surfaces of the sensed object. Points in a point clouds are always located on the external surfaces of visible objects, because this are the spots, where ray of light from the scanner reflected from an objects. This data storage format mostly supports a relatively simple description of the 3D object: geometric position in three-dimensional coordinate system, color attributes.

How to compute the normals?

Surface normals are important characteristic properties of a geometric surface, and are heavily used in many fields such as computer graphics applications:

- apply the correct light sources that generate visual effects

- change the apparent lighting of rendered elements
- In this thesis, the surface normal would be used to improve the computation of the distortion which is employed in the TLSVQ method [13]; point to plane metric calculation instead of point to point.

Whether we have surface normals information (.obj files) in the point cloud file or not (almost of .ply, .xyz etc. files), there are many methods to extract or calculate the normals of vertices or surfaces.

1. If Point clouds are Mesh vertices:

   a) Delaunay Triangulation could be applied to obtain the 3D meshes from point clouds, then surface normals are computed.

   b) It could be computed as the normalized average of the surface normals of the faces that contain that vertex [12].

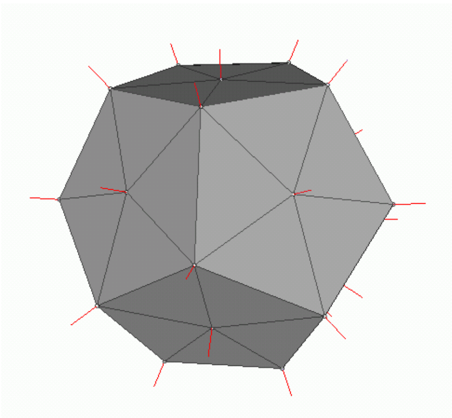

Figure 2.6: Vertex normals of a 3D dodecahedral mesh [Sandberg2012]

2. If Point clouds are points on surface:
   Since the point cloud datasets that we acquire represent a set of point samples on the real surface, there are two possibilities:

   a) Obtain the underlying surface from the acquired point cloud dataset, using surface meshing techniques, and then compute the surface normals from the mesh.

b) Use approximations to infer the surface normals from the point cloud dataset directly as the vector perpendicular to the surface in that point.

c) Perform the estimation of the surface normal by PCA –Principal Component Analysis- of a covariance matrix C created from the nearest neighbors of the query point. The problem of determining the normal to a point on the surface is approximated by the problem of estimating the normal of a plane tangent to the surface

  - we need to establish relations between point and point to obtain point feature information. Then the first step is to find the neighbour of each point data. There are many common methods to find the nearest neighbors of the query point:

    - K-nearest neighbors method: It constructs the k-neighbour field by searching the k closest points using the Euclidean distance. This method is an adaptive field estimation method, which determines the k-neighbour field by searching the k closest points of the given point. However, it, in the case of variable dense of original point cloud, causes large differences of the range of the obtained k-neighbour field, and leads to large error of normal vector estimation.

    - K-D tree method: Complex/slow method for such purpose.
      The nearest neighbour may not be in the right leaf.

    - Minimum bounding ball method (ball with minimum radius r): For the constant density of point cloud, the efficiency of the method, which adopts the minimum radius, is high. Otherwise, when the difference of density is large, the method is inefficient.

So we have the idea to adapt the radius for every point normal computation depending on the density criterion.

2.1.2 Graph

A graph, a term in math, is a pair G= (V,E) where V is a set of nodes (vertices) representing the number of components in the system, and E is a finite set of links or ordered relation pairs of distinct nodes representing the total number of interactions between the nodes as can be seen in figure 2.7 below. In most contexts, graphs permit the representation of natural relations between things and chains of these relations [5].

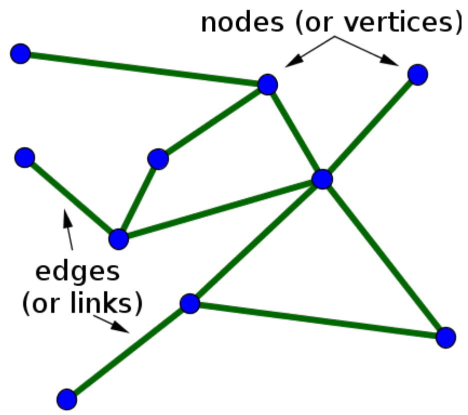

Figure 2.7: Graph with 10 nodes (or vertices) and 11 edges (or links)[11]

Graphs are very attractive when it comes to modeling real-world data, because they are intuitive, flexible, and because the theory supporting them has been maturing for centuries. As a consequence, there are several graph databases available. The same goes for graph processing, algorithms are numerous and well understood and are increasingly used in a variety of revenue-generating applications, such as social applications, online retail, business intelligence and logistics, single-source shortest path, route finding, clustering, dimensionality reduction, sampling graphs, etc. to name a few.

How to construct a good graph describing the relationships between samples has been widely studied in recent years, and it is still an open problem. The quality of graphs is very sensible to the topological structure, the choice of weighting functions and the related parameters. However, the diversity of the available graphs, the processing algorithms, and of the graph-processing platforms currently available to analysts makes the selection an important challenge. More problems arise when processing very large graphs, when visiting millions of highly connected vertices.

2.1.3 Summary

Unlike classic 3D Data defined with pixels intensities, the 3D discrete objects or data contain a list of geometry component rather than only appearance component. The geometry defines the structure of the 3D data. This mathematical component concerned with questions of shape, size, relative position of objects, and the properties of space. It tells us where the object is in space (3D coordinates), how elements of a set relate spatially to each other (topology, e.g., meshes) and how to determine a surface's orientation toward a light source (normals variation).
The other part of every 3D object is the surface description. This boils down to describing the physics of how the surface of the model interacts with light. The appearance specifies various descriptive attributes that have intuitive meaning, including color, texture, transparency and so on.
There are diversities of available models which can be employed to represent efficiently 3D discrete content according to the authors, the context, the needs of each application and the expected data.
It is worth taking into consideration that we can pass from presentation type to another using various existing techniques; for instance 3D meshes add connectivity between the points of the point cloud, which then become the vertices of triangular

patches - the 3D mesh - describing the object's shape (see Figure 2.8) and the same for graph construction using specific methods, such as in [33], to extract graph nodes from point cloud as shown in Figure 2.9.

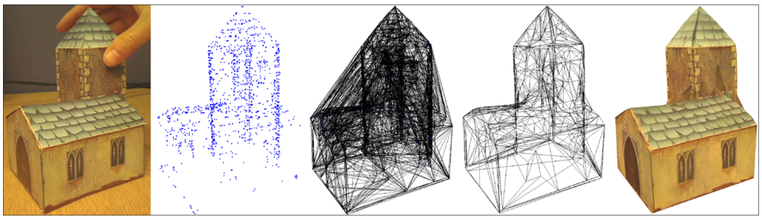

Figure 2.8: From 3D points to 3D meshes [24]

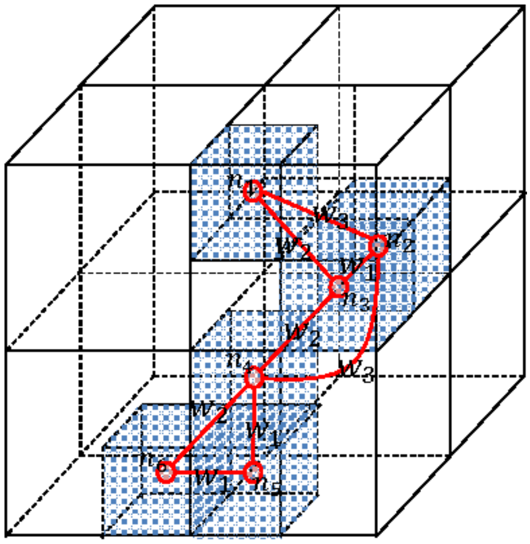

Figure 2.9: From 3D points to graph[33]

We will firstly deal with point cloud data in this PhD for many justifiable reasons. By doing so, 3D point cloud particularly provides a compact and scalable representation of the immersive contents. Moreover, point cloud data directly provides 3D geometry information, while additional processing is required to obtain this information from traditional image data. The 3D point cloud representation model is, in addition, well adapted to our need and targets (Data Compression) as well we can adapt existing tools and operators to this model. 3D point cloud is a flexible and compatible representation that guarantees the easy conversion from points to another representation models (e.g, meshes, graphs, etc.)
Nevertheless, there are several technical challenges that need to be addressed in order to achieve the objectives of this PhD.

2.2 Related Work: Trends in 3D Discrete Data Encoding

Various researchers have studied the potential of representation of dynamic and volumetric 3D data for the purpose of its compression over the years.
Compression is crucial for many applications that require storage and transmission of 3D data through a network with a limited bandwidth. The compression of 3D data has been pursued mostly in the context of compression of meshes [25]. However, point clouds have emerged as a promising solution for immersive representation of 3D contents. The representation of scenes as 3D point clouds recently has been increasing in recent years and it seems to be more computationally efficient than meshes [28].
Several methods for 3D point cloud compression have been implemented, e.g., [27]. The objective of this section is to study the state-of-the-art encoding methods of point clouds.

2.2.1 Octree-based Encoding

The first compression scheme is rather popular nowadays and it is based on the organization of all points in an octree structure to compress 3D data. In an octree, each node is referred as voxel and represents a cube in 3D space. The root node is a cube that contains all points of the point cloud (also referred as bounding box). Then, starting from the root node, this cube is divided into 8 cubes with the same size which correspond to every root node children (connected by tree branches). This process is repeated for every tree child nodes iteratively which are again divided into 8 nodes; all the points are organized into the voxel associated to each tree node during this recursive process. Naturally, a tree node is not divided if there is no point occupying the corresponding voxel.
The octree structure provides a multi-resolution spectral decomposition of the 3-D

point cloud. Unlike image/video processing in which all points in a 2-D or 3-D block correspond to a pixel position, our 3-D blocks are not necessarily fully occupied by points, and this approach is specially effective on large free space areas.

Octree data structure for 3-D point cloud compression is explored by Kammerl et al. in [15] to propose a modified octree data structure (double buffering octree) that enables detection and differential encoding of spatial changes within temporarily adjacent point clouds. By differentially encoding changes within octree data structures, the authors exploited spatial and temporal redundancies between consecutive point clouds by initially performing a full compression once for the first point cloud, and subsequently only encode extracted changes within the point distribution. In other words, initial to every incoming point cloud, the root node of the currently assigned pointer buffer is initialized with zeros. From then on, the octree structure is recursively built according to the incoming point set. Whenever new child nodes need to be created, the proposed method performs a lookup to the preceding pointer buffer within the current branch node. Once a new octree structure is built, the octree is then traversed as in the single frame case. However, in order to obtain and output the structural changes for every branch, an exclusive disjunction operation (XOR) is applied on the two bytes representing the child node configurations of the child pointer buffers.

Cohen, Tian, and Vetro [8, 9] extended some of the concepts used to code arbitrarily shaped regions in images and video and the well-known shape adaptive DCT (SA-DCT) [29] to voxelized 3D point clouds. 3D point clouds were down-sampled to a uniform grid, which in turn was partitioned into 3D blocks so that a transform could be directly applied.

A 3D lossy compression system is proposed in [18] to reduce the amount of data of a 3D point set but preserve as much information as possible. It based on plane extraction which represent the points of each scene plane as a Delaunay triangulation and a set of points/area information. It also supports a color segmentation step to preserve original scene color information and provides a realistic scene reconstruc-

tion.
The PRED (Research and Development) project [13], realized by Jamet and Baijal, has lead to a method showing improvements using TSLVQ with two quantization methods (2x2x2 or 3x3x3 Tree decomposition). The 3D point cloud is first assigned to the root object, then divided into leaves using tree structure. The quantization process is preceded by a computation of new leaves lambda criteron depending on the distortion and rate values. Then for each new leaf, all the two quantization methods will be applied. The quantization with the best lambda result will be chosen. The quantization step is continued until one criteria is full filled (number of steps, rate/distortion value reached). The final compressed point cloud are the centroids of each leaf (centroid of all points inside the leaf). The color assigned to a centroid, is the average color of all points inside the leaf.

2.2.2 Graph-based Encoding

Human visual system cannot perceive many high frequency details of a signal. This fact is used to design many transform coding schemes, i.e. point cloud can be efficiently compressed by discarding high frequency information. The main idea is to organize all points in a graph structure using K-nearest neighbor, etc., to assume that point coordinates in 3D space are signals on this graph, to obtain a frequency domain presentation of the signal and to apply the promising graph transform.
A large number of graph algorithms exploiting graph sampling techniques have been implemented for the purpose of 3-D data compression. In [22] Nguyen et al. have shown that a particularly effective way to compress moving human body sequences is to model the dynamic 3D objects using a dynamic colored mesh based on a quad subdivision and splitting the vertices into two independent sets [20], and then to couple this by applying recently developed Graph Wavelet Filter Banks GWFBs by Narang and Ortega in [19], to both time-varying geometry and color signals living on a mesh representation of the human body.

A more efficient solution can be found by introducing a hierarchical transform which permits specifying different quality levels together with providing a higher compression gain. So far, transform-based compressor [33, 26] have been targeting color information since traditional transforms work well on geometric data.
In [33], authors proposed a method to compress attributes (colors, etc.) employing graph transform (GT) to decorrelate data. Its main idea was to construct graphs on small neighborhoods of nearby occupied voxels within a 3D block of the point cloud, which was firstly organized into an octree structure. A graph Laplacian matrix was computed based on the occupancy of the voxels, whose eigen vector matrix was further applied as the transform matrix to the attributes considered as signal residing on the graph's nodes in order to compress them efficiently.
Queiroz and Chou in their paper [26] proposed a real time Region-Adaptive Hierarchical Transform (RAHT), which is a hierarchical sub-band transform ressembling a Haar Wavelet. This hierarchical transform permits the representation and the encoding of color component signals on 3D point clouds in real time. Geometry compression was discussed for completeness and authors used the well established octree method in [33, 23].

2.2.3 Visual Rendering

The rendering engine has a tremendous impact on which essential information to convey for a rich and immersive user experience. In reality, many different post-processing steps occur behind the scenes: color calibration and gamma correction processing, hole filling using interpolation techniques, etc. allow and produce different color rendering and emotional experiences to the user rendering technologies for reaching a best quality viewing experience.

Many researchers have employed specific techniques to deal with perception quality improvement problem. Coeurjolly, Gueth, and Lachaud proposed a new variational approach in [7] to regularize a digital surface while constructing a piecewise smooth quadrangulated surface from a digital surface. More formally a variational formulation which efficiently regularizes digital surface vertices is introduced, while complying with a prescribed input normal vector field estimated on the digital structure. The regularized surface is consistent with respect to an input normal vector field, and has a smooth embedding. If the input normal vector field is piecewise smooth, the regularization preserves these features. Hence, it should comply to a good approximation of the normal vector field.

2.2.4 Subjective and Objective Quality Evaluation

The increasing availability and use of point cloud data in recent years is demanding high performance compression solutions. Naturally, methods to perform the quality assessment of compressed point clouds are therefore very much needed, namely metrics to measure the geometry distortion of point clouds when positioning errors are present. This is a rather challenging problem since this 3D representation format is unstructured. In this context, few researchers are involved to perform objective and subjective quality assessment of point clouds degraded by compression artifacts. An overview of existing quality evaluation metrics is provided in [14],[31] to evaluate afterward the correlation of the most popular objective quality metrics (Point-to-Point and Point-to-Plane metrics) with human perception (subjective assessment based on subjects' scores to make judgments of quality).

2.3 Point Cloud: Data Sets

Based on their characteristics, datasets can be organized in two categories: static objects and dynamic objects.

The first category consists of point clouds representing only one object with a relative large number of points (from a few hundreds of thousands to tens of millions), colours

per point. Most of the objects from this category were obtained by reconstructing the point clouds from images of real monuments, statues or buildings obtained within two research projects [Culture 3D Clouds: French National research project (Culture 3D Clouds: French National research project (www.c3dc.fr))][BRIDging the Gap for Enhanced broadcasT (Bridget) European research project (www.ict-bridget.eu)] but also by scanning an entire campus [Stanford Large-Scale 3D Indoor Spaces Dataset]. This category of content is important for supporting storage and distribution of captured 3D content and other 3D assets.
The second category consists of sequences of point clouds that were obtained by scanning real moving people [A Voxelized Point Cloud Dataset E. d'Eon, B. Harrison, T. Myers, P. A. Chou]. This category also contains temporal component as all the sequences are represented with a point cloud for each frame, with a few hundred thousands of points for each frame and with colour information for each point. The data in this category also includes color attributes for each point.

For example, the well-known Oakland 3-D Point Cloud data set (ref: Contextual Classification with Functional Max-Margin Markov Networks) contains less than 2 million labelled 3-D points cloud laser data collected from a moving platform in a urban environment. The data was collected using Navlab11 equiped with side looking SICK LMS laser scanners and used in push-broom. The data was collected around CMU campus in Oakland, Pittsburgh, PA. Another popular data set, the NYU benchmark, provides only indoor scenes. Finally, both Sydney Urban Objects data set and the IQmulus and TerraMobilita Contest use a 3D Velodyne LIDAR mounted on a car which provides much lower point density than a static scanner. The same counts for the Vaihingen3D airborne benchmark.
A large-scale point cloud benchmark (A new large-scale point cloud classification benchmark) closes the gap and provides a large labelled 3D point cloud data set of natural scenes with over 4 billion points in total. It also covers a range of diverse urban scenes: churches, streets, railroad tracks, squares, villages, soccer fields, cas-

tles to name just a few.
L-CAS 3D Point Cloud People Dataset -Collected by the Velodyne VLP-16 3D LiDAR- contains 28,002 Velodyne scan frames acquired in one of the main buildings (Minerva Building) of the University of Lincoln, UK. Total length of the recorded data is about 49 minutes. Data were grouped into two classes according to whether the robot was stationary or moving. All data are provided for research purposes.

2.4 Rate-Distortion

and

3 Discussion and Conclusion

With almost a quarter century of practical feasibility, technologies for acquisition, compression and rendering of 3D point clouds have been advanced rapidly in these years, and the number of applications using 3D point clouds has been up and well, gaining popularity and progressing into the markets.

The 3D point cloud processing is the main object of attention of this thesis. Our overriding research strategy outlines how we would analyze 3D data represented by Point Cloud, while point clouds providing 3D geometry information of the objects within the scene, and defining objects by point clouds permits to use raw data and simplify handling and processing of data and guarantees the easy conversion from points to another representation models.

This thesis focuses on the exploitation of representation models for compression of the large size 3D data. First a short introduction was presented. This was followed by a detailed overview on 3D data representation models used in 3D data representation and encoding fields as well as different related work existing in the literature. Two encoding scheme could be identified, which are: Octree-based encoding and Graph-based encoding. Some proposed methods compress the geometry of the 3D point cloud while others are based on attribute (color) compression. Beyond these, there are also some approaches that utilized both geometry and attributes for compression.

Key requirements for our future methods during the coming year are:

1. Lossy compression with Distortion/bit-rate control

2. Low complexity encoding
3. Geometry/ attributes coding
4. High perceptual quality of point cloud rendering

Our main goals for near future work are:

- Improvement of the Rate/Distortion computation by taking into consideration the surface normals.
- The computation of the representant of a leaf proposed in [13] could be further improved by considering the leaves around (context algorithm).
- The 3D point clouds will be represented relying on tree structures such as the octree and K-dtree, etc. for recursively splitting and subdividing the point cloud which should be grown to the largest extent possible.
- The tree structure based reduction: Pruning techniques, firstly proposed by Breiman [6, Chapter 3], will be employed on the obtaining tree in order to obtain right sized trees while this can be used to improve quality control, especially when using points clouds with constraints (hole, high resolution region needed).

During the rest of this year and the incoming year, we aim to participate in:

- Journées Informatique et Géométrie / GT GDMM, 20, 21, 22 juin 2018.
- National/ International Conference before first year-end.
- Next edition of 2019 IEEE SPS Summer School on Signal Processing.

The professional training that I validated during the first 5 months are:

- Comprendre les enjeux d'un état de l'art et de la propriété intellectuelle et industrielle (15 hours).
- Anglais pour la recherche (20 hours).

- La motivation : rendre actif les étudiants 4A (4 hours).
- Café form@doc (sciences) (3 hours).

Bibliography

[1] Edward H. Adelson, Edward H. Adelson, and James R.” Bergen. “The Plenoptic Function and the Elements of Early Vision”. In: *COMPUTATIONAL MODELS OF VISUAL PROCESSING* (1991), pp. 3–20. DOI: 10.1.1.2.9848.

[2] Enam Maglo Adrien. *Progressive and Random Accessible Mesh Compression.* Tech. rep. 2013. URL: http://tel.archives-ouvertes.fr/tel-00966180/.

[3] John G Apostolopoulos et al. “The road to immersive communication”. In: *Proceedings of the IEEE* 100.4 (2012), pp. 974–990. DOI: 10.1109/JPROC.2011.2182069.

[4] Norman I Badler and Andrew S Glassner. “3D object modeling”. In: *SIGGRAPH 97 Introduction to Computer Graphics Course Notes* 5 (1997).

[5] Béla Bollobás. *Modern graph theory.* Vol. 184. Springer Science & Business Media, 2013. DOI: 10.1007/978-1-4612-0619-4.

[6] Leo Breiman. *Classification and regression trees.* Wadsworth International Group, 1984. URL: http://doi.org/10.1002/cyto.990080516.

[7] David Coeurjolly, Pierre Gueth, and Jacques-Olivier Lachaud. “Digital surface regularization by normal vector field alignment”. In: *International Conference on Discrete Geometry for Computer Imagery.* Springer. 2017, pp. 197–209. DOI: 10.1007/978-3-319-66272-5_17.

[8] Robert A Cohen, Dong Tian, and Anthony Vetro. “Attribute compression for sparse point clouds using graph transforms”. In: *Image Processing (ICIP), 2016 IEEE International Conference on.* IEEE. 2016, pp. 1374–1378. DOI: 10.1109/ICIP.2016.7532583.

[9] Robert A Cohen, Dong Tian, and Anthony Vetro. "Point cloud attribute compression using 3-D intra prediction and shape-adaptive transforms". In: *Data Compression Conference (DCC), 2016*. IEEE. 2016, pp. 141–150. DOI: 10.1109/DCC.2016.67.

[10] Digi-Capital. "Augmented/Virtual Reality revenue forecast revised to hit $120 billion by 2020". In: (). URL: http://www.digi-capital.com/news/2016/01/augmentedvirtual-reality-revenue-forecast-revised-to-hit-120-billion-by-2020/.

[11] Nykamp DQ. *Small undirected network with labeled nodes and edges*. URL: http://mathinsight.org/image/small_undirected_network_labeled.

[12] Andrew Glassner. "Building vertex normals from an unstructured polygon list". In: *Graphics Gems*. Elsevier, 1994, pp. 60–73.

[13] Ronan Jamet and Apoorv Baijal. "Immersive Data Representation based on Vector Quantization of Points Cloud". In: *PRED Project, Computer science department, Polytech' Nantes* (2018), pp. 1–54.

[14] Alireza Javaheri et al. "Subjective and objective quality evaluation of compressed point clouds". In: *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)* (Oct. 2017). DOI: 10.1109/mmsp.2017.8122239. URL: http://dx.doi.org/10.1109/MMSP.2017.8122239.

[15] Julius Kammerl et al. "Real-time compression of point cloud streams". In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 778–785. DOI: 10.1109/ICRA.2012.6224647.

[16] Ahmet Kondoz and Tasos Dagiuklas. *Novel 3D Media Technologies*. Springer, 2015. DOI: 10.1007/978-1-4939-2026-6.

[17] Gauthier Lafruit et al. "Technical report of the joint ad hoc group for digital representations of light/sound fields for immersive media applications". In: *Doc. MPEG N16352, Geneva, Switzerland* (2016). DOI: 10.13140/RG.2.1.4739.3524.

[18] Vicente Morell et al. "Geometric 3D point cloud compression". In: *Pattern Recognition Letters* 50 (2014), pp. 55–62. DOI: 10.1109/APSIPA.2013.6694334.

[19] Sunil K Narang and Antonio Ortega. "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs". In: *IEEE transactions on signal processing* 61.19 (2013), pp. 4673–4685. DOI: 10.1109/TSP.2013.2273197.

[20] Sunil K Narang and Antonio Ortega. "Perfect reconstruction two-channel wavelet filter banks for graph structured data". In: *IEEE Transactions on Signal Processing* 60.6 (2012), pp. 2786–2799. DOI: 10.1109/TSP.2012.2188718.

[21] Ren Ng. *Digital light field photography.* stanford university California, 2006.

[22] Ha Q Nguyen, Philip A Chou, and Yinpeng Chen. "Compression of human body sequences using graph wavelet filter banks". In: *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE. 2014, pp. 6152–6156. DOI: 10.1109/ICASSP.2014.6854786.

[23] Tilo Ochotta and Dietmar Saupe. *Compression of point-based 3D models by shape-adaptive wavelet coding of multi-height fields.* SPBG'04 Symposium on Point - Based Graphics, 2004. DOI: 10.2312/SPBG/SPBG04/103-112.

[24] Qi Pan, Gerhard Reitmayr, and Tom Drummond. "ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition". In: *Procedings of the British Machine Vision Conference 2009* (2009). DOI: 10.5244/c.23.112.

[25] Jingliang Peng, Chang-Su Kim, and C-C Jay Kuo. "Technologies for 3D mesh compression: A survey". In: *Journal of Visual Communication and Image Representation* 16.6 (2005), pp. 688–733. DOI: 10.1016/j.jvcir.2005.03.001.

[26] Ricardo L de Queiroz and Philip A Chou. "Compression of 3D point clouds using a region-adaptive hierarchical transform". In: 25.8 (2016), pp. 3947–3956. DOI: 10.1109/TIP.2016.2575005.

[27] Radu Bogdan Rusu and Steve Cousins. "3d is here: Point cloud library (pcl)". In: *Robotics and automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1–4. DOI: 10.1109/ICRA.2011.5980567.

[28] Ruwen Schnabel and Reinhard Klein. "Octree-based Point-Cloud Compression." In: *Spbg* 6 (2006), pp. 111–120. DOI: `10.2312/SPBG/SPBG06/111-120`.

[29] Thomas Sikora and Bela Makai. "Shape-adaptive DCT for generic coding of video". In: *IEEE Transactions on Circuits and Systems for Video Technology* 5.1 (1995), pp. 59–62. DOI: `10.1109/76.350781`.

[30] Amir Arsalan Soltani et al. "Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1511–1519. DOI: `10.1109/CVPR.2017.269`.

[31] D Tian et al. "Geometric Distortion Metrics for Point Cloud Compression". In: *IEEE International Conference on Image Processing (ICIP)* (2017). DOI: `10.1109/ICIP.2017.8296925`.

[32] Faxin Yu et al. *Three-dimensional model analysis and processing*. Springer Science & Business Media, 2011. DOI: `10.1007/978-3-642-12651-2`.

[33] Cha Zhang, Dinei Florêncio, and Charles Loop. "Point cloud attribute compression with graph transform". In: *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE. 2014, pp. 2066–2070. DOI: `10.1109/ICIP.2014.7025414`.