1 2 9 0

UNIVERSIDADE D
COIMBRA

Mariana Assunção Roque

Um estudo empírico do efeito da compressão no desempenho de algoritmos de segmentação de nuvens de pontos

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores no ramo de Computadores orientada pelo Professor Doutor Luís Alberto da Silva Cruz e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro de 2019

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Um estudo empírico do efeito da compressão no desempenho de algoritmos de segmentação de nuvens de pontos

Mariana Assunção Roque

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Orientador: Professor Doutor Luís Alberto da Silva Cruz

Júri

Presidente:	Professora Doutora Maria do Carmo Raposo de Medeiros
Orientador:	Professor Doutor Luís Alberto da Silva Cruz
Vogal:	Professor Doutor Marco Alexandre Cravo Gomes

Setembro de 2019

You must do the things you think you cannot do.

- Eleanor Roosevelt

Agradecimentos

Esta dissertação simboliza a conclusão da última etapa do meu percurso académico na Universidade de Coimbra e, como tal, gostaria de prestar os meus agradecimentos.

Primeiro, gostaria de agradecer ao meu orientador Professor Doutor Luís Cruz por me ter dado a oportunidade de desenvolver esta dissertação bem como todas as orientações.

Agradeço também o apoio do projecto PTDC/EEI-COM/31527/2017 e do Instituto de Telecomunicações, pólo de Coimbra, bem como a colaboração do Laboratório HCMR do Instituto de Sistemas e Robótica, nas pessoas do Professor Doutor Urbano Nunes e dos seus colaboradores Luís Garrote e Tiago Barros. Desde já, um muito obrigada por todo o apoio prestado e pela ajuda disponibilizada.

A todos os colegas e amigos que me ajudaram a ultrapassar todas as barreiras e dificuldades que foram surgindo no decorrer destes anos, um muito obrigada. Guardarei sempre a vossa amizade e carinho para comigo, bem como todos os momentos que passámos juntos. Quero ainda agradecer especialmente aos que me aturaram ao longo deste percurso difícil e que sempre me motivaram e aconselharam. Sem vocês, não seria a mesma coisa.

Quero ainda agradecer ao Ricardo pelo seu apoio, motivação e por todos os momentos partilhados que contribuíram para a conclusão deste percurso.

Por último, agradeço à minha família por todo o apoio e motivação prestada ao longo deste percurso académico, em especial aos meus pais que me proporcionaram todas as condições para a realização deste percurso e nunca me deixaram desistir.

A todos,

Muito obrigada.

Contents

1 Introduction **1**

1.1 Contextualization 1

1.2 Objectives 3

1.3 Contributions 3

1.4 Acknowledgments 3

1.5 Document Structure 3

2 Background information **5**

2.1 Point Cloud Compression Algorithms 5

2.1.1 MPEG Geometry-based Point Cloud Compression Encoder 6

2.1.2 MPEG Video-based Point Cloud Compression Encoder 7

2.1.3 Fast Resampling on Point Clouds via Graphs 8

2.1.4 LASzip 9

2.2 Point Cloud Segmentation Algorithms 10

2.2.1 K-Means 10

2.2.2 Fuzzy C-Means 11

2.2.3 K-Medoids 11

2.2.4 Mean Shift 11

2.2.5 Clustering based on Euclidean Distance 12

2.2.6 Clustering based on subtractive clustering 12

2.3 Point Cloud Compression Quality Assessment 13

3 Methodology **15**

3.1 General use PC dataset 15
3.1.1 Point Cloud Test Data 15
3.1.2 Data Preparation 16
3.1.3 Segmentation Tests 17
3.1.4 Performance Measures 18
3.2 LiDAR Dataset 19
3.2.1 Point Cloud Test Data 19
3.2.2 Data Preparation 20
3.2.3 Segmentation and Classification Tests 21
3.2.4 Performance Measures 21

4 Results and Analysis **23**
4.1 Compression Results 23
4.1.1 General use PC dataset 23
4.1.2 LiDAR dataset 33
4.2 Segmentation Results 36
4.2.1 General use PC dataset 37
4.2.2 LiDAR dataset 47

5 Conclusion and Future Work **57**
5.1 Conclusion 57
5.2 Future work 58

A Compression results **62**

B Segmentation results **66**

List of Figures

2.1 Representation of an octree decomposition and partitioning byte codes (1 means sub-block will be subpartitioned, 0 otherwise). 5

2.2 Overview of the MPEG G-PCC encoder (left) and decoder (right) from [1]. 6

2.3 Overview of the MPEG V-PCC encoder and decoder from [2]. 8

2.4 Comparison between uniform and contour-enhanced resample from [3]. 9

2.5 Difference between P2Point and P2Plane metrics from [4]. 14

3.1 Representation of two composite PCs. 16

3.2 Projection of a frame from the KITTI dataset. 20

3.3 Flowchart of the approach followed. 21

4.1 Representation of the bitrate results. 24

4.2 Representation of the scaling ratio results. 25

4.3 Representation of the P2Point metric results using MPEG V-PCC. 27

4.4 Representation of the P2Point metric results using MPEG G-PCC. 28

4.5 Representation of the P2Point metric results using FRPC. 29

4.6 Representation of the P2Plane metric results using MPEG V-PCC. 30

4.7 Representation of the P2Plane metric results using MPEG G-PCC. 31

4.8 Representation of the P2Plane metric results using FRPC. 32

4.9 Representation of the overlap between the original and decompressed PC using a 10 bit depth. 34

4.10 Representation of the overlap between the original and decompressed PC using a 15 bit depth. 35

4.11 Representation of the overlap between the original and decompressed PC using a 20 bit depth. 36

4.12 PAVD results. 42

4.13 DCH result for experiment 1 using Fuzzy C-Means 43

4.14 DC results for K-Means algorithm. 44

4.15 PNP results for Pcsegdist algorithm. 45

4.16 Representation of the running time according the quality level for K-Medoids algorithm 47

4.17 Representation of each confusion matrix parameters. 55

B.1 DCH results. 70

B.2 DC results. 75

B.3 PNP results. 79

B.4 Representation of the running time of each function according the quality level 84

List of Tables

3.1 Configurations used in MPEG V-PCC, MPEG G-PCC, and FRPC 17

3.2 Configuration parameters used in segmentation tests. 18

3.3 Clustering performance measures. 18

3.4 Performance measures used in the segmentation algorithms. 19

3.5 Configurations used with MPEG G-PCC and FRPC for purposes of segmentation and classification 20

4.1 Overlap results using a 10 bit depth PC. 33

4.2 Overlap results using a 15 bit depth PC. 35

4.3 Overlap results using a 20 bit depth PC. 36

4.4 Number of PC with visual differences to the original one 37

4.5 Confusion matrix obtained after the classification performed on frame 000000 48

4.6 Confusion matrix obtained after the classification performed on frame 000001 49

4.7 Confusion matrix obtained after the classification performed on frame 000002 50

4.8 Confusion matrix obtained after the classification performed on frame 000003 51

4.9 Confusion matrix obtained after the classification performed on frame 000004 52

4.10 Parameters from each confusion matrix 53

A.1 Bitrate and scaling ratio results of experiment 1. 62

A.2 Bitrate and scaling ratio results of experiment 2. 63

A.3 Bitrate and scaling ratio results of experiment 3. 63

A.4 Results of MPEG V-PCC of experiment 1 63

A.5 Results of MPEG G-PCC of experiment 1 64

A.6 Results of FRPC of experiment 1 64
A.7 Results of MPEG V-PCC of experiment 2 64
A.8 Results of MPEG G-PCC of experiment 2 64
A.9 Results of FRPC of experiment 2 64
A.10 Results of MPEG V-PCC of experiment 3 65
A.11 Results of MPEG G-PCC of experiment 3 65
A.12 Results of FRPC of experiment 3 65

B.1 Results of K-Means algorithm. 85
B.2 Results of the K-Medoids algorithm 86
B.3 Results of Fuzzy C-Means algorithm 87
B.4 Results of the Mean Shift algorithm 88
B.5 Results of the Pcsegdist 89

Abstract

Point clouds are sets of points which represent a 3D object/scene by their coordinates and optional attributes such as color, reflectance or other. Point clouds are being used in several application areas such as entertainment, terrain representation, medical imaging and, more recently, autonomous vehicle guidance systems. Due to the large size of point clouds, these applications would require a huge power processing and, in some cases, tasks may not be able to be performed in real time. Thus, compression is used to tackle the challenges of storage and real-time transmission.

It is known that lossy compression introduces geometric distortions to the point clouds which are usually dependent on the compression rate. Due to being necessary to segment the component objects of the reconstructed/decompressed point cloud, it is important to understand and characterize the effect of the type and degree of compression on the performance of the segmentation and classification tasks.

In this dissertation, two sets of experiments are described: one with general use point clouds and the other using a particular type of point clouds, more precisely LiDAR. This division was made because it is likely that the results are different for these two types due to the amount of precision and uses of each point cloud type. These experiments are designed to empirically evaluate the effect of different point cloud compression methods, employed at different compression rates, on the performance of several point cloud segmentation and classification algorithms. For that, several performance measures are used to evaluate the behavior of each case.

Keywords

Point cloud, Compression, Segmentation, Classification

Resumo

Nuvens de pontos são conjuntos de pontos que representam um objeto ou cena 3D, em que os pontos são representados pelas respetivas coordenadas 3D e atributos opcionais tais como cor, reflectância, entre outros. Estas são usadas em várias áreas de aplicação como, por exemplo, entretenimento, representação de terrenos, imagens médicas e, mais recentemente, sistemas de condução autónoma de veículos. No entanto, devido ao grande volume de dados necessários para representar as nuvens de pontos, essas aplicações iriam precisar de um grande poder de processamento e, em alguns casos, poderia não ser possível realizar as tarefas em tempo real. Portanto, a compressão é usada para combater o problema de armazenamento e transmissão em tempo real.

Sabe-se que a compressão com perdas introduz distorções geométricas que, geralmente, dependem do grau de compressão. Dado que em algumas aplicações é necessário segmentar os objectos que compõem a nuvem de pontos reconstruída/descomprimida, é importante perceber e caracterizar o efeito do tipo e grau de compressão na performance das tarefas de segmentação e classificação.

Nesta dissertação, são descritos dois tipos de experiências: uma com nuvens de pontos de uso geral e a outra usando um caso particular das nuvens de pontos, mais precisamente, o LiDAR. Esta divisão foi feita, pois é provável que os resultados destas duas experiências sejam diferentes devido às aplicações distintas destas classes de nuvens de pontos, assim como respetivos requisitos de precisão. Estas experiências foram criadas para avaliar empiricamente o efeito de diferentes métodos de compressão de nuvens de pontos usando diferentes graus de compressão na performance de vários algoritmos de segmentação e classificação. Para isso, várias medidas de performance são usadas para avaliar o comportamento de cada caso.

Palavras-Chave

Nuvem de pontos, Compressão, Segmentação, Classificação

Glossary

8iVFB 8i Voxelized Full Bodies.

AVD absolute difference of volumes.

CLARA Clustering LARge Applications.

DC distance between the centers.

DCH distance between the convex hulls.

EUVIP European Workshop on Visual Information Processing.

FRPC Fast Resampling on Point clouds via Graphs.

G-PCC Geometry-based Point Cloud Compression.

HCMR Human Centered Mobile Robotics.

HEVC High Efficiency Video Codingx.

ISR Institute of Systems and Robotics.

LiDAR Light detection and ranging.

MPEG Moving Picture Experts Group.

MPEG-4 AVC MPEG-4 Advanced Video Coding.

MSE Mean Squared Error.

P2Plane Point-to-plane.

P2Point Point-to-point.

PAM Partitioning Around Medoids.

PAVD percentage of absolute difference of volumes.

PC Point Clouds.

PCC Point Cloud Compression.

PNP percentage of absolute difference of number of points.

PSNR Peak Signal-to-Noise Ratio.

RAHT Region-adaptive Hierarchical Transform.

V-PCC Video-based Point Cloud Compression.

Chapter 1

Introduction

1.1 Contextualization

The need of more realistic models and 3D representations has become a concern over the last years, so several approaches emerged. Formerly, meshes [5] were widely used to represent 3D objects/scenes. In fact, points were converted into polygonal meshes which required large memory and had a high computational cost. Most of the time, meshes could not be processed due to their huge size and without a costly simplification. Point Clouds (PC) emerged as an alternative to meshes since they simplify the representation and allow a faster reconstitution of the surface than meshes without having to worry about simplification. This process requires less memory since data structures are not necessary and it tends to be more efficient computationally which saves computation time.

PC are sets of values that contain information about the location (represented by its Cartesian coordinates) and optionally attributes, such as color, reflectance, temperature of points that represent samples of the surfaces of simple objects or complex scenes. According to Moving Picture Experts Group (MPEG), they can be classified as [6]:

- Category 1: high-detail static PC objects such as buildings;
- Category 2: time-varying PC objects such as people performing tasks;
- Category 3: dynamically acquired PC such as the ones captured from Light detection and ranging (LiDAR) devices on vehicles/drones.

PC can be obtained [7]:

- directly: imaging systems that are projected to collect this kind of information which can consist in sparse points or dense clouds on a surface. These points are directly detected and determined with certain types of hardware. Some examples are LiDAR, time-of-flight cameras and contact technologies;

- indirectly: algorithms that can extract this kind of information from imaging systems that are not designed to deal with 3D data. Some examples are photogrammetry, depth maps and light fields.

A LiDAR sensor [7–10] is an optical remote sensor which produces a PC representation of the surrounding environment that is utilized to detect and classify objects. This sensor spins around its axis while collecting data using laser beams. The reflection of each one is used to measure the light intensity and the distance to each reflection origin. In the end, a PC image is obtained for each rotation of the sensor, having the location and shape of 3D objects. This technology can be terrestrial, i.e, sitting on the ground, or aerial, i.e., put a laser scanning sensor in the air. Terrestrial LiDAR is more accurate than the aerial one and that can be justified by the fact that in the air it is not possible to know precisely where the laser is pointing and since it can move, there can be discrepancies while scanning causing results' distortion. However, with airborne LiDAR, it is easier to collect data related to the terrain elevation. One of LiDAR disadvantages is the fact that it does not collect color or texture information. One way to avoid that problem is overlaying the information from LiDAR onto the color information captured using photographic techniques. However, this method can be imprecise. These sensors are usually used in active safety systems in order to reduce the number of accidents by implementing collision avoidance systems which are independent of the driver's control.

That said, the usage of PC has increased along the years so there is a need to research them. Since PC contain data that can occupy several gigabytes, there is a need to compress them efficiently in order to have accurate information occupying as little space as possible [4, 6]. However, the number of points can have an impact on how realistic a rendered scene/object is [6]. When talking about compressing images, the main goal is to reduce its size without compromising its quality which is why it is also important to calculate errors between PC in order to quantify any possible quality degradation that may occur [4, 11]. Compression can be classified as [11]:

- Lossy: when it is possible to reconstruct data from the original image but with some slight differences and can even introduce artifacts due to its low bitrates. It is usually used in situations that allow a minor loss of fidelity.

- Lossless: when is possible to reconstruct the original image exactly. It is used in situations that do not allow a loss of fidelity such as medical applications of image.

PCs can be used in virtual-, augmented-, and mixed-reality applications, in 3D content creation, in 3D printing, in medical applications such as imaging and body parts manufacturing, in autonomous vehicles, and drone guidance [12]. As expected, the requirements of PC representation formats and compression methods will vary according to the use case. For example, medical applications will need a more accurate reconstruction of a compressed PC than 3D printing for rapid prototyping. Therefore, there is a need for efficient compression methods for PCs enabling accurate PC representation with few bits per point. In some applications, PCs are processed to identify constituent objects, for example pedestrians and cars in automotive applications, through the use of PC segmentation algorithms. There is a significant difference between the types of data and the levels of compression required which is why there are two sets of experiments in this dissertation: one considering general use data and the other considering a particular case, LiDAR data. This last part was a collaboration with the laboratory Human Centered Mobile Robotics (HCMR) from Institute of Systems and Robotics (ISR).

1.2 Objectives

The main goal of this work is to study how compression affects the performance of processing operations such as segmentation and classification. It is important to know how compressed can the data be while still achieving good results when those processing operations are performed on the compressed data. In order to do that, several datasets are going to be compressed using different methods and parameters and, later, tested using different segmentation/classification methods.

1.3 Contributions

One contribution of this work is the paper "An Empirical Study of the Effect of Point Cloud Compression on the Performance of Segmentation" for the 8th European Workshop on Visual Information Processing (EUVIP). However, the main contribution is the performed study and quantitive analysis of the effect of PC (generic and LiDAR) compression on the performance of some segmentation algorithms.

1.4 Acknowledgments

The work developed in this dissertation was partially carried out within the framework of the project PTDC/EEI-COM/31527/2017 and supported by the IT infrastructure facilities.

1.5 Document Structure

The remainder of this dissertation is organized in the following structure: in chapter 2 the theoretical foundations of this work are presented, in chapter 3 is presented the work done in the context of this dissertation, in chapter 4 the results of this work and their analysis are presented, and, finally, in chapter 5 the final conclusions and proposals for future works are presented.

Chapter 2

Background information

The following two sections provide a brief review of the Point Cloud Compression (PCC) methods used in this work as well as the PC segmentation algorithms analyzed. The third section of this chapter describes some metrics used in PCC quality assessment.

2.1 Point Cloud Compression Algorithms

PCC methods can be loosely classified into the following three classes [7]:

- Based on 3D encoding: point information encoding exploits spatial relationships between points. Octree-based methods, mesh encoding, graph-based and voxel-based representations are some examples that fit into this category. In octree-based methods, the bounding box of the PC is recursively subdivided into eight sub-units of cubic shape. Only the sub-cubes that are flagged as occupied (contain at least one point) are again subdivided as shown in Figure 2.1. This process stops when there are no more points left or until a certain level of precision is attained. MPEG recently proposed a method based on these principles, the MPEG Geometry-based Point Cloud Compression (G-PCC) encoder [1] which is described in 2.1.1.

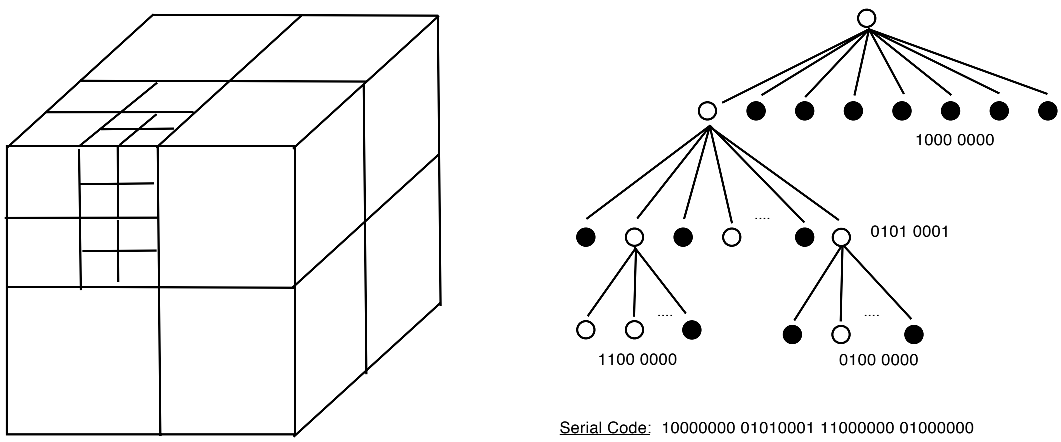


Figure 2.1: Representation of an octree decomposition and partitioning byte codes (1 means sub-block will be subpartitioned, 0 otherwise).

- Based on 2D Projections: the points in the PC are projected onto several planes and then encoded using existing 2D image/video compression methods. Methods of this type can be used to encode geometry and attribute information. MPEG Video-based Point Cloud Compression (V-PCC) [2] which is described in 2.1.2 was designed for encoding dynamic PCs based on projections.

- Hybrid and Other: methods that use combinations of the previous approaches and methods that do not fit into any other category. Two methods that belong to this category are Fast Resampling on Point clouds via Graphs (FRPC) [3] which is described in section 2.1.3 and LASzip [9] described in section 2.1.4.

2.1.1 MPEG Geometry-based Point Cloud Compression Encoder

MPEG G-PCC [1] is an algorithm that was designed to compress PCs of Category 1 and 3. Figure 2.2 presents an overview of the encoder and decoder. The modules presented as green are usually used with Category 1 PCs and orange modules with Category 3 PCs. It also supports color and/or reflectance attributes or no attributes at all.

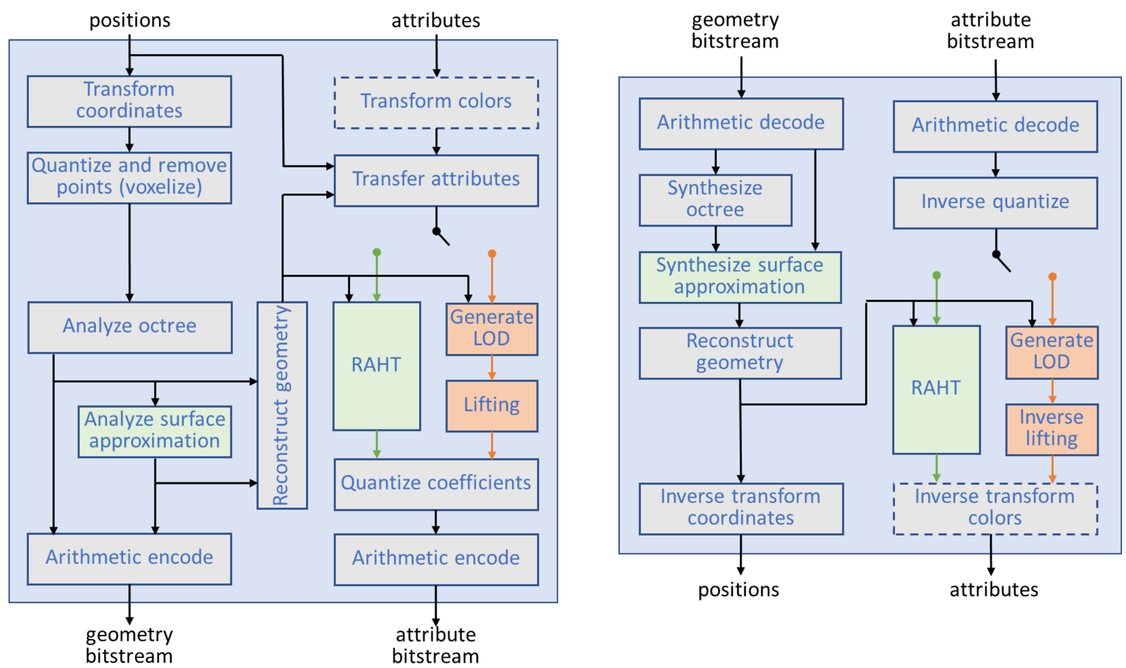


Figure 2.2: Overview of the MPEG G-PCC encoder (left) and decoder (right) from [1].

Before compression, point positions are represented by non-negative integers which where obtained by rounding the floating point positions in the internal coordinate system. It is possible that some points may have the same position (duplicated points) but they can be removed. Points having the same position but different attributes are merged into one and average attributes are computed. This whole process is called voxelization, i.e., grouping several points into one voxel. Within the voxel, points location are quantized to the voxel center and the attributes are averaged.

Then the PC geometry data is encoded using an octree-based codec, following a procedure analog to that described in section 2.1. The color attributes can be encoded using the Region-adaptive Hierarchical Transform (RAHT) method [13] based on a hierarchical sub-band transform and an arithmetic encoder. The main idea of RAHT is to group colors in a lower level to predict colors in the next level until reaching the root leaving unoccupied nodes out.

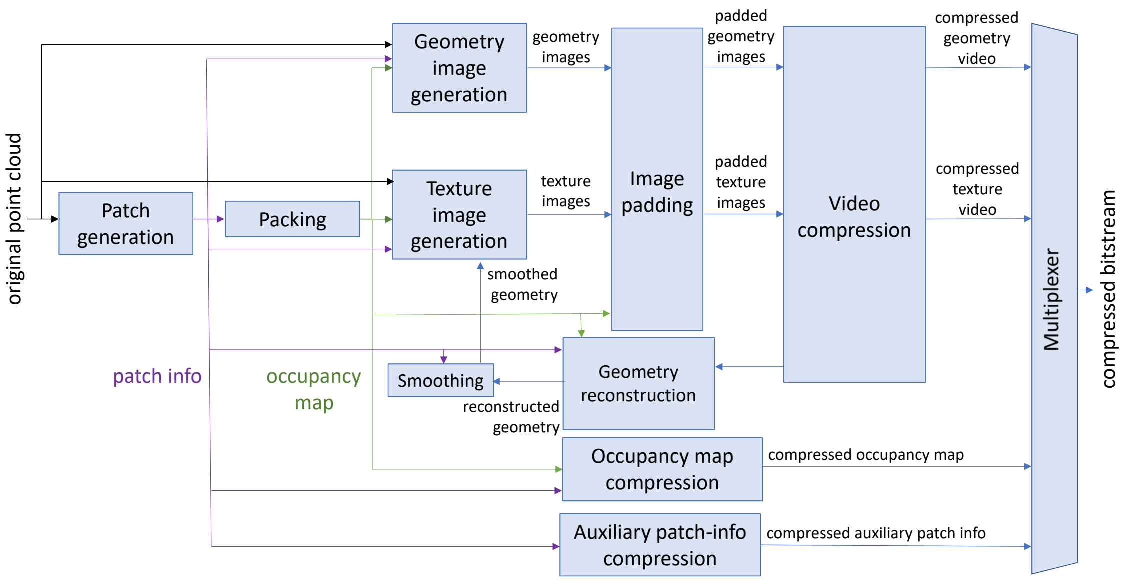
2.1.2 MPEG Video-based Point Cloud Compression Encoder

MPEG V-PCC [2, 6] is an algorithm based on 2D projections that was designed to compress Category 2 PCs. It leverages existing video codecs, such as MPEG-4 Advanced Video Coding (MPEG-4 AVC), High Efficiency Video Codingx (HEVC), to compress the projected geometry and texture. The individual PCs of the dynamic PC are projected generating two video sequences, one with geometry information and the other with attribute information. The patches generated by the projection are packed into a 2D pixel array forming the 2D (pseudo) video frames. Additional data such as occupancy maps and auxiliary patch information are also encoded and added to the bitstream to be used in the PC reconstruction. Figure 2.3 presents a diagram showing the major signal processing operations involved and bellow it is presented a short explanation of how this method works.

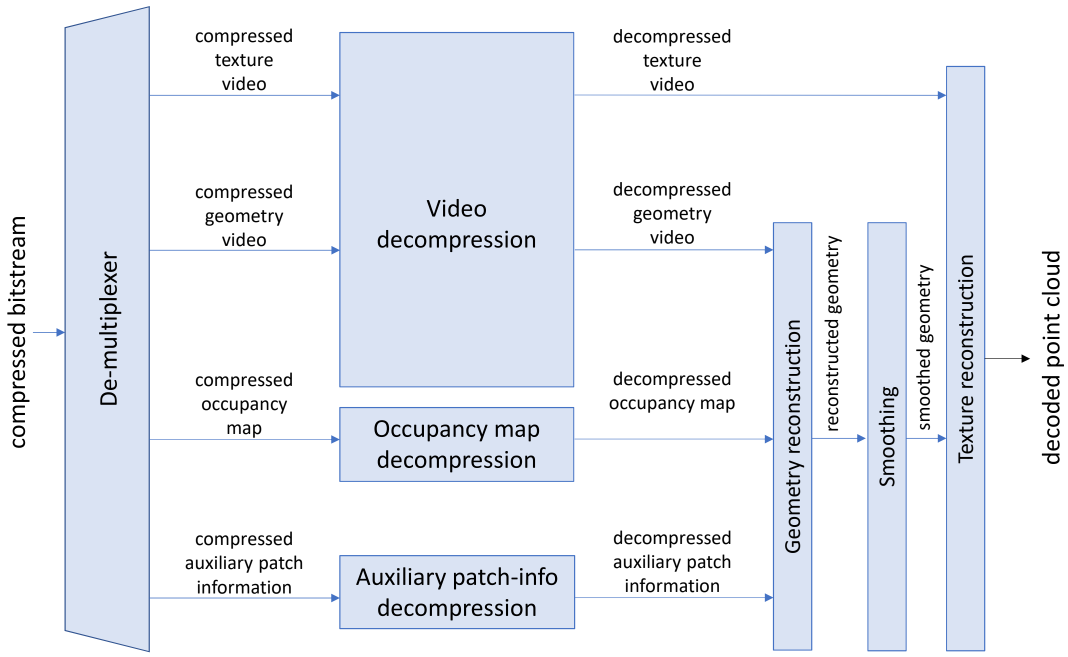
- Patch generation and packing processes: The main goal is to determine the best way to decompose the PC into patches and to efficiently fit them into a 2D grid through a simple orthogonal projection.
- Image generation and padding processes: The main goal is to transform the PC geometry and texture into 2D images temporally correlated in order to be ready for 2D video encoding.
- Auxiliary patch information and occupancy map generation: As referred before, there is a need to store patch/block metadata information such as the index of each patch projection plane, its 3D location, its 2D bounding box and the indexes of the patches that belong to each block in order to interpret the video sequences and allow correct reconstruction of the PC. This information is predicted and arithmetically encoded.

  An occupancy map is a binary map that indicates which grid cells are filled and which are empty. Empty blocks are detected and the remaining blocks are encoded with a certain user-defined precision.
- Smoothing module and reconstruction process: The main goal of the smoothing process is to smooth patch boundaries in order to minimize any possible discontinuity. For that, boundary points are moved to the centroid of their nearest neighbors.

  The reconstruction process is based on the occupancy map information in order to detect full pixels and compute their associated points using the auxiliary patch/block information.



(a) Encoder

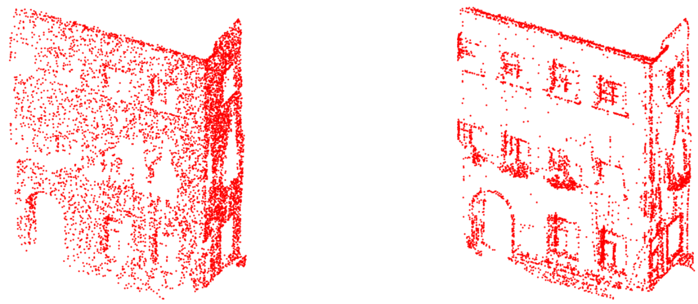


(b) Decoder

Figure 2.3: Overview of the MPEG V-PCC encoder and decoder from [2].

2.1.3 Fast Resampling on Point Clouds via Graphs

FRPC [3] is a graph-based method that selects a subset of points while keeping the contour information of the PC with a fidelity criterion set by the user as illustrated in Figure 2.4. This method takes advantage of the interaction between signals and graph structures. The graph captures local dependencies between points along with global ones and represents their surface.The point selection follows a non-uniform sampling that takes into account point utility estimates computed via specific features. The number of points is reduced by discarding less important ones but the positions and attributes of the remaining points do not change since there are no interpolations involved, only a simplification of the PC.



(a) Uniform resampling. (b) Contour-enhanced resampling.

Figure 2.4: Comparison between uniform and contour-enhanced resample from [3].

2.1.4 LASzip

Since the size of the LiDAR files increases with the sampling density of the sensor, compressing them without loss using LASzip [9] is a good choice because the size of the output files is usually less than one quarter of the original size. Its main goal is to turn LiDAR PC into more compact ones in order to enable faster transmission and have an easier management.

The LASzip compressor has the following characteristics [9]:

- lossless: compress the coordinates of the points without information loss;
- non-progressive: decompress data with the maximum details possible at a certain rate of resolution;
- streaming: compress chunks of data and outputs them when they are ready;
- order-preserving: preserve the original order of the points in the file;
- random-access: possible to decompress subsets of the points.

This compressor encodes the PC by chunks of points in order to allow random access. Since each chunk can have a different size, there is a chunk table, usually at the end of the file, which lists the starting byte of each chunk. For each new chunk, the first point is stored which is going to be used as the initial point of subsequent prediction schemes, and the entropy coder initialized. An entropy coder turns symbols into a stream of bits knowing about the distribution of symbols. At the beginning, it assumes the symbol distribution as uniform but then learns the symbol probability distribution. LASzip is based on an adaptive, context-base arithmetic coder. Also, a small chunk size means less compression because the adaptive entropy coder is reset at the start of each chunk and relearns all symbol distributions that negatively affect the compression. Below are the steps this compressor follows, for each chunk:

- Encodes a bit-mask of 6 bits that has information about whether certain attributes have changed when compared to the point before.
- All the attributes that have changed are encoded.
- Cartesian coordinates are then encoded. However they are not directly encoded, they are predicted from the previous points and the difference between them is entropy coded.

2.2 Point Cloud Segmentation Algorithms

The PC segmentation algorithms chosen for this dissertation are all based on point clustering in order to use some older methods. The methods chosen were selected according to criteria of computational complexity simplicity and clustering performance as well as availability of ready-to-use implementations.

2.2.1 K-Means

K-means [14] [15] segments the given data into an user-defined number of cluster, K. The steps of this algorithm are the following according [16]:

Algorithm 1: K-Means algorithm

```
   Input: k (the number of clusters), D (a set of points)
   Output: a set of k clusters
1  A cluster center (centroid) is chosen uniformly at random;
2  The distance from each point to the chosen centroid is computed;
3  repeat
4  |  The next centroid is chosen at random however with a distance-based probability which is
   |    proportional to its distance to the closest chosen centroid, i.e., farthest points have higher
   |    probability of being chosen;
5  |  The distance from each point to the new centroid is computed;
6  |  Points are assigned to the cluster that has the closest centroid;
7  until k centroids are selected;
8  repeat
9  |  The distance from each point to the centroid is computed using a squared Euclidean distance metric;
10 |  Points are assigned to the cluster that minimizes the distance from that point to the centroid;
11 |  The new centroid location is recalculated as the mean of the points in the cluster;
12 until the maximum number of iterations is reached or the clusters compositions do not change
     significantly;
```

Initializing centroids with the K-means++ algorithm [17] (step 1 to 7) improves the running time and the quality of the solutions. It was also demonstrated in [17] that K-means converges faster when centroids are initialized with this method.

2.2.2 Fuzzy C-Means

This algorithm [18] [19] partitions data into clusters with different levels of membership, i.e., each point belongs to all the clusters but with different membership weights. The algorithm follows the steps bellow:

Algorithm 2: Fuzzy C-Means algorithm

Input: k (the number of clusters), D (a set of points)
Output: a set of k clusters

```
Initialize D with a random membership level;
repeat
    Compute the cluster centers(centroids);
    Update membership grades based on the centroid distance to each point;
    Compute the objective function which is meant to be as small as possible since it represents the
      weighted distance from any point to the centroid;
until the objective function reduction is smaller than a predefined threshold or until the maximum
  number of iterations is reached;
```

This process is repeated because most likely the first computed centroid is incorrect and, updating them as well as the membership values, will move them to the correct location.

2.2.3 K-Medoids

This algorithm [20] partitions data into K clusters, where K is predefined. Its goal is the same as in K-means, i.e., minimize the sum of distances from the center to each point within each cluster. However, unlike the K-means case, the centers of the clusters are PC points. Points are assigned to a cluster by picking the cluster with the closest cluster center (medoid). As in the other methods used, this process is repeated until medoids do not change or until predefined conditions are met.

The algorithm used to find the medoids is the Clustering LARge Applications (CLARA) which applies the Partitioning Around Medoids (PAM) algorithm on random subsets and assigns each point to a cluster by picking the closest medoids. The steps of PAM algorithm are:

Algorithm 3: PAM algorithm

Input: k (the number of clusters), D (a set of points)
Output: a set of k clusters

```
Initial medoids positions are chosen;
Each cluster is associated to a medoid following the k-means ++ initialization algorithm explained in
  2.2.1;
Each point within each cluster is tested as a possible medoid by choosing the point with the lowest sum
  of distances within its cluster;
All points are assigned to the cluster that has the closest medoid;
```

2.2.4 Mean Shift

This method [21] partitions data into an algorithm-defined number of clusters. The steps of Mean Shift algorithm are the following:

Algorithm 4: Mean Shift algorithm

Input: D (a set of points), B (bandwith)
Output: a set of k clusters
repeat
    A random seed point is chosen which is used as the starting mean point;
    repeat
        Compute squared distances from the mean point to each point;
        Points which squared distance is within a given bandwidth will be assigned a vote, added to the cluster and flagged as visited;
        A new mean point is computed and the old one is saved;
    until *distance between the new and the old mean point is smaller than a defined threshold*;
    if *distance between the mean points of new cluster and existing ones is within half of B* **then**
        Merge clusters;
until *all points flagged as visited*;
Each point belongs to the cluster for it which has the most votes;

2.2.5 Clustering based on Euclidean Distance

PC data is segmented into clusters based on a minimum Euclidean distance [22] which represents the minimum distance between points from different clusters. Points with a smaller distance than this threshold belong to the same cluster. The algorithm does the following:

Algorithm 5: Clustering based on Euclidean Distance algorithm

Input: D (a set of points), R (user-defined range)
Output: a set of k clusters
repeat
    if *point not valid* **then**
        Point is removed;
    The neighbors within R of each point are computed, including the point itself;
    These points are assigned as belonging to the same cluster;
    if *any of these neighbors already belongs to a cluster* **then**
        The remaining neighbors will be assigned to that cluster;
    else
        A new cluster is created;
    end
until *repeating N times, being N the number of points in the PC*;

2.2.6 Clustering based on subtractive clustering

In subtractive clustering [23], each point is a potential cluster center. The process of this algorithm is the following:

Algorithm 6: Subtractive clustering algorithm

Input: D (a set of points), R (cluster influence range)
Output: a set of k clusters' centers

```
Based on the point neighbors, its probability of being a cluster center is computed;
The point with the highest potential to be the first cluster center is chosen ;
repeat
    The points near the first cluster center are removed and the vicinity is computed using the influence
      range;
    The remaining point with the highest potential as the next cluster center is chosen;
until all the data is within R of a cluster center;
```

2.3 Point Cloud Compression Quality Assessment

Errors between original and compressed PCs, i.e., the geometry distortion of the compressed point cloud when compared to the original one can be measured using :

- Point-to-point (P2Point) distance [4, 24]: every point in the original PC has a match in the compressed PC which is its closest neighbor. So it is necessary to find a point in the compressed PC (A) for each original PC (B) point. Then, their distance is computed as shown in Figure 2.5. These global distortion can be measured using:
  - Mean Squared Error (MSE) [11]: $\frac{\sum_{i=1}^{N} d(A_i,B_i)^2}{N}$, being N the number of points in the compressed PC.
  - Hausdorff distance: $max(d(A_i,B_i))$. It is the maximum of all distances.
  - Peak Signal-to-Noise Ratio (PSNR) [11]: represents the ratio between the highest possible value of a signal and the value of the degrading noise that affects it. It was computed using MSE and Hausdorff.
- Point-to-plane (P2Plane) distance [4, 24]: This approach is more complex than the P2Point metric because, after finding the original PC (A) point correspondence in the compressed PC (B) point, the error vector is computed and projected in the normal direction as shown in Figure 2.5. It can be obtained using:
  - MSE: $\frac{\sum_{i=1}^{N} E(i,j)^2 \times N_j}{N}$, being N the number of points in the compressed PC.
  - Hausdorff distance: $max(d(A_i,B_i))$, being B_i the closest point.
  - PSNR: It was computed using MSE and Hausdorff.

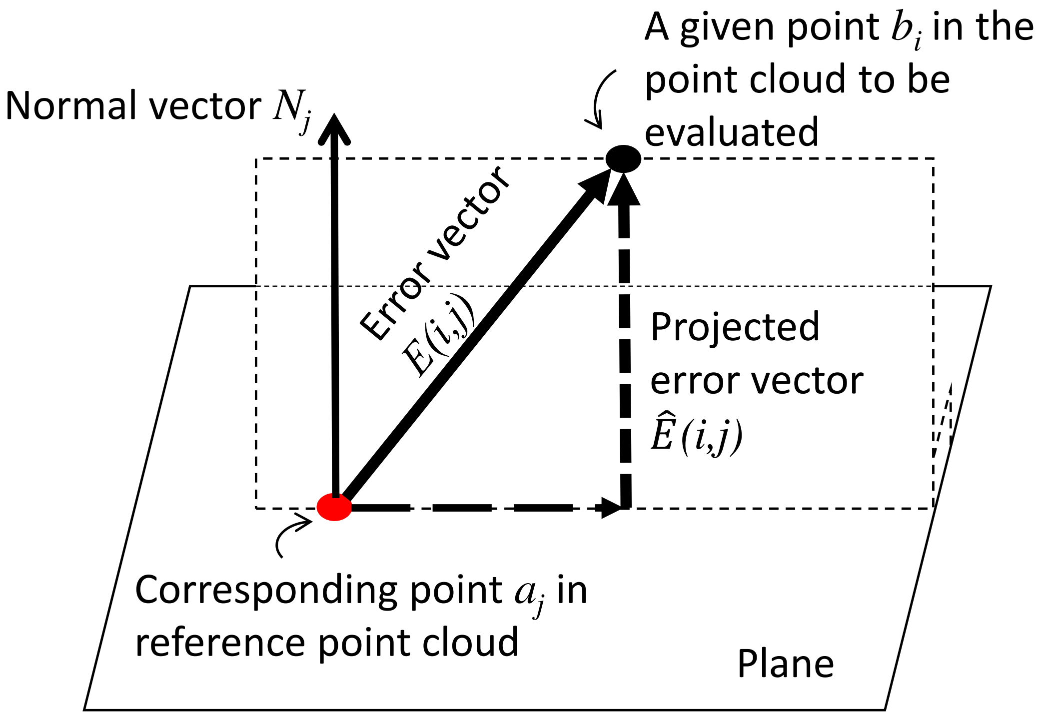


Figure 2.5: Difference between P2Point and P2Plane metrics from [4].

Chapter 3

Methodology

To perform the desired evaluations of the segmentation algorithms, three main tasks were done. First an uncompressed test point cloud was prepared based on some well known PCs presented in section 3.1.1. Then that PC was encoded using the encoders described in section 2.1 at different compression rates. Afterwards all the PCs, original and compressed, were processed with the segmentation/classification methods chosen and the results analyzed. The following sections describe all these steps in detail.

Also, as mentioned in Section 1, there is a need to divide these experiments into two sets of experiments: one related to general use PC data and the other involving a particular case of PCs, LiDAR data. The level of detail of the data is related to its use case which is likely to have different levels of compression without losing information, thus requiring two separate experiments with different requirements.

3.1 General use PC dataset

This section is divided into four subsections that describe which data was used, how it was prepared, which segmentation tests were performed and under what conditions, and which performance measures were used to evaluate the results obtained.

3.1.1 Point Cloud Test Data

Half of these experiments were based on the four dynamic PCs *longdress*, *loot*, *red and black*, and *soldier* taken from the 8i Voxelized Full Bodies (8iVFB) dataset [25]. The individual PCs in these dynamic PCs have been voxelized, having a 10 bit depth spatial resolution. These PCs, besides geometry information, contain color attributes. From these four PCs, frame 1300 of the *longdress* PC, frame 1200 of *loot*, frame 1550 of *red and black* and frame 690 of *longdress* were chosen to be our static PCs because these are the ones most used on the MPEG tests using static PCs.

The other half, which was chosen due to its simplicity, was based on static PCs from:

- two taken from the Stanford 3D scanning repository [26] (*bunny* and *dragon*) which were captured using a laser triangulation range scanner, more precisely Cyberware 3030 MS scanner.
- one from the activities of the MPEG standardization committee [27] (*egyptian_mask*).
- one from [28] (*vase*) which was captured by Intel RealSense R200.
- one from MeshLab being artificially produced [29] (*torus*)
- two synthesized using mathematical formulas [29] (*sphere* and *cube*).

Since a PC with multiple objects was needed to properly test the clustering algorithms, composite PCs were created by assembling PCs from the first four static ones while the remaining ones are assembled in others composite PCs and then voxelized using a 10 bit depth.

In Figure 3.1 are illustrated two examples of such assembling.

Figure 3.1: Representation of two composite PCs.

3.1.2 Data Preparation

Each tested PC was compressed using MPEG V-PCC, MPEG G-PCC, and FRPC with the encoding parameters listed in Table 3.1. The values to each parameter were chosen in order to cover as many cases as possible. In MPEG G-PCC and FRPC, - means that the quality level does not exist.

Table 3.1: Configurations used in MPEG V-PCC, MPEG G-PCC, and FRPC

	MPEG V-PCC		MPEG G-PCC	FRPC
Quality level	**GeometryQP**	**LosslessGeo**	**PosQuantScale**	**Density**
q01	51	0	0.125	10
q02	48	0	0.25	20
q03	44	0	0.375	30
q04	40	0	0.4375	40
q05	36	0	0.5	50
q06	32	0	0.5625	60
q07	28	0	0.625	70
q08	24	0	0.6875	80
q09	20	0	0.75	90
q10	16	0	0.875	100
q11	12	0	0.9375	-
q12	8	0	1	-
q13	4	0	-	-
q14	0	0	-	-
q15	-	1	-	-

The parameters listed have the following meanings and characteristics:

- **MPEG V-PCC**
  - GeometryQP: geometry information quantization stepsize. It has a range from 0 to 51.
  - LosslessGeo: binary flag indicating that geometry is encoded without loss.
- **MPEG G-PCC**
  - PosQuantScale: position quantization scale parameter, larger values mean finer encoding. It has a range from 0 to 1.
- **FRPC**
  - Density: FRPC subsampling ratio. It has a range from 0 to 100.

3.1.3 Segmentation Tests

The uncompressed and compressed PCs were segmented using the segmentation algorithms listed in section 2.2. Their already existent MATLAB implementations were used with configuration parameters set according Table 3.2. Some of these parameters were set as default by the algorithms, however others were adjusted in order to correctly identify the objects in each original PC. This adjustment was made by trial and error aided by visual confirmation of the segmentation of PCs. However, this was only made for the original PCs because the compressed ones use the same values as their reference PC.

Table 3.2: Configuration parameters used in segmentation tests.

Method	Parameter	Value
K-Means	Distance metric	Squared Euclidean distance
	Maximum number of iterations	100
	Number of times to repeat clustering using new initial cluster centroid positions	1
	Centroid initialization method	K-means++ algorithm
Fuzzy C-means	Maximum number of iterations	100 or 200
	Minimum improvement in objective function between consecutive iterations	1E-05
	Amount of fuzzy overlap between clusters	2
K-medoids	Algorithm to find medoids	CLARA
	Distance metric	Squared Euclidean distance
	Maximum number of iterations	100 or 285
	Number of times to repeat clustering using new initial cluster centroid positions	5
	Number of samples to take from data	40+2*number of clusters
	Medoid initialization method	k-means++ algorithm
Mean Shift	Bandwith parameter	26 or 50 or 300
Pcsegdist	Minimum Euclidean distance	10 or 25 or 100
Subclust	Range of influence of the cluster center	0.6 or 0.65 or 0.8

3.1.4 Performance Measures

To evaluate the compression results, the measures presented in section 2.3 are used. Besides that, other measures such as bitrate and scaling ratio are used. The bitrate represents the number of bits per pixel in the compressed frame. To obtain it, it is necessary to divide the number of bits in the compressed frame by the number of points in the original one. However, for FRPC, bitrate information is not available as this method only provides a PC simplification, reducing the number of points according certain defined parameters. The scaling ratio represents the relationship between the input and the output file.

To assess the results of the segmentations performed, each cluster of the segmented PC was processed to compute some key indicators listed in Table 3.3.

Value	Definition
C	Cluster Center
C_{orig}	Corresponding Original PC Cluster
DC	$distance(C, C_{orig})$
CH	Cluster Convex Hull
CH_{orig}	Corresponding Original PC CH
DCH	Distance from CH to CH_{orig}
V	CH Volume
V_{orig}	Corresponding Original PC CH
AVD	$\mid V - V_{orig} \mid$
PAVD	$\frac{AVD}{V_{orig}} \times 100$
P	Cluster Number of Points
P_{orig}	Corresponding Original PC Cluster
NP	$\mid P - P_{orig} \mid$
PNP	$\frac{NP}{NP_{orig}} \times 100$

Table 3.3: Clustering performance measures.

The accuracy of the clustering is measured by several values. The first is the agreement in the number of clusters in the original (uncompressed) PC and the number of clusters obtained on each of the reconstructed PCs. When the number of clusters in the original and compressed PC match, one can measure the degradation of the clustering by the values indicated in Table 3.3, namely the distance between the centers (DC) of the matching clusters in the original and compressed PC, as well as the absolute difference of volumes (AVD) and percentage of absolute difference of volumes (PAVD) of the convex hull of the matching clusters, the percentage of absolute difference of number of points (PNP) of the matching clusters and the distance between the convex hulls (DCH) of the matching clusters.

The algorithm which clustering is based on Euclidean distance, Pcsegdist, does not give any information related to the clusters' center, however, as the information of the points of each cluster is available, it was possible to compute the mean of each cluster and use it as cluster center. On the other hand, Subclust, which is based on subtractive clustering, only returns centers of clusters and DC as presented in Table 3.4.

Table 3.4: Performance measures used in the segmentation algorithms.

	C	**PNP**	**AVD**	**PAVD**	**DCH**	**DC**
K-Means	Yes	Yes	Yes	Yes	Yes	Yes
K-Medoids	Yes	Yes	Yes	Yes	Yes	Yes
Fuzzy C-Means	Yes	Yes	Yes	Yes	Yes	Yes
Mean Shift	Yes	Yes	Yes	Yes	Yes	Yes
Pcsegdist	Yes	Yes	Yes	Yes	Yes	Yes
Subclust	Yes	No	No	No	No	Yes

3.2 LiDAR Dataset

This section is divided into four subsections that describe which data was used, how it was prepared, which segmentation/classification tests were performed and under what conditions, and which performance measures were used to evaluate the results obtained.

3.2.1 Point Cloud Test Data

The KITTI dataset [30] was recorded in Germany using a moving plataform. The PCs used in this dissertation were obtained from a Velodyne 3D laser scanner that continuously rotates around its axis at 10Hz using 64 beams. Each scan is stored as a floating point binary and each point is stored as a cartesian coordinate together with a reflectance value which was not used in this dissertation. Each PC has a different number of points. An example of a projection of a frame of this dataset is presented in Figure 3.2.

Since voxelization was needed, several tests were performed in order to realize the most effective bit depth. The first tests used a 10, 15, and 20 bit depth while the second ones used a 11, 12, 13, 14, 15, and 20 bit depth. As MPEG V-PCC does not allow input PCs with a bit depth higher than 10, this method was not used in the majority of the cases.

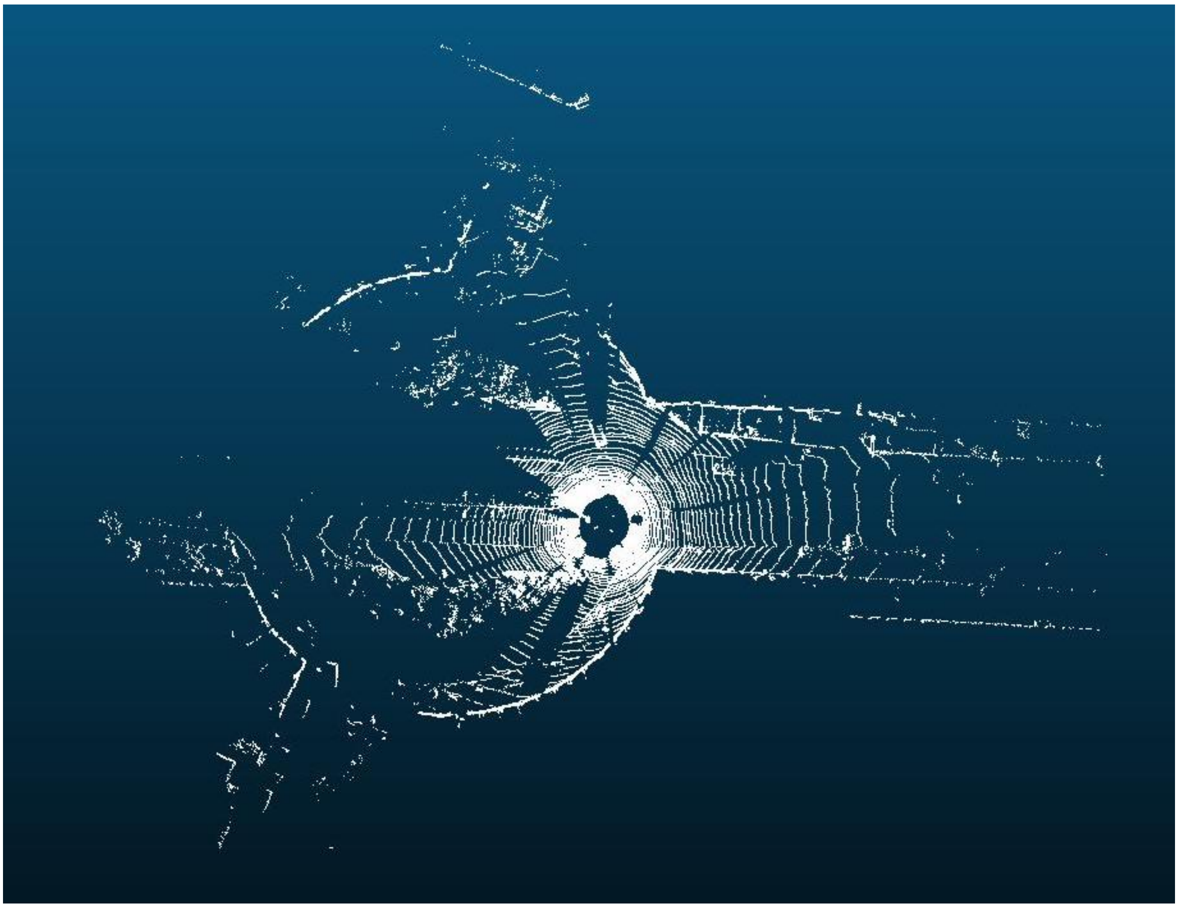


Figure 3.2: Projection of a frame from the KITTI dataset.

3.2.2 Data Preparation

In a first approach, three bit depths (10, 15, and 20) were used and each tested PC was compressed using MPEG V-PCC (only for a 10 bit depth), MPEG G-PCC, and FRPC and LASzip using the encoding parameters listed in Table 3.1.

Later, for purposes of segmentation and classification, it was decided to use 11, 12, 13, 14, 15 and 20 bit depth with the quality levels presented in Table 3.5. The range of the compression rates was reduced to a low, two medium and a high level due to the complexity of the segmentation methods that were going to be used.

Table 3.5: Configurations used with MPEG G-PCC and FRPC for purposes of segmentation and classification

Quality level	PosQuantScale	Density
q01	0.25	30
q02	0.375	50
q03	0.5625	70
q04	0.75	90

The parameters listed have the same meaning and characteristics as in Section 3.1.2.

3.2.3 Segmentation and Classification Tests

Before any other operation, it was necessary to train the feedforward neural network using some already implemented MATLAB functions. After that, the network was ready to perform the segmentation/classification tests needed.

First, three objects were identified in each original PC (Van, Pedestrian and Bicycle) and their bounding box features taken. Then, to create a cluster for each object in the compressed PCs, it was necessary to use that bounding box features to identify the objects and extract their points and real features. In the final step, each cluster is assigned a label, i.e., is classified. In Figure 3.3 is illustrated an overview of theses processes.

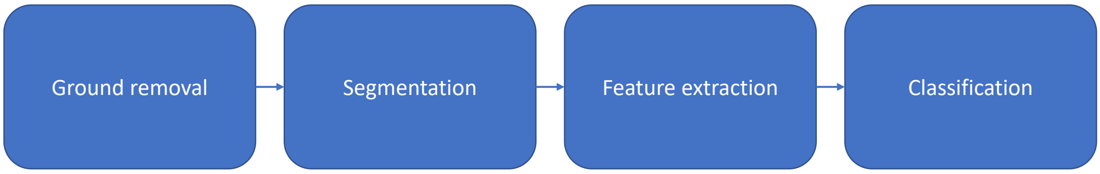


Figure 3.3: Flowchart of the approach followed.

3.2.4 Performance Measures

Another possible approach to evaluate the original and compressed PC is to overlap them in order to know the percentage of equality they have. For this, it is used a 2.5D grid map where each cell represents a height [31]. So the information is upsampled because it can also use information from the neighbor cells in order to make the surface smoother.

To assess the results of the segmentations and classifications performed, each original cluster and its label is compared to the predicted one and a confusion matrix is constructed which is evaluated using three parameters [32]:

- Error: ratio between the total of incorrect predictions and the total number of the dataset.
- Specificity: ratio between the total of correct negative predictions and the total of negatives in the dataset.
- Precision: ratio between the total of correct positive predictions and the total number of positive predictions.

Chapter 4

Results and Analysis

This chapter will be divided into two sections. One related to compression results and the other one focused on the segmentation results.

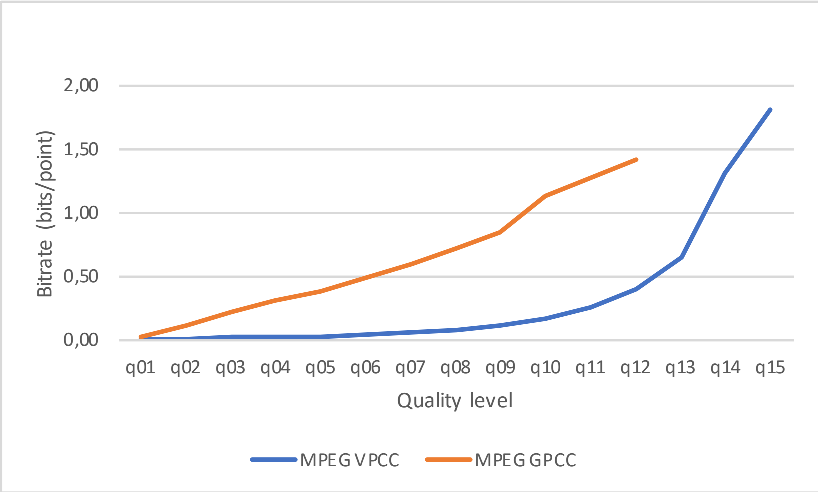
4.1 Compression Results

As mentioned before, this section will be divided into two subsections which correspond to the two different datasets used.

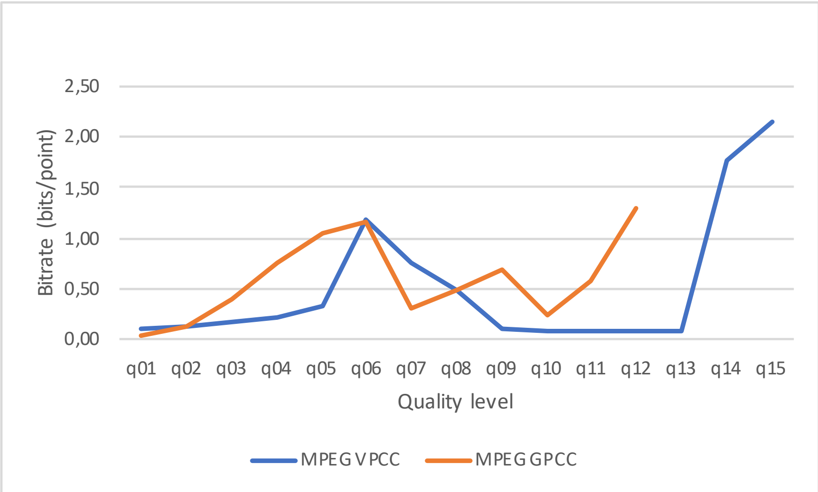
4.1.1 General use PC dataset

All the results expressed use the performance measures referred in Section 3.1.4. Three experiments are presented in this subsection: one was performed with a composite PC using the 8iVFB dataset which is called from now on experiment 1, experiments 2 and 3 are a composite PC using the remaining PCs as explained in Section 3.1.1. The main difference between experiment 2 and 3 is that 3 was a bigger distance between objects. In this case, the main goal is to study whether distance influences compression or not.

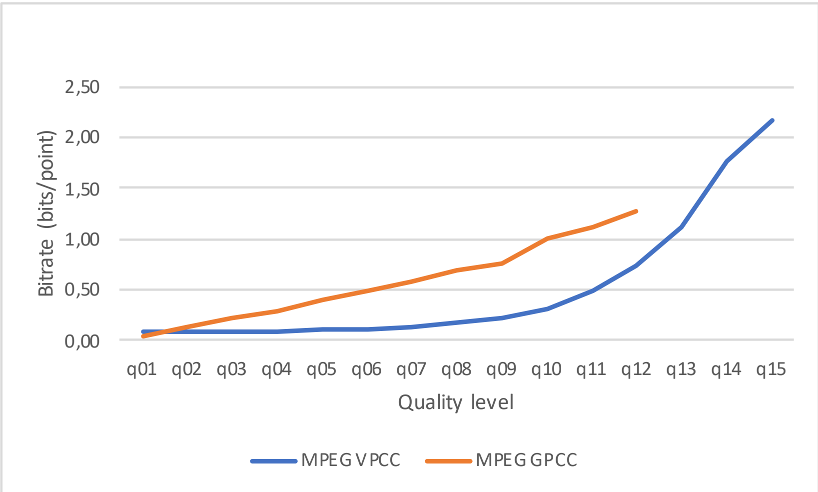
In Tables A.1, A.2, and A.3 are represented the values related to bitrate and scaling ratio. From that and from Figure 4.1, it is possible to conclude that, as expected because the quality levels were defined like that, the bitrate increases along with the quality level in most of the cases. However, in the experiment where objects are too close to each other, this did not happen which can be explained by geometry distortions introduced by the PCC due to the proximity of the objects. This rise is exponential in MPEG V-PCC and linear in MPEG G-PCC due to the algorithms' characteristics. Bitrate results for FRPC are not available as not proper coding is performed, only a PC simplification as explained in Section 2.1.3.



(a) Experiment 1



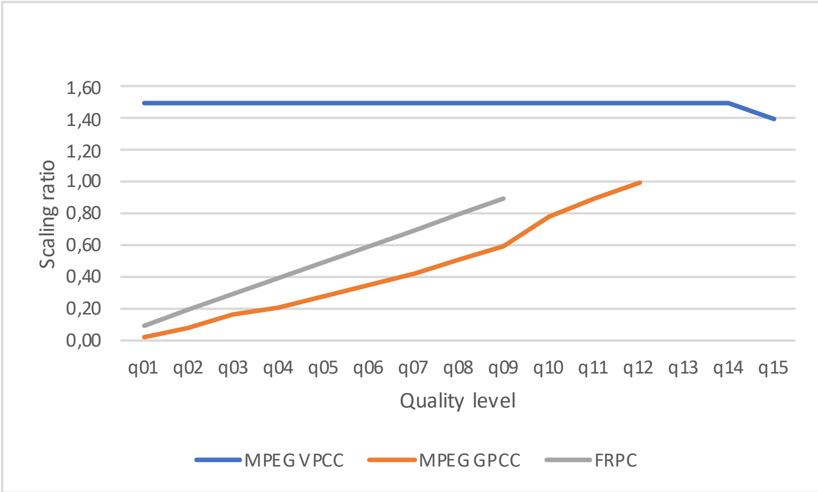
(b) Experiment 2



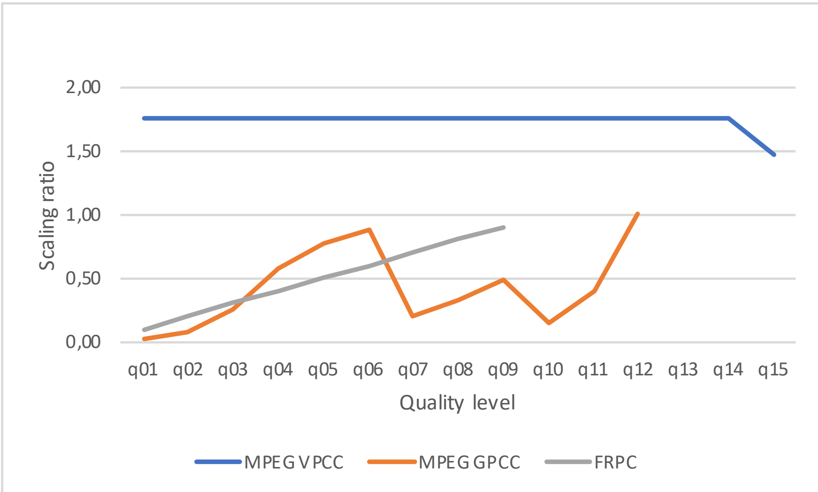
(c) Experiment 3

Figure 4.1: Representation of the bitrate results.

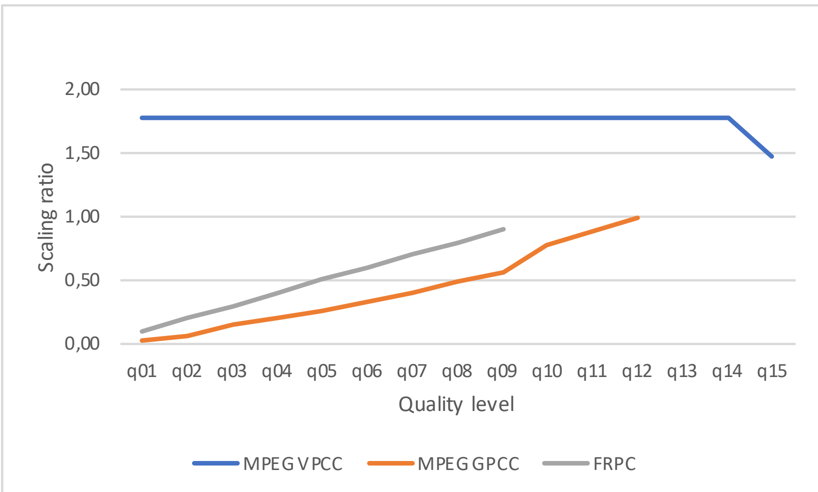
The scaling ratio is constant for MPEG V-PCC and rises linearly for MPEG G-PCC and FRPC as shown in Figure 4.2. This happens because, in MPEG G-PCC, the number of points increases along with the quality level as well as in FRPC while in MPEG V-PCC, the number of points is the same in every quality level except in the lossless case which decreasees due to the algorithm characteristics.



(a) Experiment 1



(b) Experiment 2



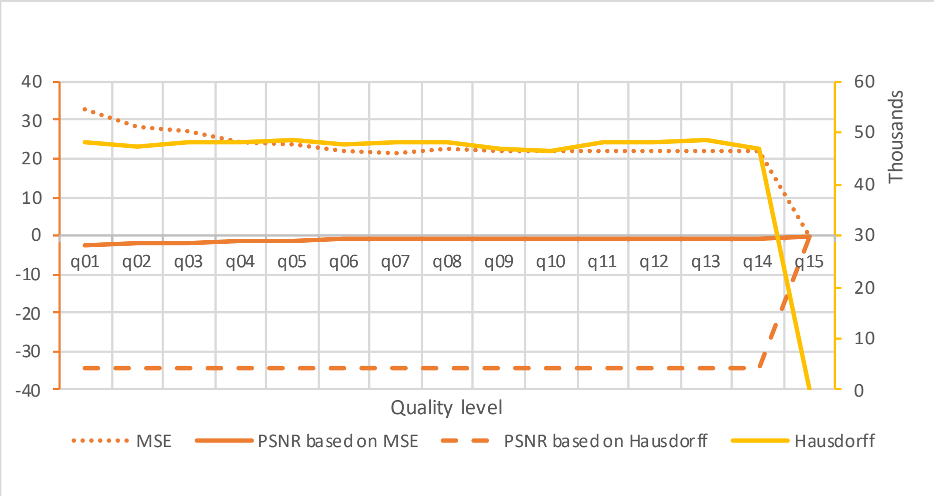
(c) Experiment 3

Figure 4.2: Representation of the scaling ratio results.

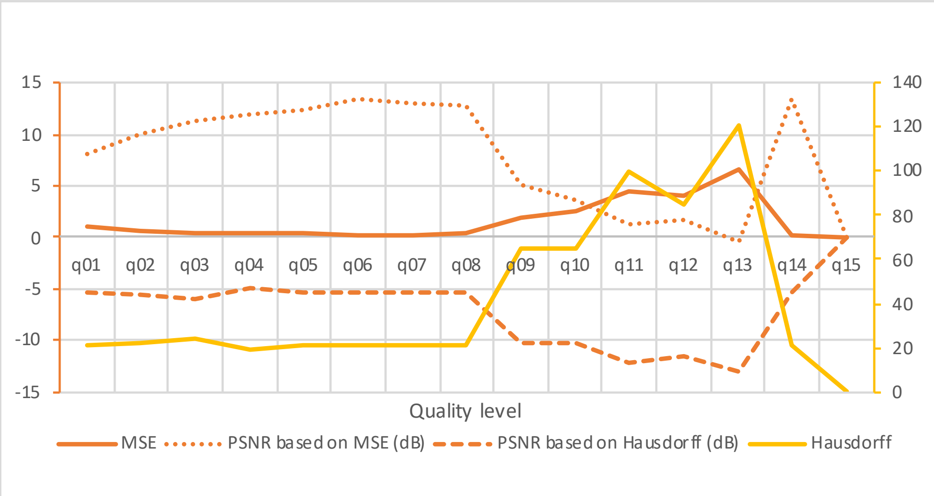
In Tables A.4, A.5, A.6, A.7, A.8,A.9, A.10, A.11 and A.12 are represented values related to P2Point and P2Plane metrics which results are discussed bellow.

The lossless cases which are q12 for MPEG G-PCC, q10 for FRPC, and q15 for MPEG V-PCC are not discussed here because their values are usually zero or infinite.

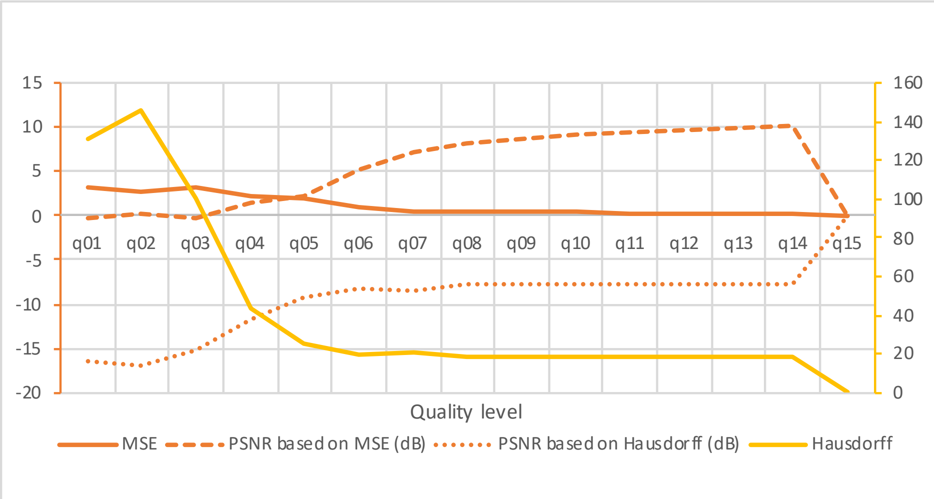
In P2Point using MPEG V-PCC, as shown in Figure 4.3, MSE usually decreases along with the quality level.However, there are no conclusions for what happens in experiment 2 because usually a small value means higher quality and lower error which doesn't correspond to what was suppose to happen since the higher quality levels are the ones with higher MSE. As expected after the analysis of MSE, its PSNR increases along with the quality level except for experiment 2. Hausdorff has a higher range in comparison with the other values, but it does not show a pattern as each experiment does something different. This also happens for its PSNR.



(a) Experiment 1



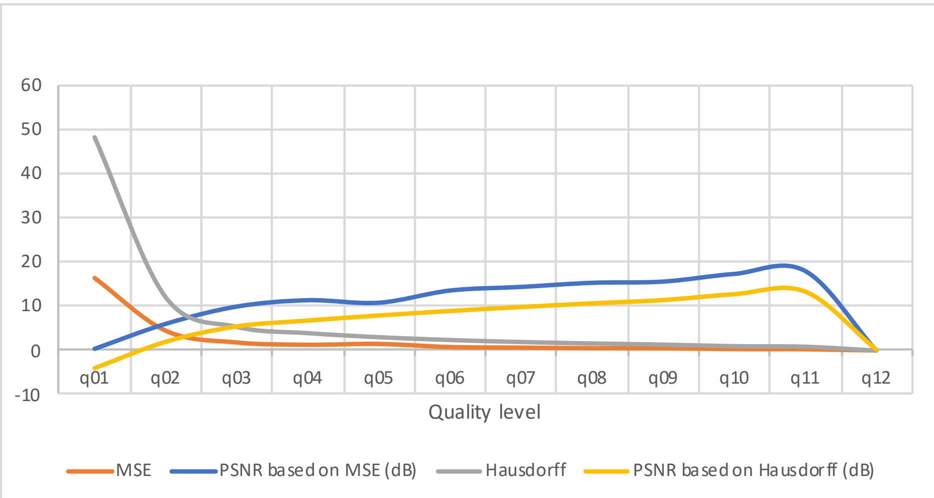
(b) Experiment 2



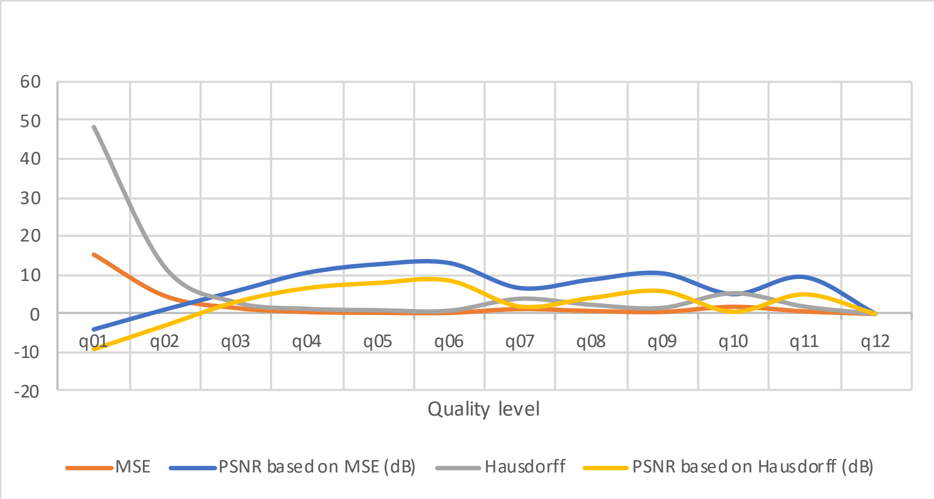
(c) Experiment 3

Figure 4.3: Representation of the P2Point metric results using MPEG V-PCC.

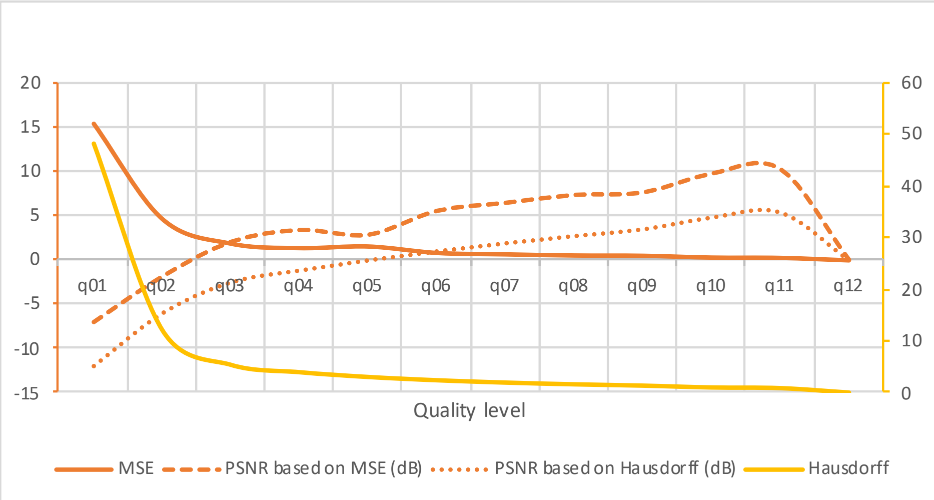
In P2Point using MPEG G-PCC and FRPC, MSE decreases with the quality level and its PSNR increases, as shown in Figure 4.4. Hausdorff also decreases and its PSNR also increases. This values are expected because a higher PSNR value usually means higher quality. It is also important to refer that PSNR of lossless cases is infinity however in the graphics bellow is represented by zero.



(a) Experiment 1

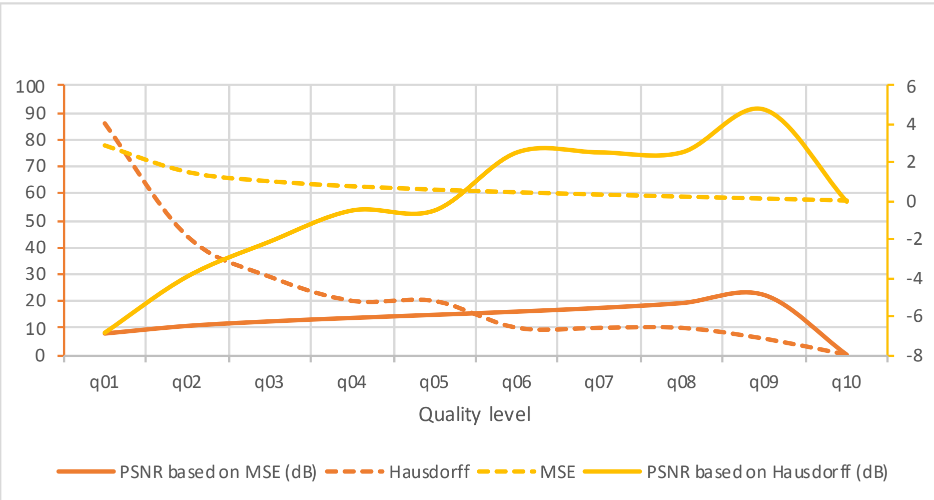


(b) Experiment 2

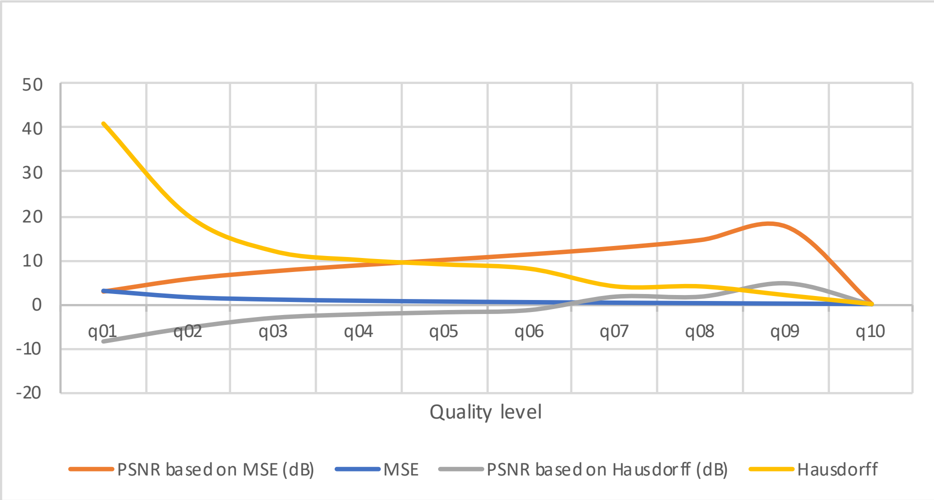


(c) Experiment 3

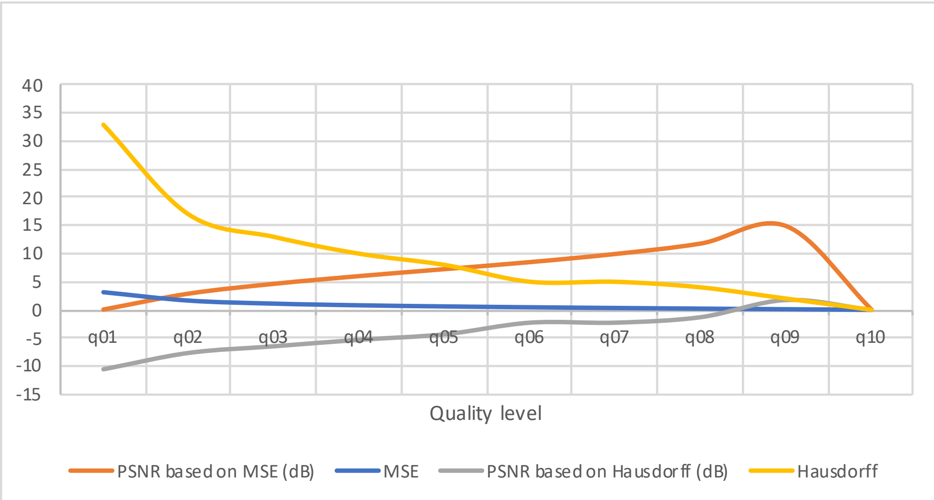
Figure 4.4: Representation of the P2Point metric results using MPEG G-PCC.



(a) Experiment 1



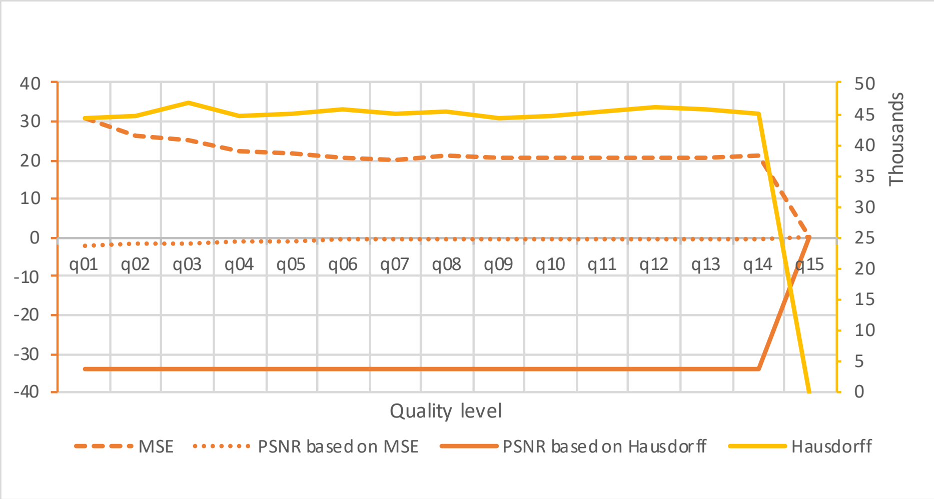
(b) Experiment 2



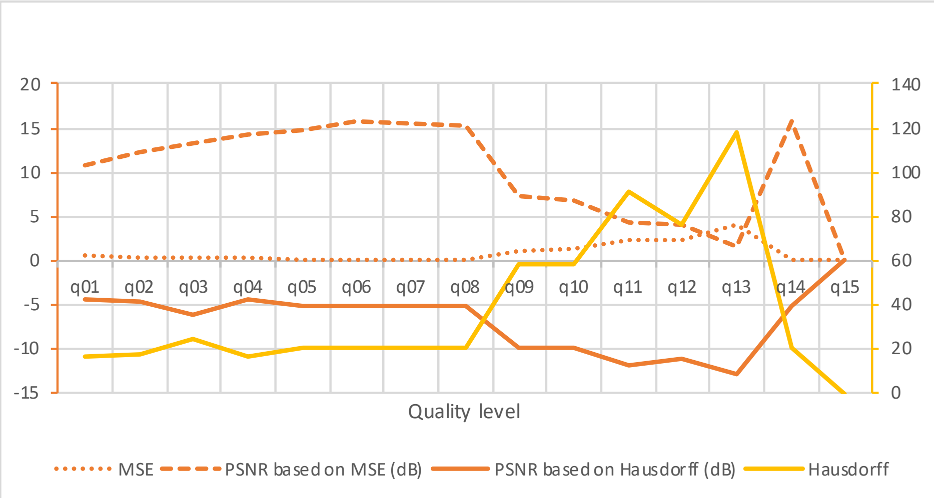
(c) Experiment 3

Figure 4.5: Representation of the P2Point metric results using FRPC.

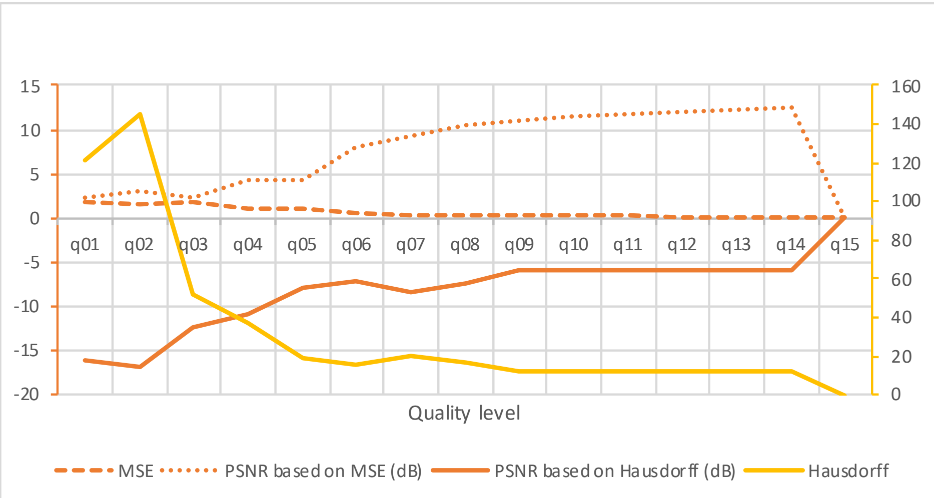
The behavior of P2Plane is similar to P2Point behavior as shown in Figures 4.7, 4.6 and 4.8. So what was explained before, can also be applied in this case.



(a) Experiment 1

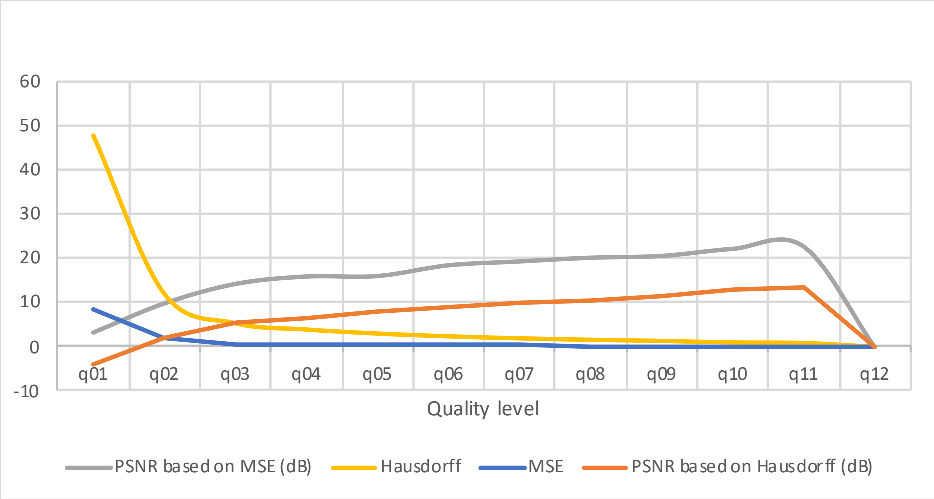


(b) Experiment 2

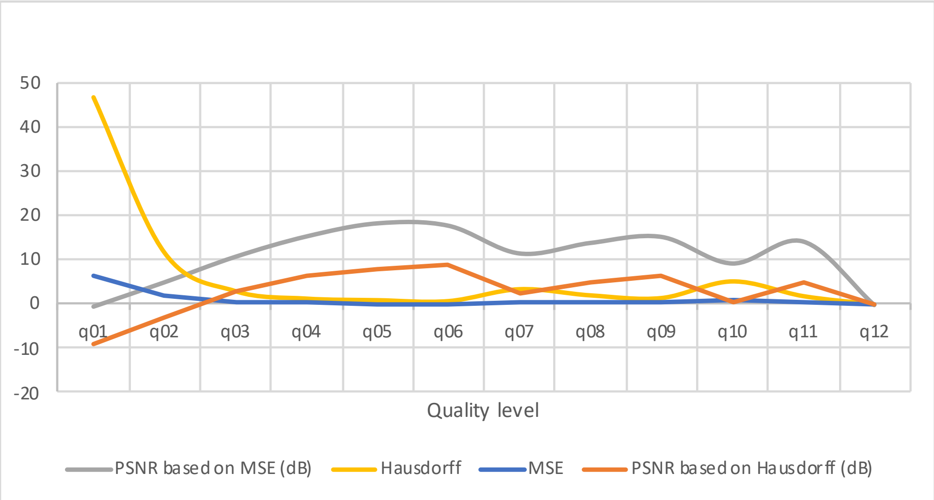


(c) Experiment 3

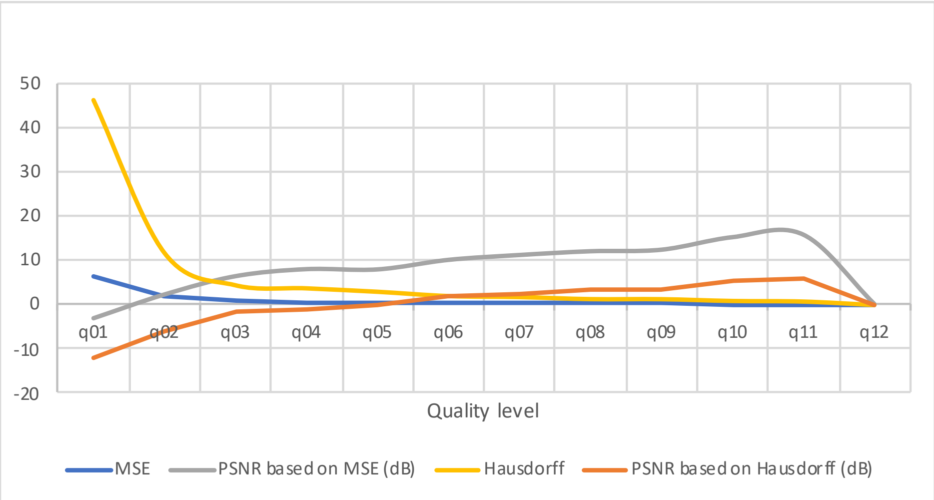
Figure 4.6: Representation of the P2Plane metric results using MPEG V-PCC.



(a) Experiment 1



(b) Experiment 2

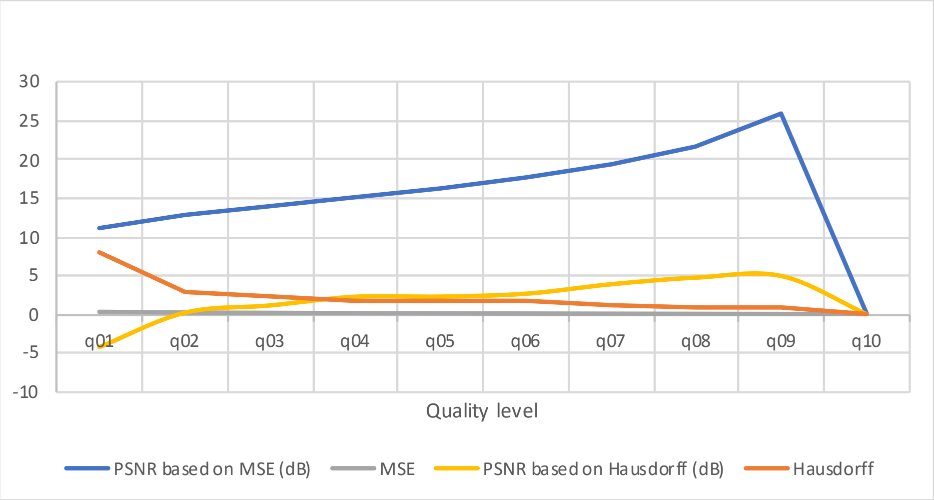


(c) Experiment 3

Figure 4.7: Representation of the P2Plane metric results using MPEG G-PCC.

(a) Experiment 1

(b) Experiment 2



(c) Experiment 3

Figure 4.8: Representation of the P2Plane metric results using FRPC.

It is possible to conclude that distance between objects does influence compression specially when using MPEG V-PCC, since the results of experiment 2 sometimes have fluctuations and erratic behavior.

4.1.2 LiDAR dataset

All the results are expressed using the performance measures referred in Section 3.2.4. Three experiments are presented in this subsection: one using a 10 bit depth, other using a 15 bit depth and the remaining one using a 20 bit depth. In this case, the main goal is to study whether bit depth influences compression or not.

For MPEG V-PCC, only some compression ratios were tested and for the remaining, the overlap was assumed to be zero in order to build a proper graphic. As it can be inferred from Figure 4.9 and Table 4.1, a 10 bit depth has a 2.5D overlap close to zero which means that this configuration has lost almost all the information that represents the frame used in this work. However, when using a 2.5D overlap with neighborhood, results improved although not surpassing the 25%. It is possible to conclude that is not worthy to segment this PC since the most part of the information was lost meaning that a 10 bit depth is low for this specific PC.

Table 4.1: Overlap results using a 10 bit depth PC.

		overlap 2.5D	**overlap 2.5D with neighborhood**
MPEG G-PCC	**q01**	0.01%	3.94%
	q02	0.04%	9.89%
	q03	0.04%	6.49%
	q04	0.10%	11.68%
	q05	0.10%	10.45%
	q06	0.09%	8.94%
	q07	0.21%	14.09%
	q08	0.18%	11.38%
	q09	0.17%	11.16%
	q10	0.24%	12.56%
	q11	0.44%	19.10%
	q12	0.45%	17.53%
MPEG V-PCC	**q06**	0.53%	12.83%
	q07	0.49%	11.86%
	q08	0.44%	10.97%
	q09	0.43%	10.33%
	q10	0.41%	9.84%
	q15	0.45%	17.52%
FRPC	**q01**	0.10%	23.23%
	q02	0.14%	21.50%
	q03	0.19%	20.72%
	q04	0.20%	19.88%
	q05	0.26%	18.12%
	q06	0.28%	17.65%
	q07	0.31%	17.01%
	q08	0.33%	16.78%
	q09	0.37%	16.90%
	q10	0.45%	17.53%

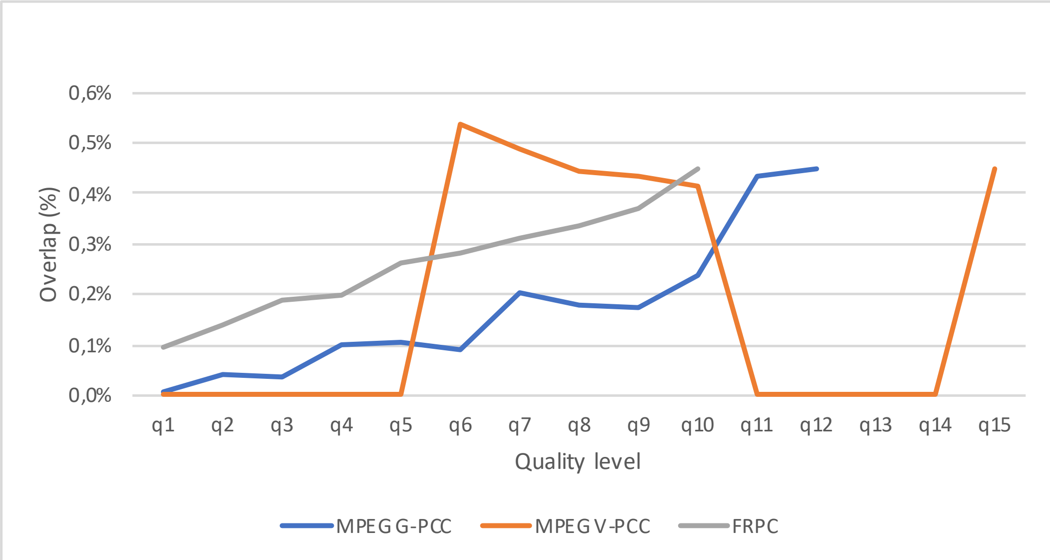


Figure 4.9: Representation of the overlap between the original and decompressed PC using a 10 bit depth.

As mentioned before, MPEG V-PCC does not support a bit depth higher than 10, so for higher bit depths, information related to this method is not available.

As it can be inferred from the Table 4.2, a 15 bit depth has a 2.5D overlap lower than 50%. As expected, this configuration has more information than the 10 bit depth dataset. When using a 2.5D overlap with neighborhood, results improved although not surpassing the 80%. Despite the fluctuations in MPEG G-PCC which can be due to geometric distortions introduced by the compression, this PC is a good example to segment because, as shown in Figure 4.10, along with the increasing compression ratio, there is a 2.5D overlap rise which although not being the ideal scenario, it is close.

Table 4.2: Overlap results using a 15 bit depth PC.

		Overlap 2.5D	Overlap 2.5D with neighborhood
MPEG G-PCC	**q01**	4.11%	40.21%
	q02	23.24%	63.40%
	q03	37.41%	72.09%
	q04	38.95%	73.39%
	q05	48.06%	79.55%
	q06	41.58%	75.54%
	q07	54.78%	78.12%
	q08	42.56%	77.11%
	q09	52.53%	80.51%
	q10	48.83%	78.64%
	q11	39.40%	78.43%
	q12	48.55%	79.02%
FRPC	**q01**	8.39%	77.91%
	q02	15.24%	78.29%
	q03	20.87%	78.32%
	q04	25.62%	78.31%
	q05	29.81%	78.35%
	q06	33.56%	78.33%
	q07	37.15%	78.29%
	q08	40.58%	78.23%
	q09	44.20%	78.26%
	q10	47.81%	78.27%

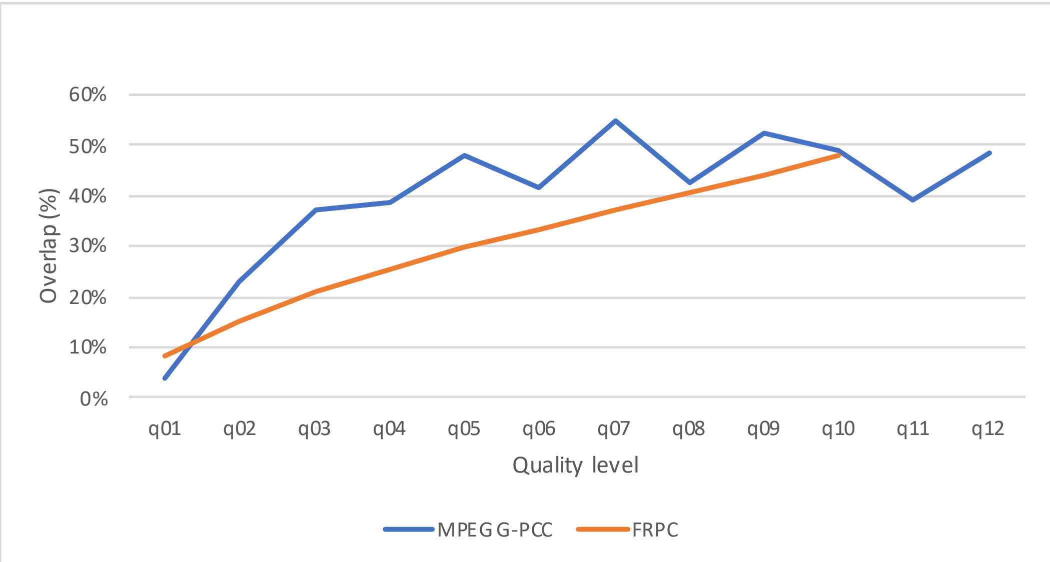


Figure 4.10: Representation of the overlap between the original and decompressed PC using a 15 bit depth.

As it can be inferred from Figure 4.11 and from Table 4.3, a 20 bit depth has a 2.5D overlap close to 90 for MPEG G-PCC which means that this configuration has almost all the information that represents the frame used in this work for all the compression rates. For FRPC, the ideal scenario was found due to the 10% growth of the overlap from each compression ratio to the next one. When using a 2.5D overlap with neighborhood, is possible to improve results. It is not worthy to segment the MPEG G-PCC PCs since the overlap is similar for all compression ratios.

Table 4.3: Overlap results using a 20 bit depth PC.

		Overlap 2.5D	**Overlap 2.5D with neighborhood**
MPEG G-PCC	**q01**	89.76%	95.73%
	q02	93.83%	96.97%
	q03	94.42%	97.31%
	q04	93.21%	97.33%
	q05	93.92%	97.61%
	q06	93.91%	97.44%
	q07	93.46%	97.47%
	q08	93.42%	97.48%
	q09	94.02%	97.62%
	q10	94.09%	97.60%
	q11	93.43%	97.53%
	q12	93.39%	97.53%
FRPC	**q01**	12.53%	91.28%
	q02	24.39%	92.58%
	q03	35.23%	93.53%
	q04	45.41%	94.31%
	q05	54.66%	94.81%
	q06	63.40%	95.37%
	q07	71.25%	95.88%
	q08	78.98%	96.22%
	q09	86.02%	96.64%
	q10	92.74%	97.05%

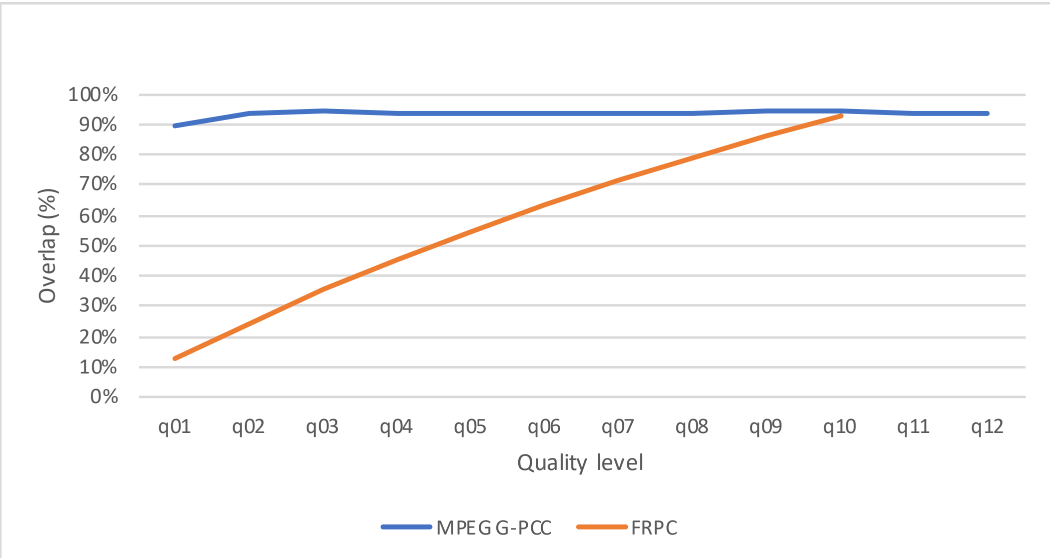


Figure 4.11: Representation of the overlap between the original and decompressed PC using a 20 bit depth.

4.2 Segmentation Results

As mentioned before, this section will be divided into two subsections which correspond to the two different datasets used.

4.2.1 General use PC dataset

Tables in Section B list the results of experiment 1 for the different compression rates and different clustering algorithms and encoder methods as explained in Section 3.1. All the results are expressed using the performance measures defined in Chapter 3. The experiments presented in this Section are the ones also presented in Section 4.1.1. The Figures presented are from cluster 1, however the remaining clusters usually present the same behavior.

For the K-Means algorithm, there are usually cases that failed to converge in the number of iterations defined or cases for which the clusters were not properly compared due to different cluster centers. For the Mean Shift algorithm, the number of clusters for the PC compressed usually using MPEG V-PCC sometimes is different from the original. The same happened with Pcsegdist and subclust. This means that the segmentation algorithm failed due to the distortions introduced by the compression. One way to obtain better results is to distance objects from each other, which is analyzed by comparing experiment 2 and 3.

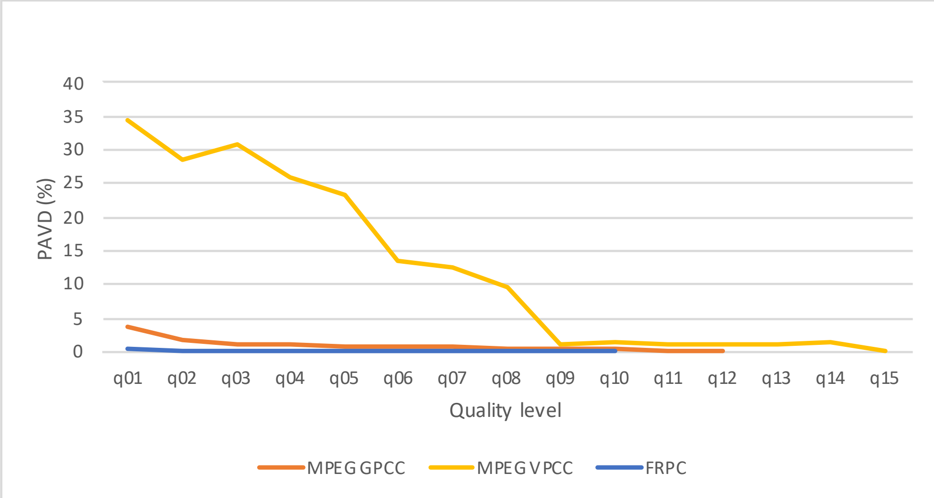
The center of each cluster is computed except for Subclust as referred in Table 3.4. K-medoids' centers vary the most which can induce errors in these performance measures.

Sometimes, even though centers seem to be well computed, visually, objects are wrongly segmented which can explain some inconclusive behavior in the performance measures used. Table 4.4 represents the number of PC with visual differences when compared to the original one.

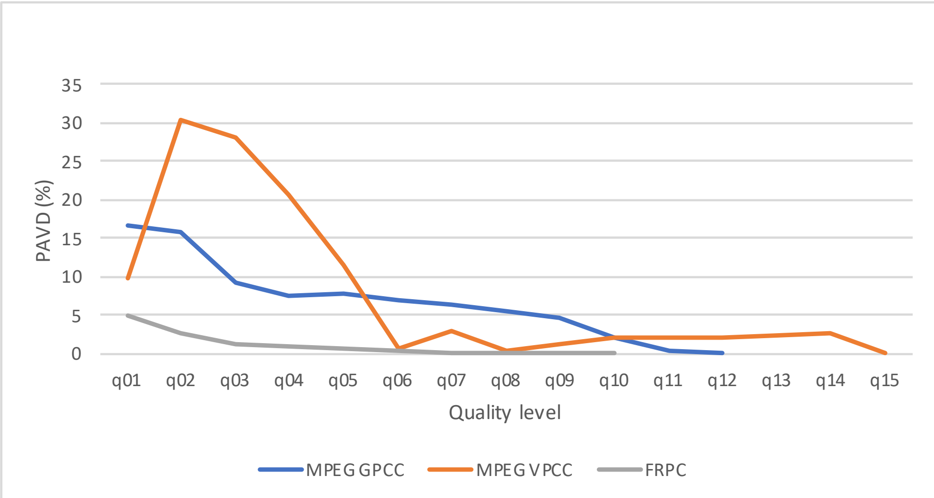
Table 4.4: Number of PC with visual differences to the original one

		K-Means	**K-Medoids**	**Fuzzy C-Means**	**Mean Shift**	**Pcsegdist**
Experiment 1	**MPEG VPCC**	2	9	0	14	14
	MPEG GPCC	3	7	0	2	0
	FRPC	6	3	0	1	0
Experiment 2	**MPEG VPCC**	12	15	15	13	0
	MPEG GPCC	4	12	0	6	1
	FRPC	5	10	0	4	0
Experiment 3	**MPEG VPCC**	1	0	6	0	0
	MPEG GPCC	1	0	0	0	0
	FRPC	0	0	0	0	0

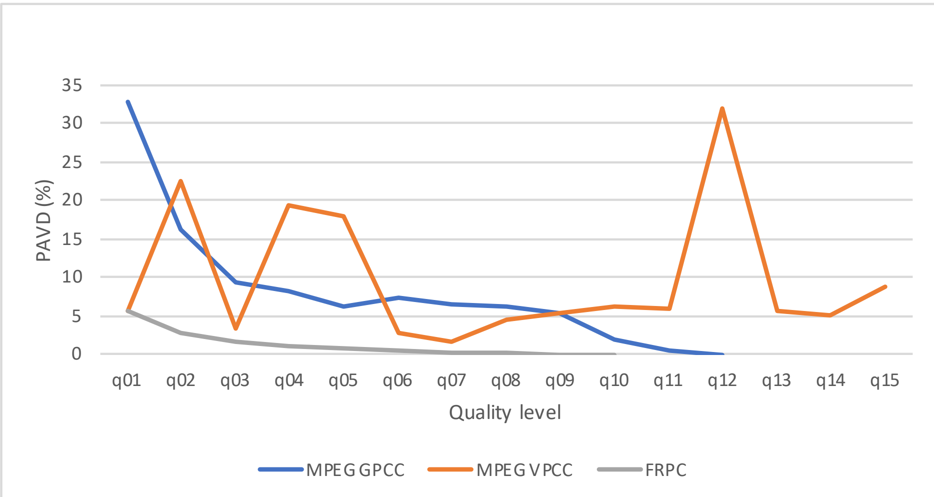
For the Fuzzy C-Means, it was observed according Figure 4.12a that, in experiment 1, PAVD decreases with increasing quality level for MPEG G-PCC, as it happens for the other two PCC methods although for a smaller range. However, for experiments 2 and 3 using MPEG V-PCC, PAVD fluctuates as shown in Figures 4.12b and 4.12c. Observing Figures 4.12e and 4.12f, it is possible to conclude that something similar happened for Pcsegdist algorithm although experiment 1 is constant for FRPC and MPEG G-PCC and decreases for MPEG V-PCC as shown in Figure 4.12d. However, it was not possible to conclude the same for the other methods for which was not possible to find any consistent relationship as shown in Figures 4.12g, 4.12h, 4.12i, 4.12j, 4.12k, 4.12l, 4.12m and 4.12n although, K-medoids for experiment 3, seems to behave just like Fuzzy C-Means, as shown in Figure 4.12o.



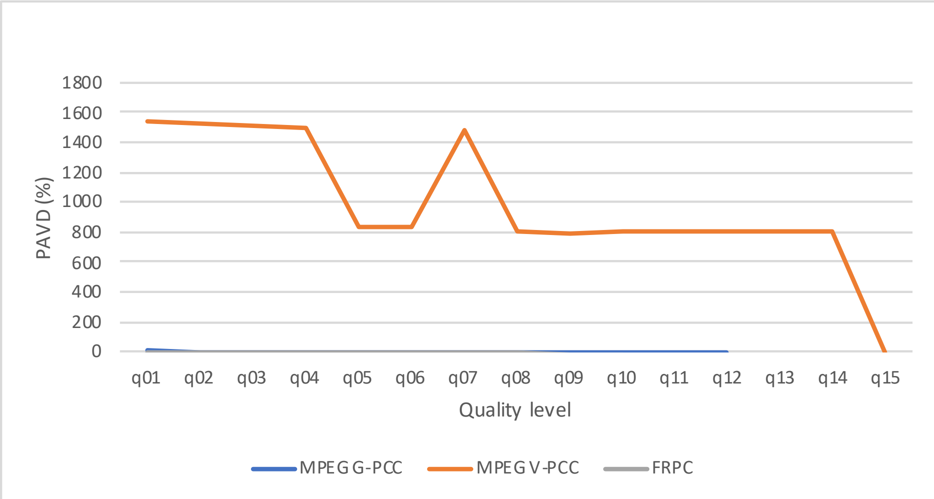
(a) Fuzzy C-Means - Experiment 1



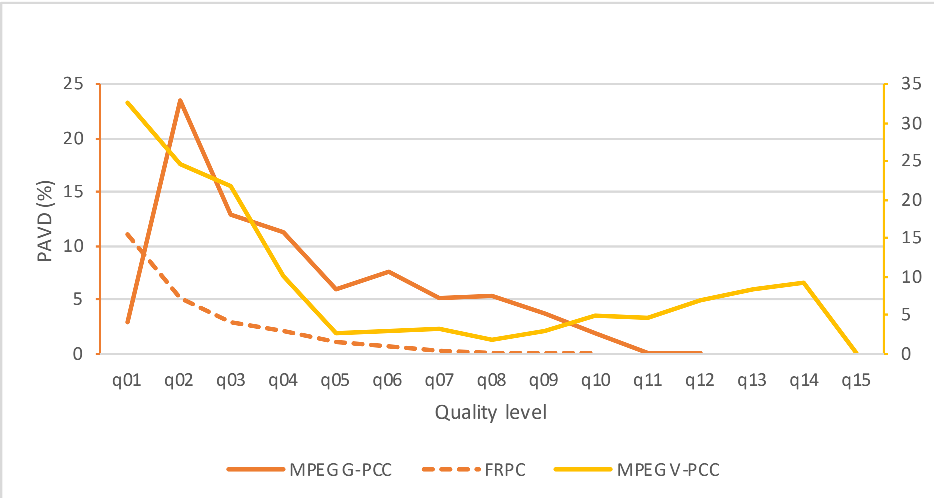
(b) Fuzzy C-Means - Experiment 2



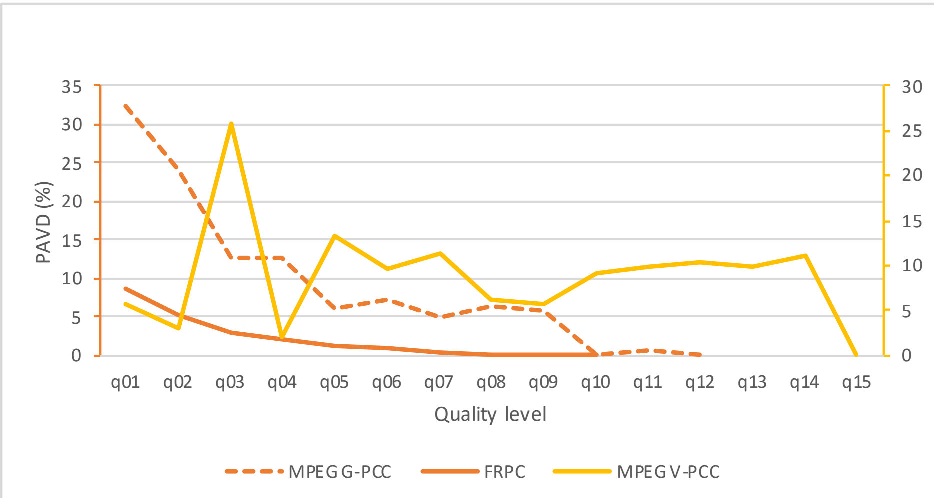
(c) Fuzzy C-Means - Experiment 3



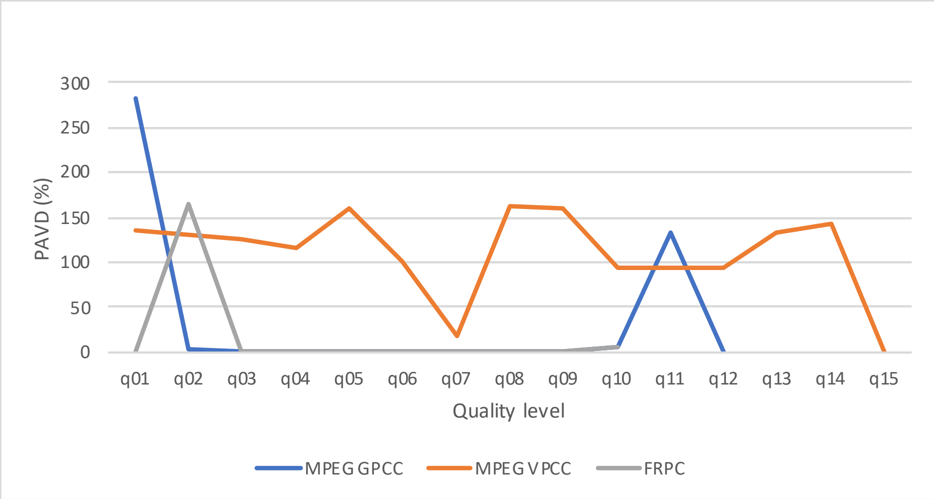
(d) Pcsegdist - Experiment 1



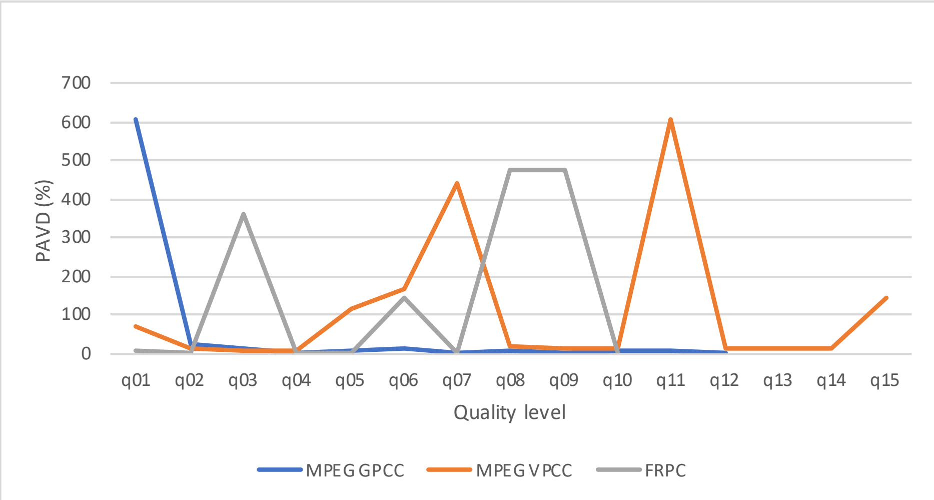
(e) Pcsegdist - Experiment 2



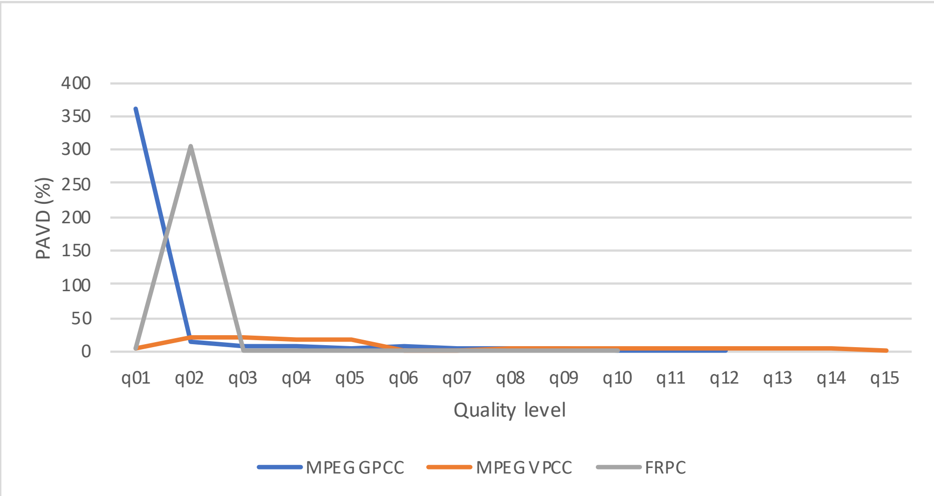
(f) Pcsegdist - Experiment 3



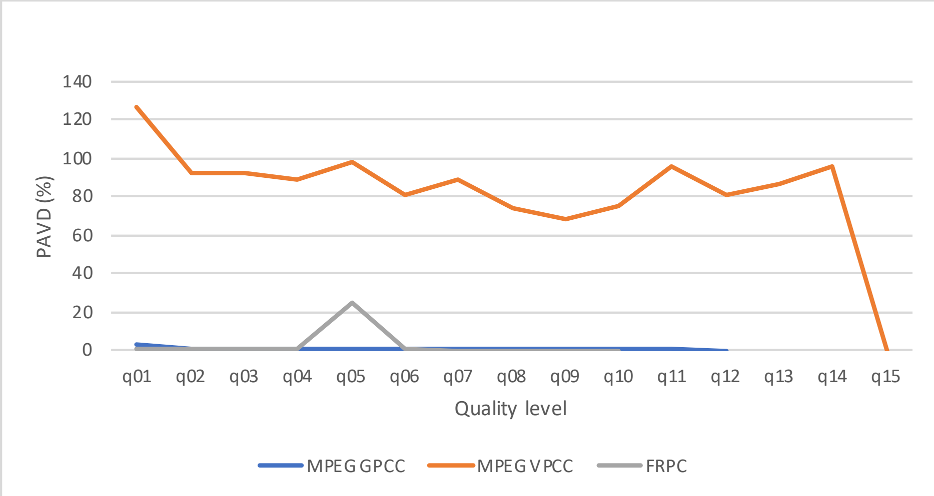
(g) K-Means - Experiment 1



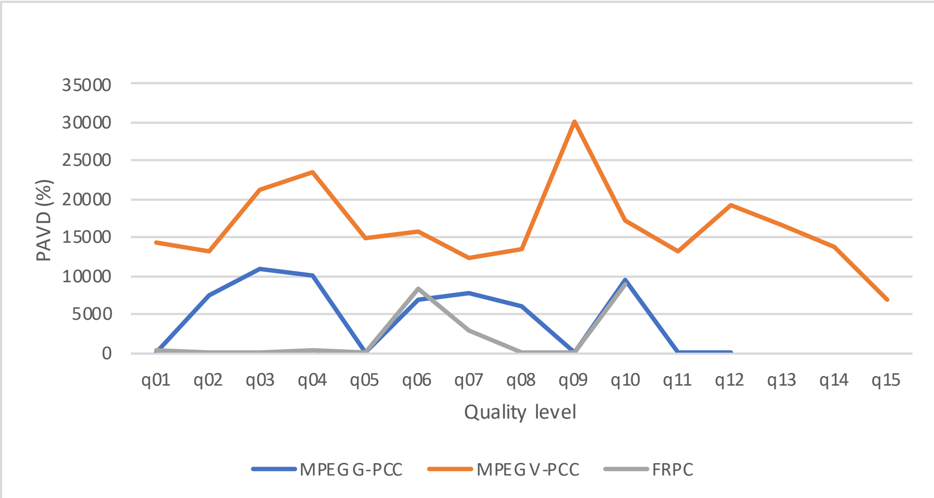
(h) K-Means - Experiment 2



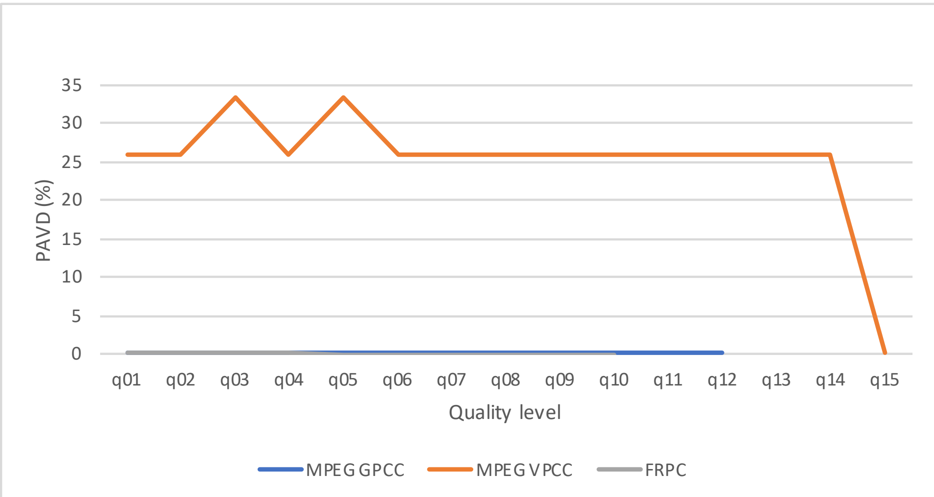
(i) K-Means - Experiment 3



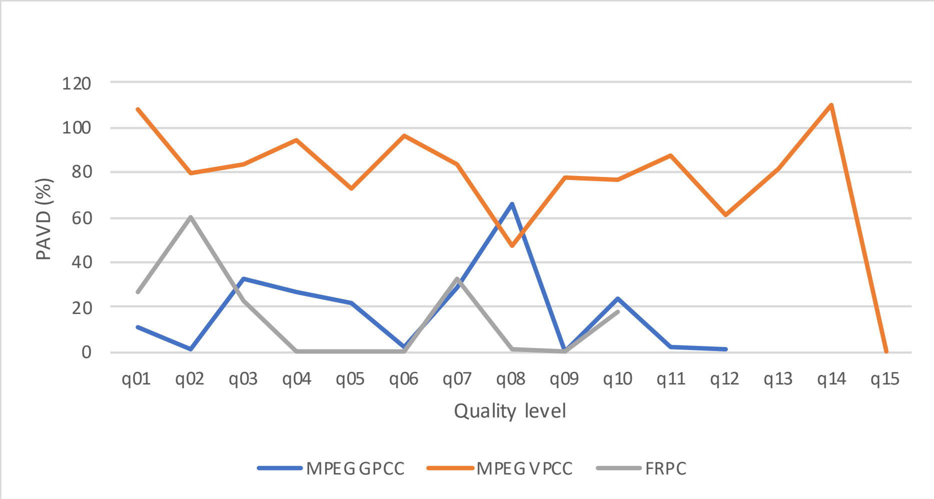
(j) Mean Shift - Experiment 1



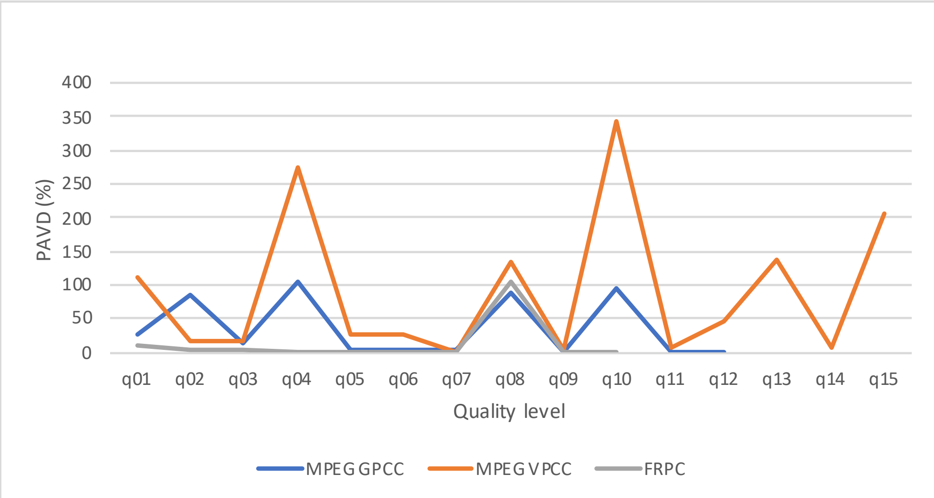
(k) Mean Shift - Experiment 2



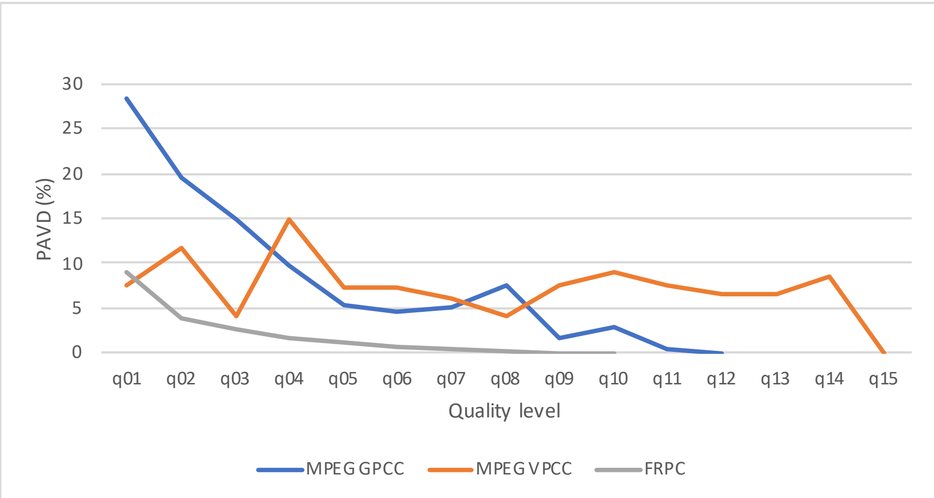
(l) Mean Shift - Experiment 3



(m) K-Medoids - Experiment 1



(n) K-Medoids - Experiment 2



(o) K-Medoids - Experiment 3

Figure 4.12: PAVD results.

The DCH illustrated in Figure 4.13 decreases with increasing quality level for FRPC and MPEG G-PCC, however for MPEG V-PCC it fluctuates. This happened for the rest of the cases which are represented in Figure B.1. This behavior is expected because as the quality level increases, the convex hull becomes more similar to the original one, so it makes sense that the distance between them is reduced along with the compression rate.

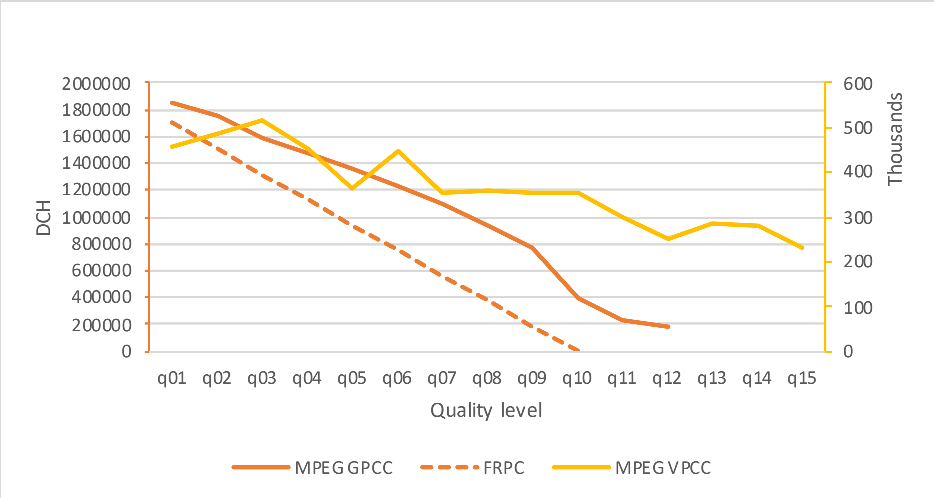
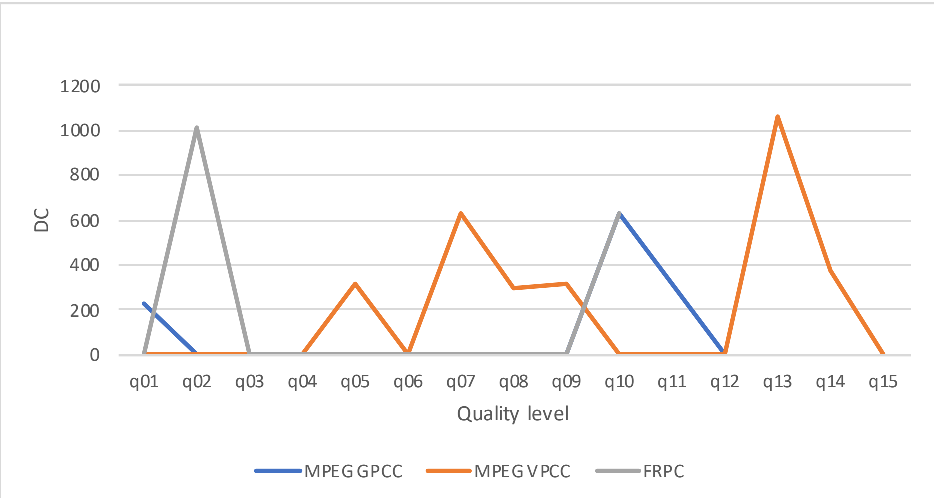
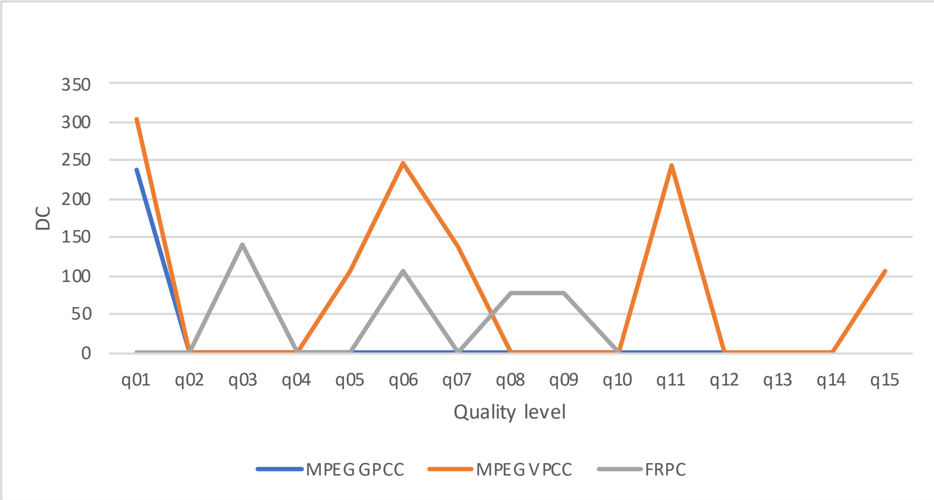


Figure 4.13: DCH result for experiment 1 using Fuzzy C-Means

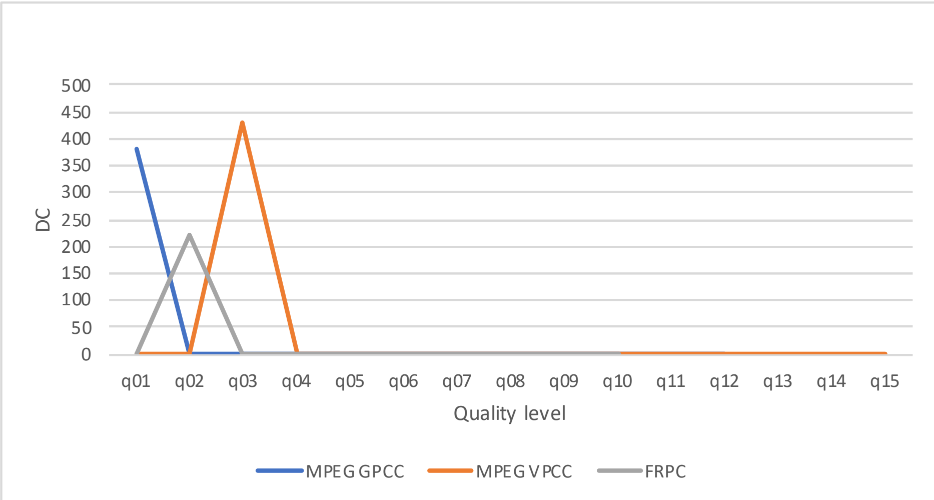
Concerning the DC, it was not found any consistent relationship with the quality level as illustrated in Figure 4.14 for the case of K-Means algorithm. Similar behavior was shown by Mean Shift, K-Medoids, Pcsegdist, Fuzzy C-Means and Subclust algorithms as illustrated in Figure B.2. This means that correctly finding the coordinates of each center is not directly related to the quality level of each PC.



(a) K-Means - Experiment 1



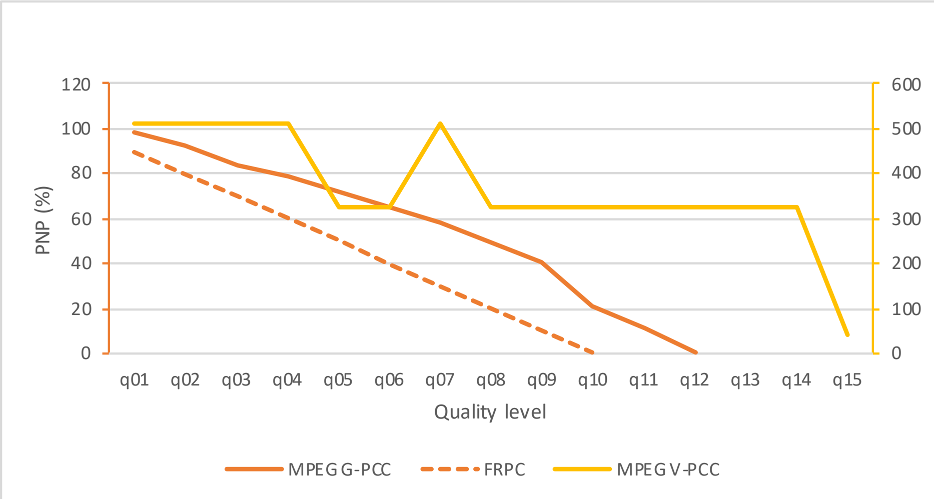
(b) K-Means - Experiment 2



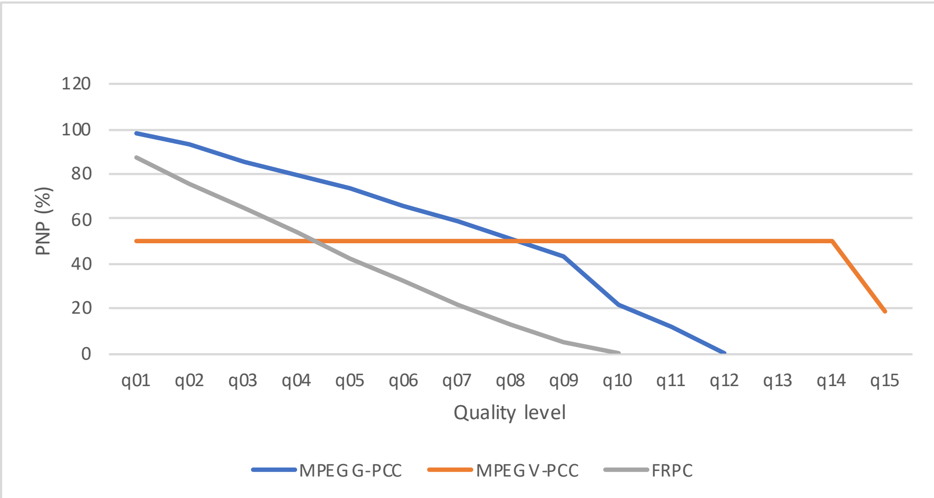
(c) K-Means - Experiment 3

Figure 4.14: DC results for K-Means algorithm.

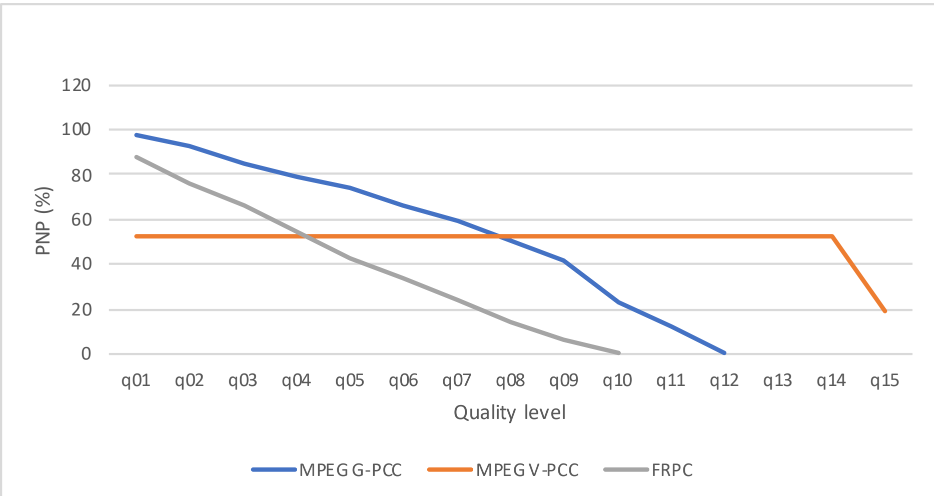
For Pcsegdist, the PNP decreases with increasing quality level for FRPC and MPEG G-PCC, for MPEG V-PCC it is constant or fluctuates as show in Figures 4.15. The same happened for K-Means, Fuzzy C-Means, K-Medoids and Mean Shift as illustrated in Figure B.3. This behavior is expected because as the quality level increases, it becomes more similar to the original one, so it makes sense that the difference of the number of points of each clusters is reduced along with the compression rate.



(a) Pcsegdist - Experiment 1



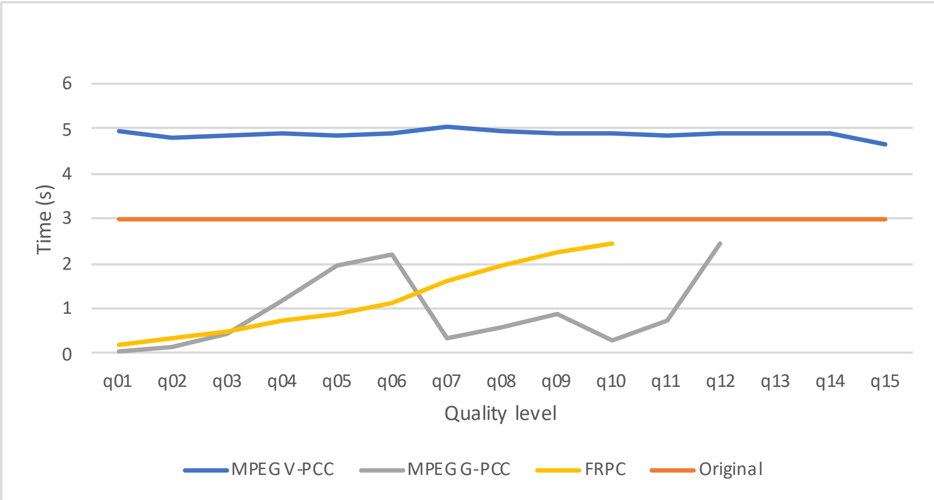
(b) Pcsegdist - Experiment 2



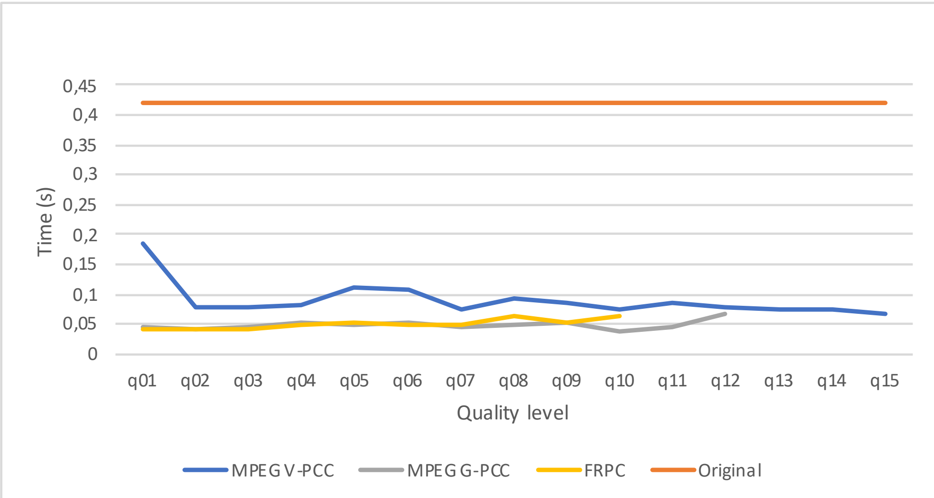
(c) Pcsegdist - Experiment 3

Figure 4.15: PNP results for Pcsegdist algorithm.

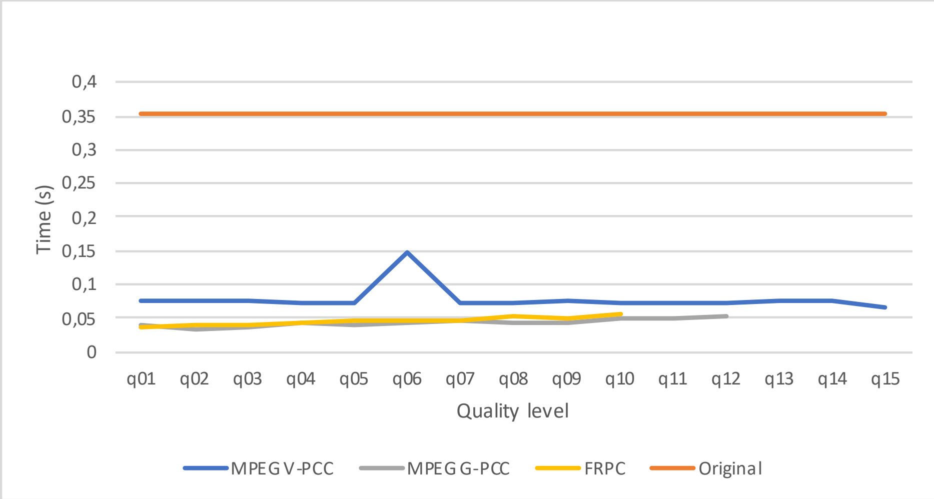
The running time for MPEG V-PCC half of the times is higher than the original and lower using the other methods as shown in Figure 4.16 for K-Medoids. For FRPC, in most cases the running times rises along the increasing compression ratio and MPEG G-PCC also shows some tendency to increase. The running times of the remaining algorithms is represented in Figure B.4. This can help to prove that compression does have some advantages such as running faster than an uncompressed PC for certain methods and running faster for lower compression ratios, also depending on the used methods.



(a) K-Medoids - Experiment 1



(b) K-Medoids - Experiment 2



(c) K-Medoids - Experiment 3

Figure 4.16: Representation of the running time according the quality level for K-Medoids algorithm

As it can be inferred, there is a pattern in most cases especially for the experiments which objects are apart from each other which means that this results are independent of the used dataset. What matters the most are the compression rates used which for good visual quality and better results, should use quality levels higher than q5.

4.2.2 LiDAR dataset

Tables 4.5, 4.6, 4.7, 4.8, and 4.9 represent the confusion matrices obtained after segmentation and classification performed on each used frame. Each original PC has three clusters and each cluster has a label (Pedestrian, Cyclist, Van). However, if no object is found where it was supposed to exist an object, it is classified as empty. To obtain these matrices, each one of the originals clusters' labels are compared to the ones from the compressed PC.

Table 4.5: Confusion matrix obtained after the classification performed on frame 000000

			Predicted object			
			Empty	**Cyclist**	**Pedestrian**	**Van**
Observed object	**11 bits**	**Empty**	0	0	0	0
		Cyclist	0	6	3	0
		Pedestrian	0	0	9	0
		Van	0	0	8	1
	12 bits	**Empty**	0	0	0	0
		Cyclist	0	7	0	2
		Pedestrian	0	0	9	0
		Van	0	0	8	1
	13 bits	**Empty**	0	0	0	0
		Cyclist	0	8	0	1
		Pedestrian	0	0	9	0
		Van	0	0	6	3
	14 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	0	9	0
		Van	0	0	2	7
	15 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	0	8	1
		Van	0	0	1	8
	20 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	0	7	2
		Van	0	0	0	9

Table 4.6: Confusion matrix obtained after the classification performed on frame 000001

			Predicted object			
			Empty	**Cyclist**	**Pedestrian**	**Van**
Observed object	**11 bits**	**Empty**	0	0	0	0
		Cyclist	0	7	0	2
		Pedestrian	1	0	8	0
		Van	0	0	9	0
	12 bits	**Empty**	0	0	0	0
		Cyclist	0	8	0	1
		Pedestrian	0	0	9	0
		Van	0	0	9	0
	13 bits	**Empty**	0	0	0	0
		Cyclist	0	7	0	2
		Pedestrian	0	0	9	0
		Van	0	0	5	4
	14 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	0	9	0
		Van	0	0	7	2
	15 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	0	8	1
		Van	0	0	8	1
	20 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	0	9	0
		Van	0	0	8	1

Table 4.7: Confusion matrix obtained after the classification performed on frame 000002

			Predicted object			
			Empty	**Cyclist**	**Pedestrian**	**Van**
Observed object	**11 bits**	**Empty**	0			
		Cyclist	0	7	0	2
		Pedestrian	0	0	9	0
		Van	0	0	7	2
	12 bits	**Empty**	0			
		Cyclist	0	7	0	2
		Pedestrian	0	0	9	0
		Van	0	1	4	4
	13 bits	**Empty**	0	0	0	0
		Cyclist	0	7	0	2
		Pedestrian	0	0	9	0
		Van	0	0	3	6
	14 bits	**Empty**	0	0	0	0
		Cyclist	0	8	0	1
		Pedestrian	0	0	9	0
		Van	0	0	1	8
	15 bits	**Empty**	0	0	0	0
		Cyclist	0	8	0	1
		Pedestrian	0	0	9	0
		Van	0	0	1	8
	20 bits	**Empty**	0	0	0	0
		Cyclist	0	4	0	5
		Pedestrian	0	0	9	0
		Van	0	0	0	9

Table 4.8: Confusion matrix obtained after the classification performed on frame 000003

			Predicted object			
			Empty	**Cyclist**	**Pedestrian**	**Van**
Observed object	**11 bits**	**Empty**	0	0	0	0
		Cyclist	0	7	1	1
		Pedestrian	1	0	8	0
		Van	0	0	7	2
	12 bits	**Empty**	0	0	0	0
		Cyclist	0	8	0	1
		Pedestrian	0	0	9	0
		Van	0	0	8	1
	13 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	0	9	0
		Van	0	0	2	7
	14 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	0	9	0
		Van	0	0	0	9
	15 bits	**Empty**	0	0	0	0
		Cyclist	0	8	0	1
		Pedestrian	0	0	9	0
		Van	0	0	0	9
	20 bits	**Empty**	0	0	0	0
		Cyclist	0	8	0	1
		Pedestrian	0	0	9	0
		Van	0	0	0	9

Table 4.9: Confusion matrix obtained after the classification performed on frame 000004

			Predicted object			
			Empty	**Cyclist**	**Pedestrian**	**Van**
Observed object	**11 bits**	**Empty**	0			
		Cyclist	0	5	1	3
		Pedestrian	0	0	9	0
		Van	0	0	8	1
	12 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	1	8	0
		Van	0	0	5	4
	13 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	1	8	0
		Van	0	0	1	8
	14 bits	**Empty**	0	0	0	0
		Cyclist	0	8	0	1
		Pedestrian	0	0	9	0
		Van	0	0	0	9
	15 bits	**Empty**	0	0	0	0
		Cyclist	0	7	0	2
		Pedestrian	0	3	6	0
		Van	0	0	0	9
	20 bits	**Empty**	0	0	0	0
		Cyclist	0	9	0	0
		Pedestrian	0	1	8	0
		Van	0	0	0	9

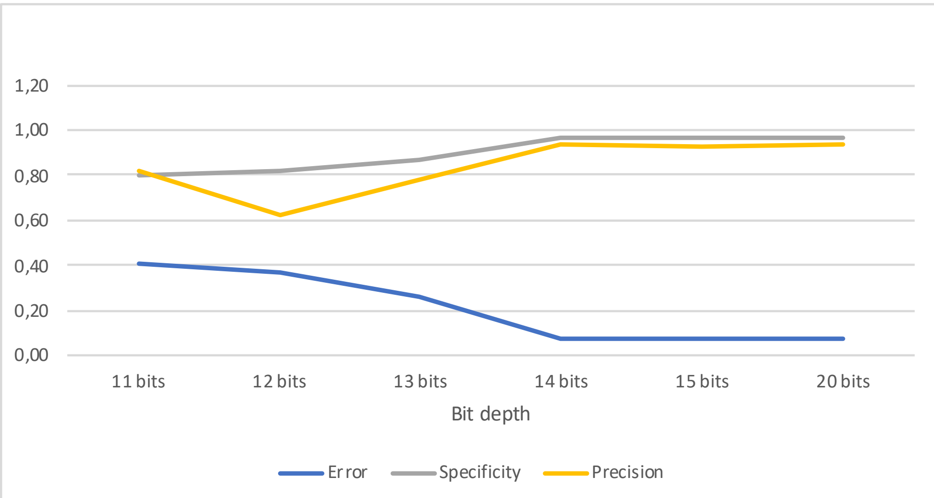
As it can be inferred from Tables 4.5,4.6,4.7,4.8, and 4.9, results are slightly better for bit depths above 13. This happens due to the information loss that was suffered during the voxelization process.

With 14 bit depth, in q1 and q3 from MPEG G-PCC was found a van instead of a pedestrian. For example, for a 15 bit depth, in q1 from FRPC was found a van instead of a pedestrian and in q1 from MPEG G-PCC was found a pedestrian instead of a van. Another example is in q1 and in q4 from FRPC from 20 bit depth was found a van instead of a pedestrian. In the second example, these mistakes were made in lower compression rates which can prove that compression does have an influence on processes such as segmentation and classification. However, in the first and last example, this theory does not apply which can be due to any geometric distortions introduced by the voxelization or the compression.

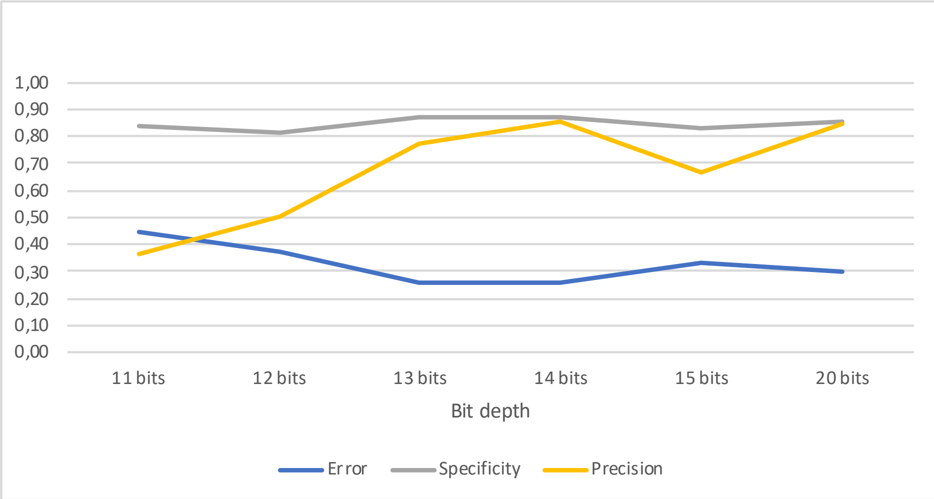
Table 4.10 presents three parameters related to the classification performed. According Figure 4.17, it is possible to conclude that the error rate decreases along with the increasing bit depth while specificity and precision increases. This behavior is expected because as the bit depth increases, there is more information in each PC, so there is supposed to have less errors in the classification process.

Table 4.10: Parameters from each confusion matrix

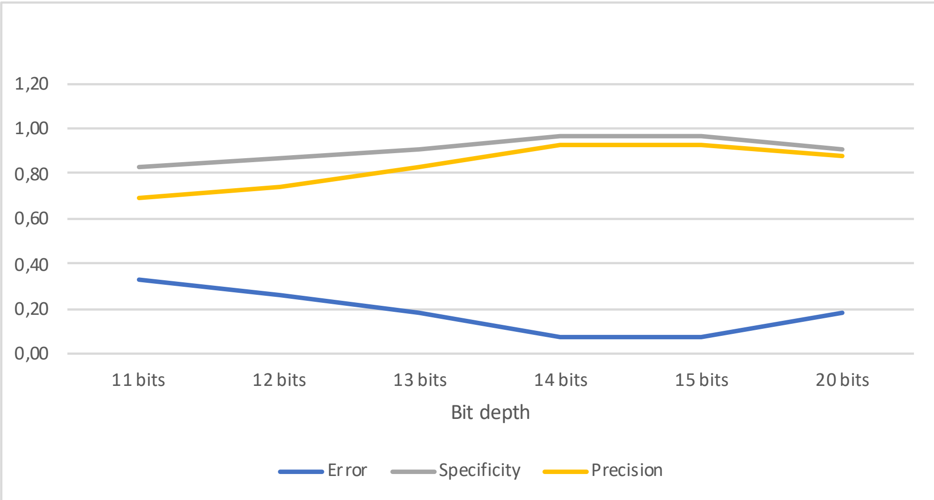
		Error	Precision	Specificity
Frame 000000	**11 bits**	0.41	0.82	0.80
	12 bits	0.37	0.62	0.81
	13 bits	0.26	0.78	0.87
	14 bits	0.07	0.94	0.96
	15 bits	0.07	0.93	0.96
	20 bits	0.07	0.94	0.96
Frame 000001	**11 bits**	0.44	0.37	0.84
	12 bits	0.37	0.50	0.81
	13 bits	0.26	0.77	0.87
	14 bits	0.26	0.85	0.87
	15 bits	0.33	0.67	0.83
	20 bits	0.30	0.84	0.85
Frame 000002	**11 bits**	0.33	0.69	0.83
	12 bits	0.26	0.74	0.87
	13 bits	0.19	0.83	0.91
	14 bits	0.07	0.93	0.96
	15 bits	0.07	0.93	0.96
	20 bits	0.19	0.88	0.91
Frame 000003	**11 bits**	0.37	0.54	0.87
	12 bits	0.33	0.68	0.83
	13 bits	0.07	0.94	0.96
	14 bits	0.00	1.00	1.00
	15 bits	0.04	0.97	0.98
	20 bits	0.04	0.97	0.98
Frame 000004	**11 bits**	0.44	0.58	0.78
	12 bits	0.22	0.84	0.89
	13 bits	0.07	0.93	0.96
	14 bits	0.04	0.97	0.98
	15 bits	0.19	0.84	0.91
	20 bits	0.04	0.97	0.98



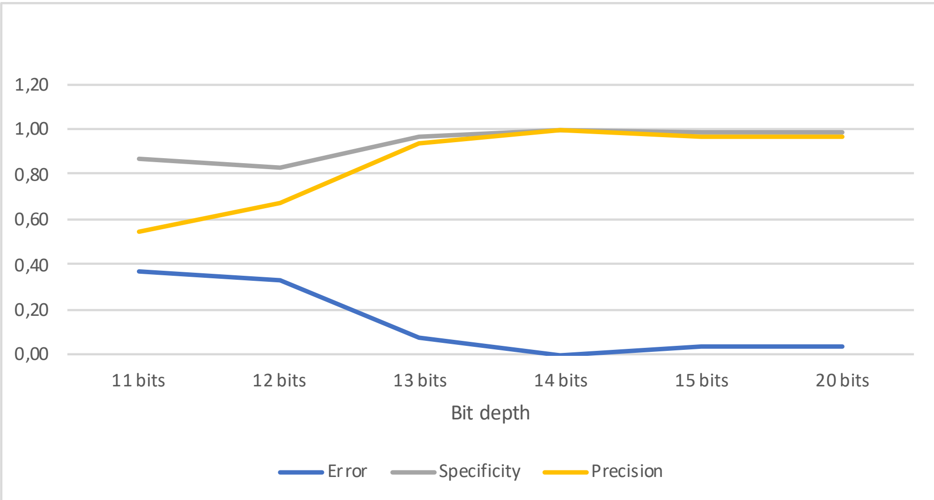
(a) Frame 000000



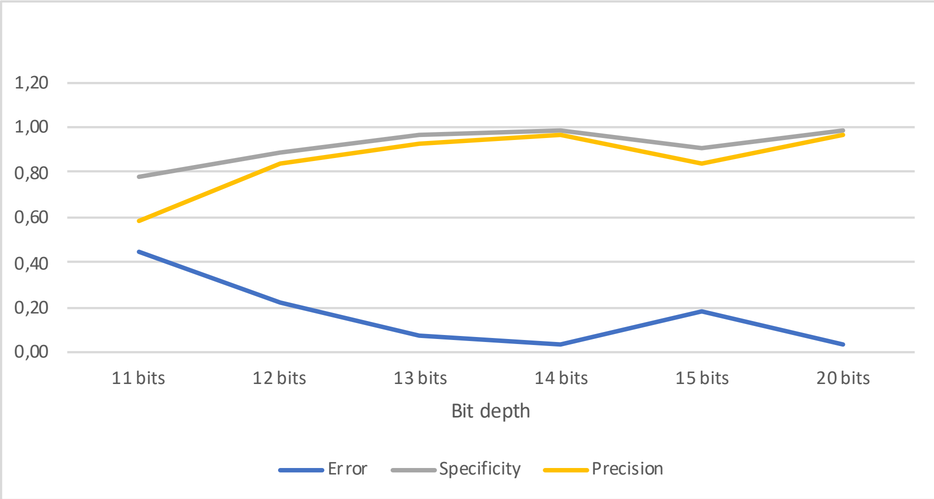
(b) Frame 000001



(c) Frame 000002



(d) Frame 000003



(e) Frame 000004

Figure 4.17: Representation of each confusion matrix parameters.

As it can be inferred, there is a pattern in most cases for these three parameters and also for the bit depths that achieved better results which is also an important matter. However, there is not a clear pattern related to the compression rates since there are misidentified objects in lower and higher quality levels.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This dissertation intended to research whether compression influences segmentation by clustering algorithms performance, uncovering performance trends and cutoff points after which the segmentation fails entirely. A secondary goal was to understand if some clustering algorithms are more robust to information loss due to PCC.

The first part of this work consisted in creating the general use dataset used by joining different objects in the same PC and in voxelizing the PCs from the LiDAR dataset. Then, they are compressed with different compression rates and encoder methods.

For the general use dataset, compression results show that, under certain conditions, bitrate increases along with the increasing compression ratio and the scaling ration is constant for MPEG V-PCC and it increases along with the quality level for the remaining methods. The remaining metrics behave according the PCC used.

For the lidar dataset, compression results show that in FRPC for higher bit depths, the overlap rises along with the increasing quality level and the same happens for MPEG G-PCC although for a 20 bit depth, the overlap remains constant which can mean that compression at lower rates was not very effective due to having too much information. For MPEG V-PCC, it was not possible to have conclusions because this method only allows PCs with a bit depth lower than 10.

The second part consists in applying several segmentation methods to the general use dataset and remove the ground, segment, and classify the LiDAR dataset.

For the general use dataset, it was possible to conclude that PAVD decreases with increasing compression rate for Fuzzy C-Means and Pcsegdist and has inconclusive behavior for the remaining methods. DC also has inconclusive behavior for all methods and DCH as well as PNP decrease along the rise of the quality level except for MPEG V-PCC which is usually constant or fluctuates. The running time is usually higher than the original when using MPEG V-PCC and lower when using the remaining methods which also show a tendency for increasing time along with the increasing compression rate. From these metrics, DC and PAVD are probably the ones not to use due to the lack of results with a pattern, although PAVD results had a pattern for Fuzzy C-Means and Pcsegdist. It is also possible to conclude that Fuzzy C-Means is the most robust algorithm and that all algorithms showed a cliff effect for MPEG G-PCC and FRPC but seemed insensitive to compression with MPEG V-PCC, for the performance measures that are conclusive. This means that MPEG G-PCC and FRPC are the obvious choices to compress PCs.

For LiDAR dataset, results show that the error rate decreases along with the increasing bit depth while specificity and precision increase. This behaviors were expected because with the bit depth increase, there is more information in the PC.

From compression and segmentation/classification results, it is possible to conclude, without surprise, that compression does influence the working of the tested algorithms and therefore their outputs. For the general use PC dataset, according the analyzed results, all compression rates show promising results although higher quality levels tend to have better results. However, if it necessary to have a good visual quality, it is a good idea to use compression rates higher than q5. For the LiDAR dataset, it was difficult to identify the compression rates than should not be used due to the diminished list of compression rate which can lead to uncertain patterns, but, most of the times, q1 should not be used as it has misidentified objects. However, as for voxelization, it was possible to find a pattern and a bit depth bellow 13 should not be used. It was not possible to choose one of the PCC as the one with best results because the two methods used with this dataset showed similar results.

5.2 Future work

Further work is needed to fully investigate the effects uncovered here. For example, perform similar tests but with different PCC methods and/or different segmentation methods. Moreover, the comparison between PCs may be improved by optimizing the process of comparison which sometimes fails due to different number of clusters or due to the misidentified clusters' centers. These experiments can also be performed using different datasets and new performance metrics.

Bibliography

[1] Khaled Mammou, Philip A. Chou, David Flyn, and Maja Krivokućan. PCC Test Model Category 13 v3. *document ISO/IEC JTC1/SC29/WG11 MPEG, N 17762*, July 2018.

[2] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko. Emerging MPEG Standards for Point Cloud Compression. volume 9, pages 133–148, March 2019.

[3] Siheng Chen, Dong Tian, Chen Feng, and Jelena Vetro, Anthonyand Kovacevic. Fast resampling of 3d point clouds via graphs. volume abs/1702.06397, 2017.

[4] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro. Geometric distortion metrics for point cloud compression. In *IEEE International Conference on Image Processing (ICIP)*, pages 3460–3464, Sep. 2017.

[5] Carsten Moenning and Neil A. Dodgson. A new point cloud simplification algorithm. 2003.

[6] E. S. Jang, M. Preda, K. Mammou, A. M. Tourapis, J. Kim, D. B. Graziosi, S. Rhyu, and M. Budagavi. Video-based point-cloud-compression standard in mpeg: From evidence collection to committee draft [standards in a nutshell]. volume 36, pages 118–123, May 2019.

[7] Stuart Perry, Luis Cruz, Manuela Pereira, Antonio Pinheiro, Emil Dumic, Guillaume Lavoué, Evangelos Alexiou, and Touradj Ebrahimi. Overview of Point Cloud Technology.

[8] Australian UAV Pty Ltd. Drone data vs lidar: Clarifying misconceptions. `https://www.auav.com.au/articles/drone-data-vs-lidar/`, 2017. Accessed: 2019-05-31.

[9] Martin Isenburg. Laszip: lossless compression of lidar data. volume 79, Feb 2013.

[10] Patrik Nygren and Michael Jasinski. A comparative study of segmentation and classification methods for 3d point clouds, 20016.

[11] M Manju, P Abarna, U Akila, and S Yamini. Peak signal to noise ratio & mean square error calculation for various images using the lossless image compression in ccsds algorithm. volume 119, pages 14471–14477, 01 2018.

[12] Stuart Perry. JPEG Pleno Point Clouds - Use Cases and Requirements Ver.2.3. *ISO/IEC JTC1/SC29/WG1 N80018*, September 2018.

[13] Ricardo L. de Queiroz and Philip A. Chou. Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform. volume 25, pages 3947–3956, Aug 2016.

[14] Mathworks. kmeans. `https://www.mathworks.com/help/stats/kmeans.html`, . Accessed: 2019-06-17.

[15] Mathworks. kmeans. `https://www.mathworks.com/help/stats/k-means-clustering-12.html`, . Accessed: 2019-07-3.

[16] S. Lloyd. Least squares quantization in pcm. volume 28, pages 129–137, March 1982.

[17] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *SODA*, pages 1027–1035. SIAM, 2007.

[18] Mathworks. fcm. `https://www.mathworks.com/help/fuzzy/fcm.html`, . Accessed: 2019-06-17.

[19] Mathworks. Fuzzy c-means clustering. `https://www.mathworks.com/help/fuzzy/fuzzy-c-means-clustering.html`, . Accessed: 2019-06-17.

[20] Mathworks. kmedoids. `https://www.mathworks.com/help/stats/kmedoids.html`, . Accessed: 2019-06-17.

[21] Bart Finkston. Mean shift. `https://www.mathworks.com/matlabcentral/fileexchange/10161-mean-shift-clustering`. Accessed: 2019-06-17.

[22] Mathworks. pcsegdist. `https://www.mathworks.com/help/vision/ref/pcsegdist.html`, . Accessed: 2019-06-17.

[23] Mathworks. subclust. `https://www.mathworks.com/help/fuzzy/subclust.html`, . Accessed: 2019-08-15.

[24] R. N. Mekuria, Z. Li, C. Tulvan, and P. Chao. Evaluation criteria for PCC (point cloud compression). *document ISO/IEC JTC1/SC29/WG11 MPEG2016/n16332*, 2016.

[25] Eugene dEon, Bob Harrison, Taos Myers, and Philip A. Chou. 8i Voxelized Full Bodies - A Voxelized Point Cloud Dataset. *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, January 2017.

[26] Stanford Computer Graphics Laboratory. The stanford 3d scanning repository. `http://graphics.stanford.edu/data/3Dscanrep/`. Accessed: 2019-08-12.

[27] Draft Call for Proposals for Point Cloud Compression. *document ISO/IEC JTC1/SC29/WG11 MPEG2014/N16538*, October 2016.

[28] E. Alexiou and T. Ebrahimi. On subjective and objective quality evaluation of point cloud geometry. In *Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3, May 2017.

[29] E. Alexiou, T. Ebrahimi, M. V. Bernardo, M. Pereira, A. Pinheiro, L. A. Da Silva Cruz, C. Duarte, L. G. Dmitrovic, E. Dumic, D. Matkovics, and A. Skodras. Point cloud subjective evaluation methodology based on 2d rendering. In *Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, May 2018.

[30] Andreas Geiger, P Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: the KITTI dataset. volume 32, pages 1231–1237, 09 2013.

[31] C. Premebida, J. Sousa, L. Garrote, and U. Nunes. Polar-grid representation and kriging-based 2.5d interpolation for urban environment modelling. In *IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1234–1239, September 2015.

[32] Basic evaluation measures from the confusion matrix. `https://classeval.wordpress.com/introduction/basic-evaluation-measures/`. Accessed: 2019-08-15.

Appendix A

Compression results

Table A.1: Bitrate and scaling ratio results of experiment 1.

	MPEG V-PCC		MPEG G-PCC		FRPC
	Bitrate (bits/point)	**Scaling ratio**	**Bitrate (bits/point)**	**Scaling ratio**	**Scaling ratio**
q01	0.01	1.50	0.03	0.02	0.10
q02	0.01	1.50	0.10	0.07	0.20
q03	0.01	1.50	0.22	0.16	0.30
q04	0.02	1.50	0.30	0.21	0.40
q05	0.03	1.50	0.38	0.28	0.50
q06	0.04	1.50	0.49	0.30	0.60
q07	0.05	1.50	0.60	0.42	0.70
q08	0.07	1.50	0.72	0.51	0.80
q09	0.11	1.50	0.85	0.59	0.90
q10	0.16	1.50	1.12	0.79	-
q11	0.26	1.50	1.27	0.89	-
q12	0.40	1.50	1.42	1.00	-
q13	0.65	1.50	-	-	-
q14	1.31	1.50	-	-	-
q15	1.80	1.40	-	-	-

Table A.2: Bitrate and scaling ratio results of experiment 2.

	MPEG V-PCC		MPEG G-PCC		FRPC
	Bitrate (bits/point)	**Scaling ratio**	**Bitrate (bits/point)**	**Scaling ratio**	**Scaling ratio**
q01	0.11	1.76	0.05	0.02	0.10
q02	0.14	1.76	0.13	0.07	0.20
q03	0.17	1.76	0.40	0.26	0.30
q04	0.22	1.76	0.76	0.57	0.40
q05	0.32	1.76	1.05	0.78	0.50
q06	1.17	1.76	1.17	0.89	0.60
q07	0.76	1.76	0.30	0.20	0.70
q08	0.49	1.76	0.48	0.30	0.80
q09	0.10	1.76	0.69	0.49	0.90
q10	0.10	1.76	0.24	0.15	-
q11	0.09	1.76	0.58	0.40	-
q12	0.09	1.76	1.28	1.00	-
q13	0.09	1.76	-	-	-
q14	1.76	1.76	-	-	-
q15	2.15	1.47	-	-	-

Table A.3: Bitrate and scaling ratio results of experiment 3.

	MPEG V-PCC		MPEG G-PCC		FRPC
	Bitrate (bits/point)	**Scaling ratio**	**Bitrate (bits/point)**	**Scaling ratio**	**Scaling ratio**
q01	0.09	1.78377	0.05	0.0166489	0.0999753
q02	0.09	1.78377	0.12	0.0665954	0.199992
q03	0.09	1.78377	0.23	0.146962	0.299967
q04	0.09	1.78377	0.29	0.196539	0.399984
q05	0.10	1.78377	0.40	0.258489	0.5
q06	0.11	1.78377	0.49	0.300223	0.599975
q07	0.13	1.78377	0.58	0.40023	0.699992
q08	0.17	1.78377	0.69	0.486969	0.799967
q09	0.22	1.78377	0.76	0.569144	0.899984
q10	0.32	1.78377	1.01	0.774973	-
q11	0.49	1.78377	1.12	0.881608	-
q12	0.74	1.78377	1.27	1.00	-
q13	1.11	1.78377	-	-	-
q14	1.76	1.78377	-	-	-
q15	2.18	1.47217	-	-	-

Table A.4: Results of MPEG V-PCC of experiment 1

	P2Point		P2Plane		P2Point		P2Plane	
	MSE	**PSNR based on MSE (dB)**	**MSE**	**PSNR based on MSE (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**
q01	33.09	-2.64	30.89	-2.30	48226	-34.28	44246	-33.91
q02	28.50	-1.99	26.53	-1.68	47195	-34.19	44699	-33.95
q03	27.31	-1.81	25.42	-1.50	48349	-34.29	46655	-34.14
q04	24.30	-1.30	22.62	-0.99	48208	-34.28	44740	-33.95
q05	23.50	-1.16	21.96	-0.86	48466	-34.30	45135	-33.99
q06	22.10	-0.89	20.66	-0.60	47963	-34.26	45716	-34.05
q07	21.59	-0.79	20.23	-0.51	48123	-34.27	45095	-33.99
q08	22.29	-0.93	20.95	-0.66	48113	-34.27	45453	-34.02
q09	21.76	-0.82	20.50	-0.56	47096	-34.18	44467	-33.93
q10	21.84	-0.84	20.57	-0.58	46393	-34.11	44710	-33.95
q11	22.20	-0.91	20.90	-0.65	48077	-34.27	45361	-34.01
q12	22.21	-0.91	20.94	-0.66	48077	-34.27	46124	-34.09
q13	22.20	-0.91	20.91	-0.65	48435	-34.30	45742	-34.05
q14	22.21	-0.91	20.95	-0.66	46820	-34.15	45088	-33.99
q15	0.00	inf	0.00	inf	0	inf	0	inf

Table A.5: Results of MPEG G-PCC of experiment 1

	P2Point		P2Plane		P2Point		P2Plane	
	MSE	**PSNR based on MSE (dB)**	**MSE**	**PSNR based on MSE (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**
q01	16.47	0.38	8.55	3.23	48.00	-4.26	47.87	-4.25
q02	4.50	6.02	1.87	9.82	12.00	1.76	12.00	1.76
q03	1.83	9.92	0.68	14.23	5.30	5.28	5.30	5.28
q04	1.32	11.36	0.46	15.92	3.92	6.62	3.92	6.62
q05	1.50	10.79	0.45	16.02	3.00	7.78	3.00	7.78
q06	0.80	13.54	0.26	18.46	2.37	8.80	2.37	8.80
q07	0.66	14.36	0.21	19.30	1.92	9.72	1.92	9.72
q08	0.53	15.28	0.17	20.17	1.59	10.55	1.59	10.55
q09	0.50	15.56	0.16	20.58	1.30	11.30	1.30	11.30
q10	0.34	17.29	0.11	22.17	0.98	12.64	0.98	12.64
q11	0.29	17.96	0.10	22.72	0.85	13.24	0.85	13.24
q12	0.00	inf	0.00	inf	0.00	inf	0.00	inf

Table A.6: Results of FRPC of experiment 1

	P2Point		P2Plane		P2Point		P2Plane	
	MSE	**PSNR based on MSE (dB)**	**MSE**	**PSNR based on MSE (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**
q01	2.84	8.01	0.23	18.86	86	-6.79	22.22	-0.92
q02	1.47	10.88	0.19	19.83	44	-3.88	15.48	0.66
q03	0.99	12.59	0.15	20.70	29	-2.07	11.88	1.80
q04	0.73	13.90	0.12	21.64	20	-0.46	10.24	2.45
q05	0.56	15.08	0.10	22.71	20	-0.46	10.24	2.45
q06	0.42	16.28	0.07	24.00	10	2.55	5.85	4.88
q07	0.31	17.67	0.05	25.62	10	2.55	5.85	4.88
q08	0.20	19.50	0.03	27.85	10	2.55	5.85	4.88
q09	0.10	22.54	0.01	31.62	6	4.77	2.90	7.93
q10	0.00	inf	0.00	inf	0.00	inf	0.00	inf

Table A.7: Results of MPEG V-PCC of experiment 2

	P2Point		P2Plane		P2Point		P2Plane	
	MSE	**PSNR based on MSE (dB)**	**MSE**	**PSNR based on MSE (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**
q01	0.949903	8.0047	0.4931	10.852	21	-5.441	16.772	-4.464
q02	0.596592	10.025	0.3091	12.23	22	-5.643	17.162	-4.564
q03	0.456115	11.191	0.2839	13.25	24	-6.021	24	-6.021
q04	0.380162	11.982	0.2273	14.215	19	-5.006	16.772	-4.464
q05	0.345572	12.396	0.203	14.707	21	-5.441	20.152	-5.262
q06	0.277583	13.348	0.1615	15.7	21	-5.441	20.152	-5.262
q07	0.296828	13.057	0.1711	15.448	21	-5.441	20.152	-5.262
q08	0.311893	12.842	0.1788	15.257	21	-5.441	20.152	-5.262
q09	1.88486	5.0287	1.0971	7.3791	65	-10.30	58.117	-9.862
q10	2.56844	3.6848	1.278	6.7161	65	-10.30	58.117	-9.862
q11	4.40028	1.3467	2.233	4.2925	100	-12.22	90.808	-11.8
q12	4.08904	1.6653	2.3089	4.0914	85	-11.51	75.98	-11.03
q13	6.70736	-0.484	4.1439	1.6074	121	-13.05	118.52	-12.96
q14	0.26719	13.513	0.158	15.796	21	-5.441	20.152	-5.262
q15	0.00	inf	0.00	inf	0	inf	0	inf

Table A.8: Results of MPEG G-PCC of experiment 2

	P2Point		P2Plane		P2Point		P2Plane	
	MSE	**PSNR based on MSE (dB)**	**MSE**	**PSNR based on MSE (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**
q01	15.341	-4.07706	6.51663	-0.308718	48	-9.0309	46.7145	-8.913
q02	4.675	1.08369	1.84561	5.12011	12	-3.0103	11.6694	-2.889
q03	1.5569	5.85888	0.491231	10.8687	3	3.0103	2.99341	3.0199
q04	0.5194	10.6261	0.171261	15.4449	1.30341	6.53186	1.30915	6.6116
q05	0.3131	12.8249	0.0868314	18.3947	0.979637	7.87086	0.953299	7.9892
q06	0.2901	13.1561	0.0982125	17.8598	0.853307	8.47046	0.748182	9.0414
q07	1.298	6.64872	0.413574	11.616	3.91848	1.85034	3.43099	2.4273
q08	0.7852	8.8316	0.240134	13.977	2.37052	4.03308	2.04249	4.6799
q09	0.5372	10.4798	0.175212	15.3459	1.58687	5.77609	1.43501	6.213
q10	1.8701	5.06286	0.696284	9.30365	5.30341	0.511459	5.19034	0.6296
q11	0.6732	9.49987	0.22304	14.2977	1.91994	4.94863	1.86852	5.0665
q12	0.00	inf	0.00	inf	0.00	inf	0.00	inf

Table A.9: Results of FRPC of experiment 2

	P2Point		P2Plane		P2Point		P2Plane	
	MSE	**PSNR based on MSE (dB)**	**MSE**	**PSNR based on MSE (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**
q01	3.099	2.86924	0.2289	14.1856	41	-8.346	4.8801	0.897217
q02	1.5959	5.75137	0.1618	15.6922	20	-5.229	3.9449	1.8212
q03	1.0569	7.54139	0.1203	16.9792	12	-3.01	3.1951	2.73672
q04	0.7711	8.91048	0.0915	18.166	10	-2.218	3.274	2.63073
q05	0.5786	10.158	0.0675	19.4886	9	-1.761	1.7842	5.26707
q06	0.4322	11.4244	0.047	21.0593	8	-1.249	1.3066	6.52154
q07	0.3096	12.8736	0.0291	23.1429	4	1.7609	1.3011	6.63831
q08	0.202	14.7273	0.0161	25.7036	4	1.7609	0.9986	7.78772
q09	0.1001	17.7767	0.0057	30.2403	2	4.7712	0.9771	7.88213
q10	0.00	inf	0.00	inf	0.00	inf	0.00	inf

Table A.10: Results of MPEG V-PCC of experiment 3

	P2Point		P2Plane		P2Point		P2Plane	
	MSE	**PSNR based on MSE (dB)**	**MSE**	**PSNR based on MSE (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**
q01	3.15	-0.21	1.74	2.36	131.00	-16.40	121.40	-16.07
q02	2.78	0.30	1.53	2.94	146.00	-16.87	145.61	-16.86
q03	3.24	-0.30	1.73	2.39	100.00	-15.23	52.27	-12.41
q04	2.12	1.51	1.14	4.19	44.00	-11.66	37.28	-10.94
q05	1.85	2.11	1.13	4.26	25.00	-9.21	18.44	-7.89
q06	0.90	5.23	0.47	8.01	20.00	-8.24	16.00	-7.27
q07	0.56	7.29	0.30	9.27	21.00	-8.45	20.59	-8.37
q08	0.45	8.22	0.27	10.44	18.00	-7.78	16.43	-7.38
q09	0.41	8.69	0.24	10.95	18.00	-7.78	12.00	-6.02
q10	0.37	9.07	0.22	11.39	18.00	-7.78	12.00	-6.02
q11	0.34	9.42	0.20	11.87	18.00	-7.78	12.00	-6.02
q12	0.32	9.69	0.18	12.12	18.00	-7.78	12.00	-6.02
q13	0.31	9.92	0.18	12.31	18.00	-7.78	12.00	-6.02
q14	0.29	10.14	0.17	12.47	18.00	-7.78	12.00	-6.02
q15	0.00	inf	0.00	inf	0.00	inf	0.00	inf

Table A.11: Results of MPEG G-PCC of experiment 3

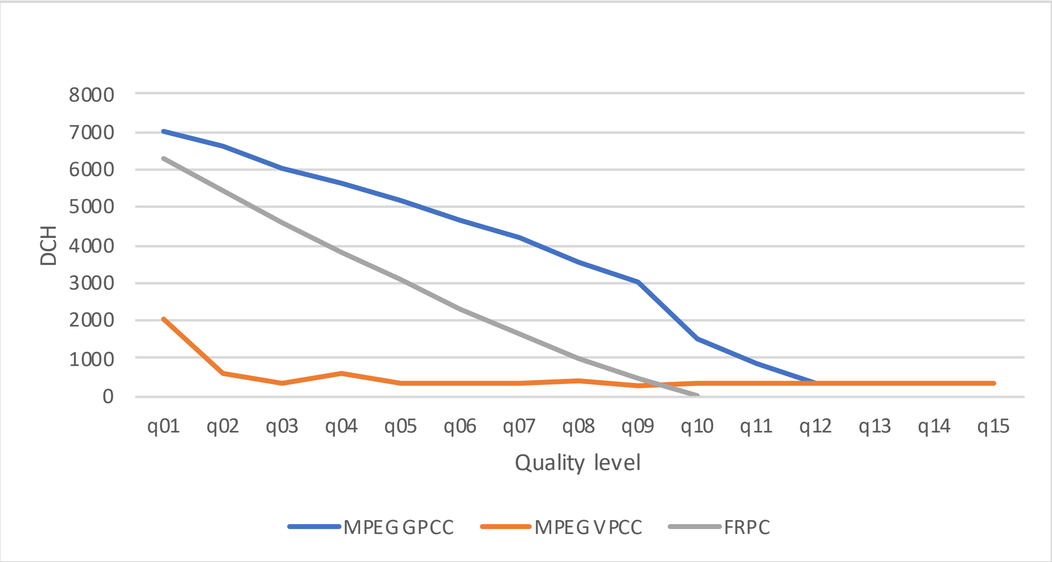
	P2Point		P2Plane		P2Point		P2Plane	
	MSE	**PSNR based on MSE (dB)**	**MSE**	**PSNR based on MSE (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**
q01	15.373	-7.09625	6.43349	-3.31325	48	-12.0412	46.3926	-11.89
q02	4.644	-1.89767	1.82109	2.1679	12	-6.0206	11.7782	-5.94
q03	1.8767	2.03728	0.690197	6.38148	5.30341	-2.49884	4.47053	-1.732
q04	1.3772	3.38116	0.474993	8.00434	3.91848	-1.15996	3.77658	-1
q05	1.5592	2.84221	0.481753	7.94297	3	0	3	######
q06	0.8331	5.56405	0.290416	10.141	2.37046	1.02288	2.0258	1.7052
q07	0.6756	6.47429	0.224286	11.2632	1.91994	1.93833	1.79226	2.2372
q08	0.5484	7.3803	0.183707	12.1299	1.58683	2.76591	1.30333	3.5218
q09	0.516	7.64476	0.169385	12.4825	1.30341	3.52156	1.31768	3.5731
q10	0.3137	9.80572	0.087152	15.3684	0.979637	4.86056	0.914411	5.1598
q11	0.2781	10.3291	0.0760619	15.9595	0.853307	5.46016	0.779832	5.8512
q12	0.00	inf	0.00	inf	0.00	inf	0.00	inf

Table A.12: Results of FRPC of experiment 3

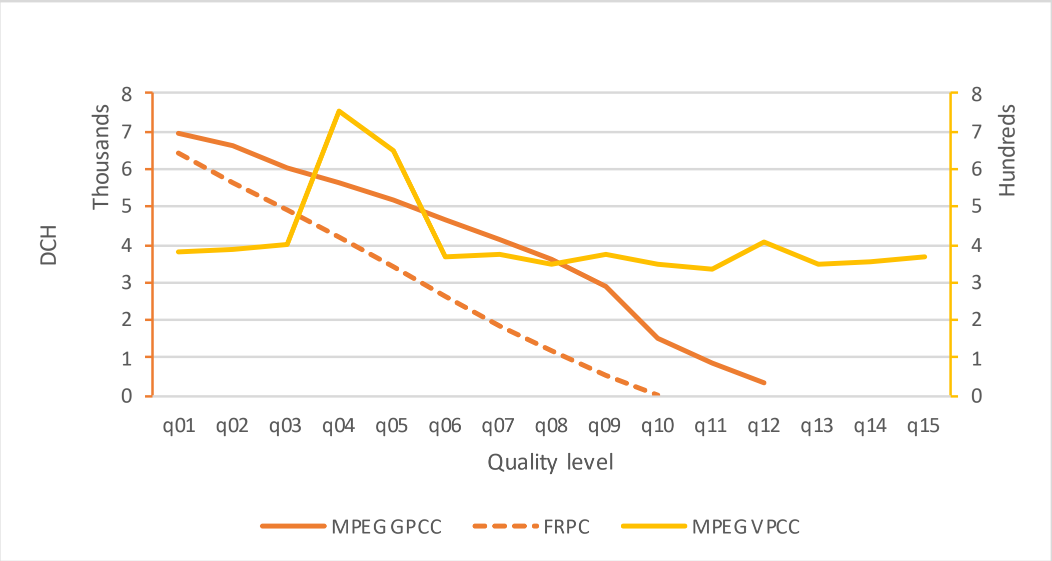
	P2Point		P2Plane		P2Point		P2Plane	
	MSE	**PSNR based on MSE (dB)**	**MSE**	**PSNR based on MSE (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**	**Hausdorff**	**PSNR based on Hausdorff (dB)**
q01	2.9634	0.05335	0.2243	11.2627	33	-10.41	8.0542	-4.28903
q02	1.5523	2.8616	0.1576	12.7951	17	-7.533	2.8742	0.18603
q03	1.0421	4.59231	0.1203	13.9679	13	-6.368	2.307	1.14069
q04	0.7608	5.95856	0.0931	15.0829	10	-5.229	1.7762	2.2764
q05	0.5737	7.18462	0.0712	16.2455	8	-4.26	1.7688	2.29435
q06	0.4309	8.42748	0.0512	17.678	5	-2.218	1.623	2.66799
q07	0.3095	9.86455	0.0354	19.2836	5	-2.218	1.2211	3.90374
q08	0.2025	11.7079	0.0204	21.6643	4	-1.249	1	4.77122
q09	0.1001	14.7651	0.0079	25.7989	2	1.7609	0.9546	4.973
q10	0.00	inf	0.00	inf	0.00	inf	0.00	inf

Appendix B

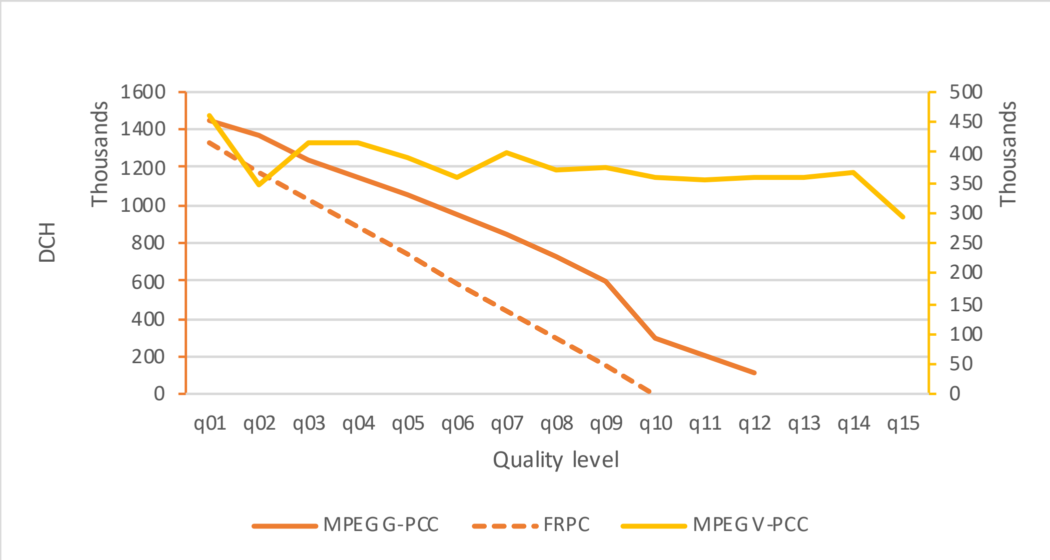
Segmentation results



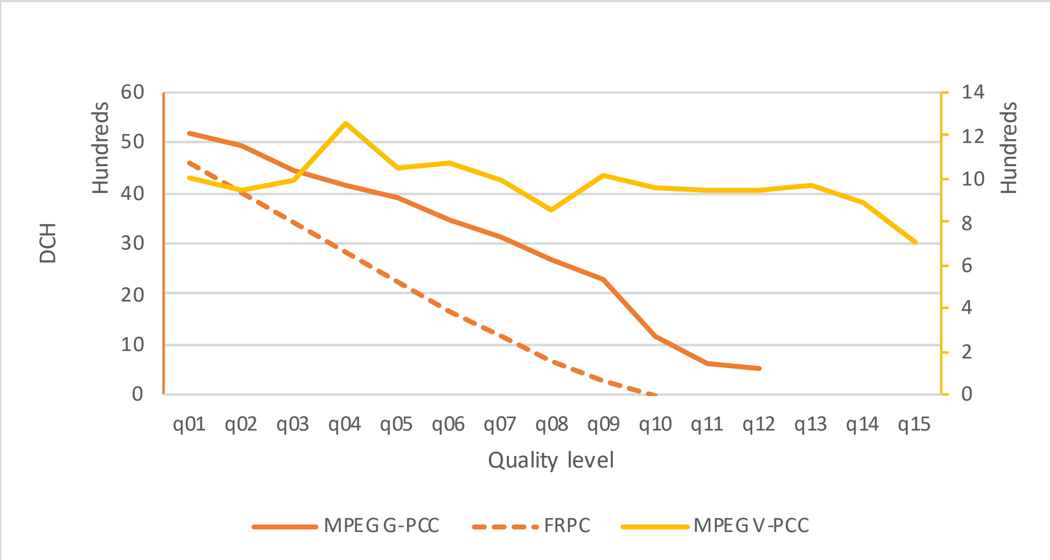
(a) Fuzzy C-Means - Experiment 2



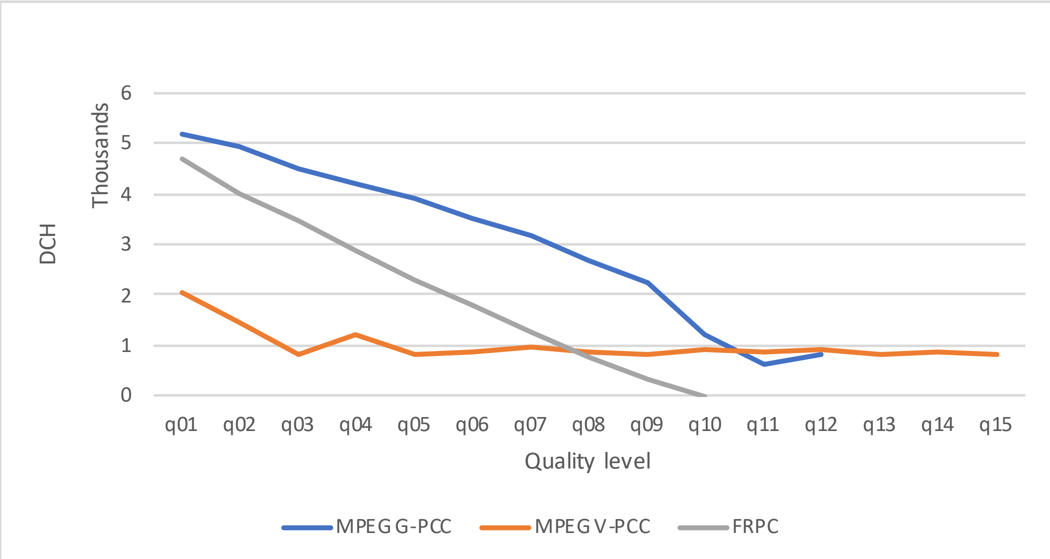
(b) Fuzzy C-Means - Experiment 3



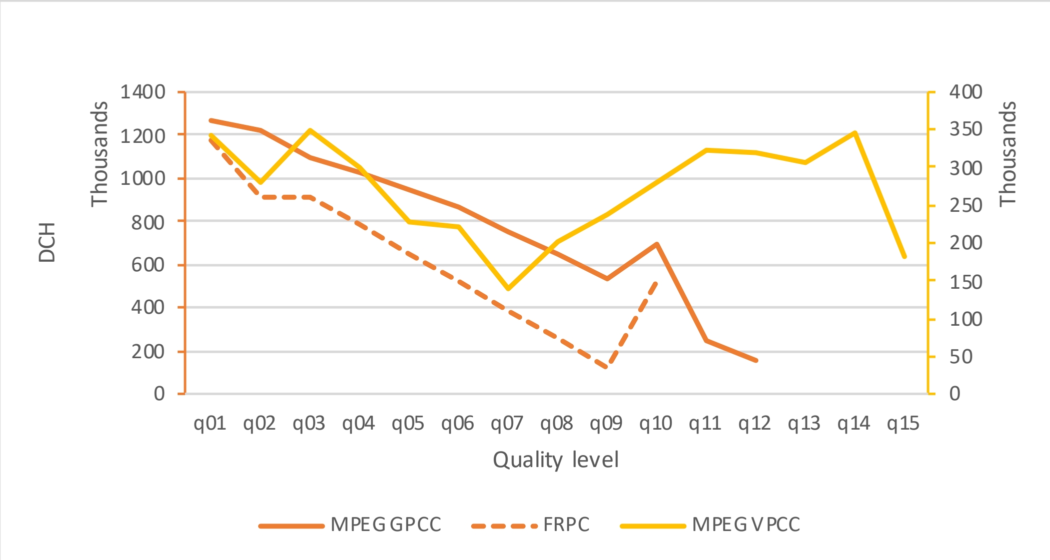
(c) Pcsegdist - Experiment 1



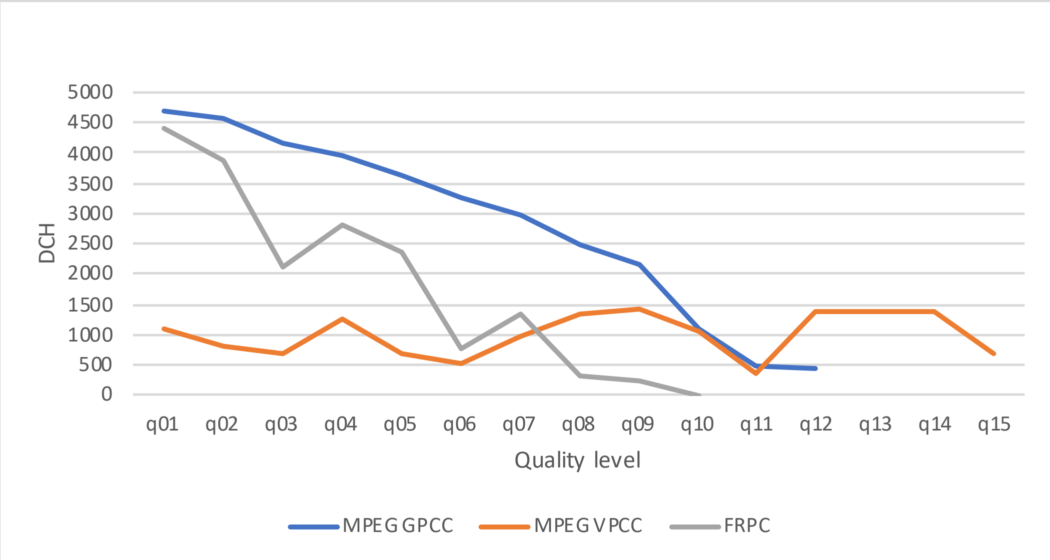
(d) Pcsegdist - Experiment 2



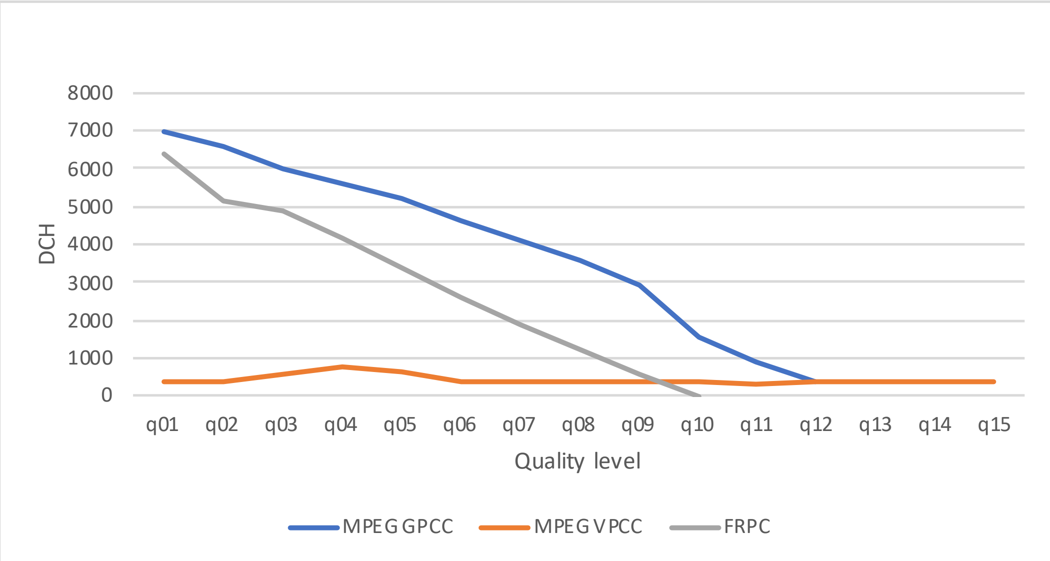
(e) Pcsegdist - Experiment 3



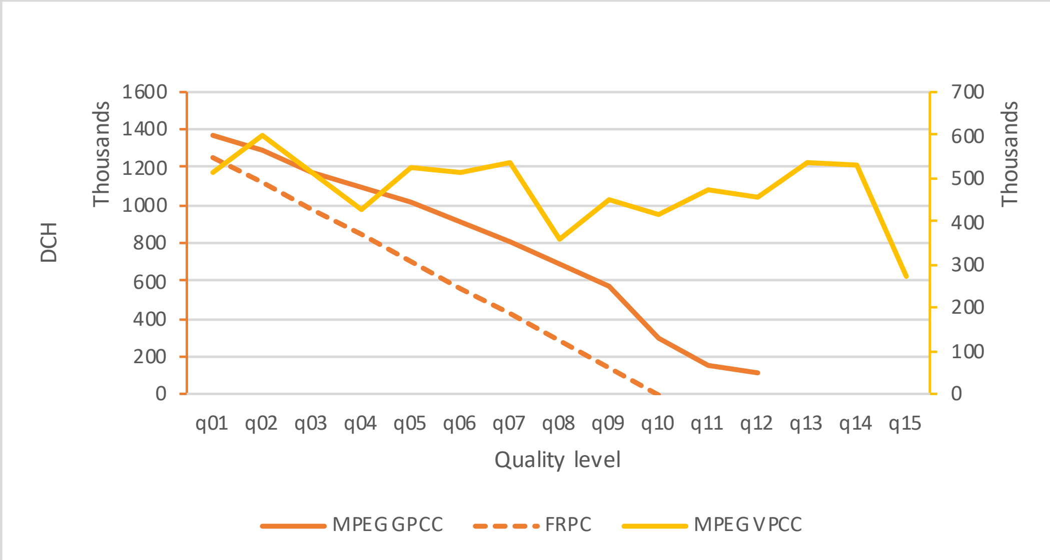
(f) K-Means - Experiment 1



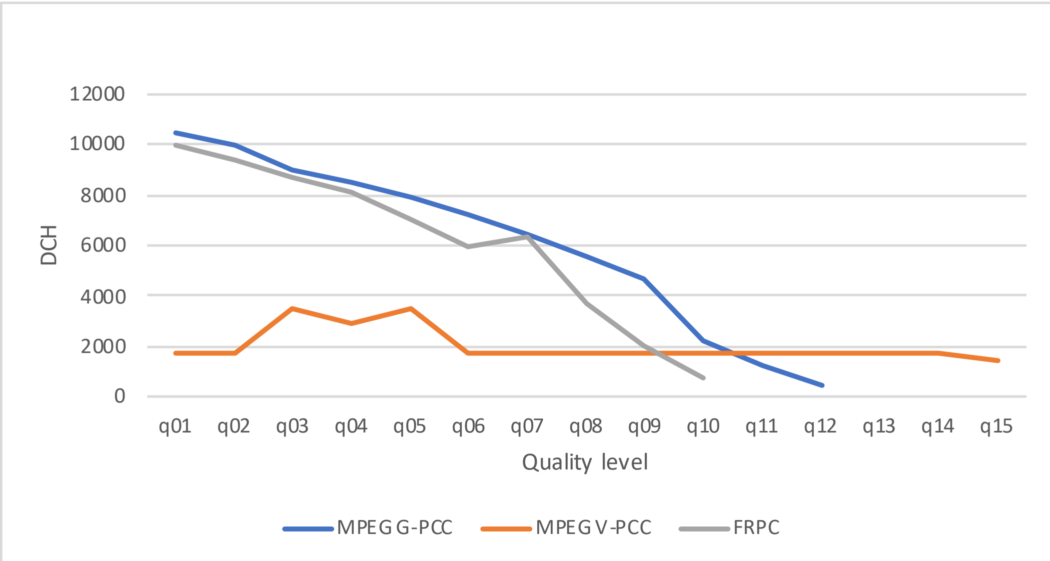
(g) K-Means - Experiment 2



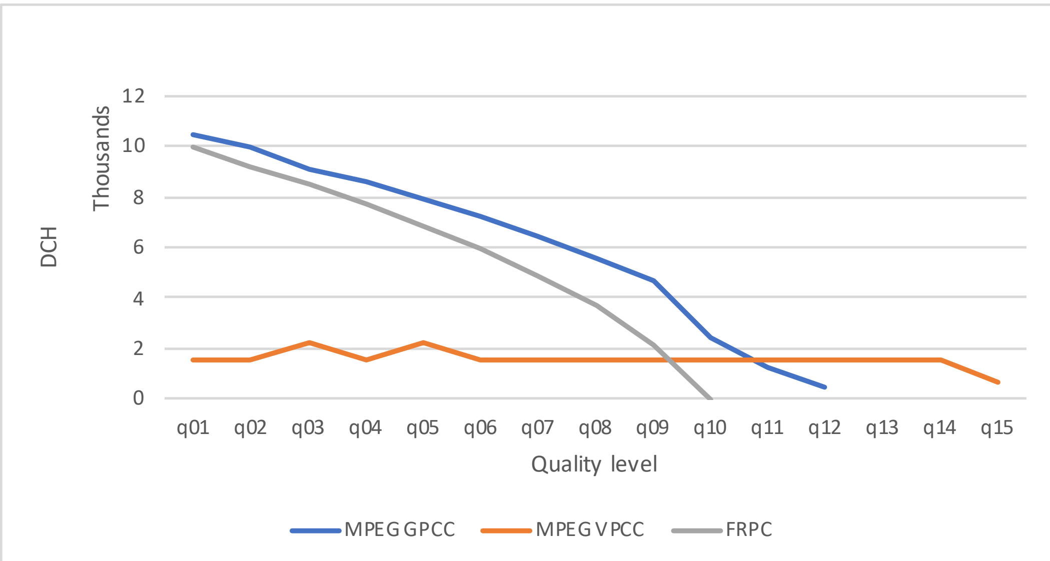
(h) K-Means - Experiment 3



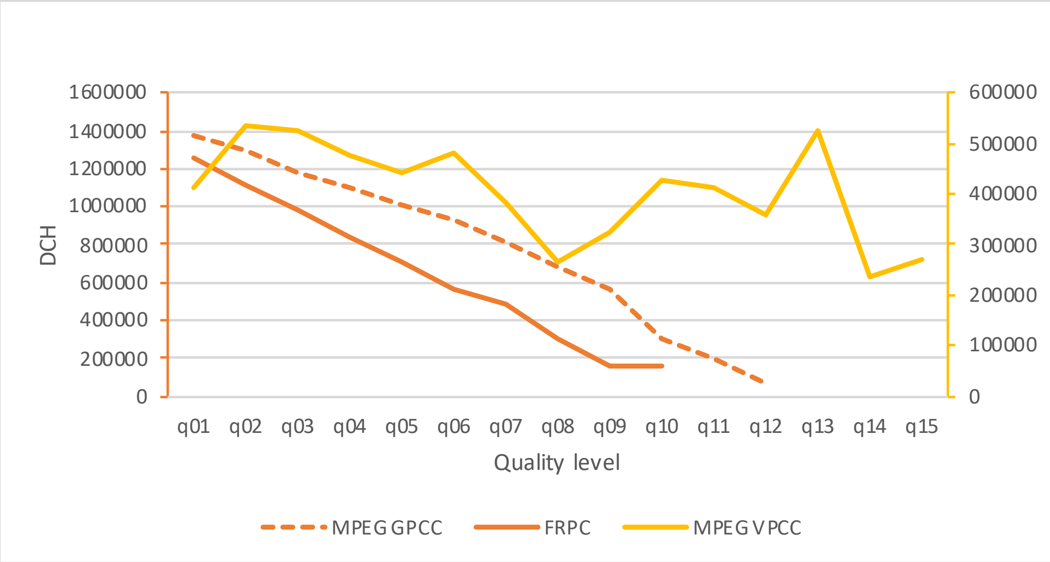
(i) Mean Shift - Experiment 1



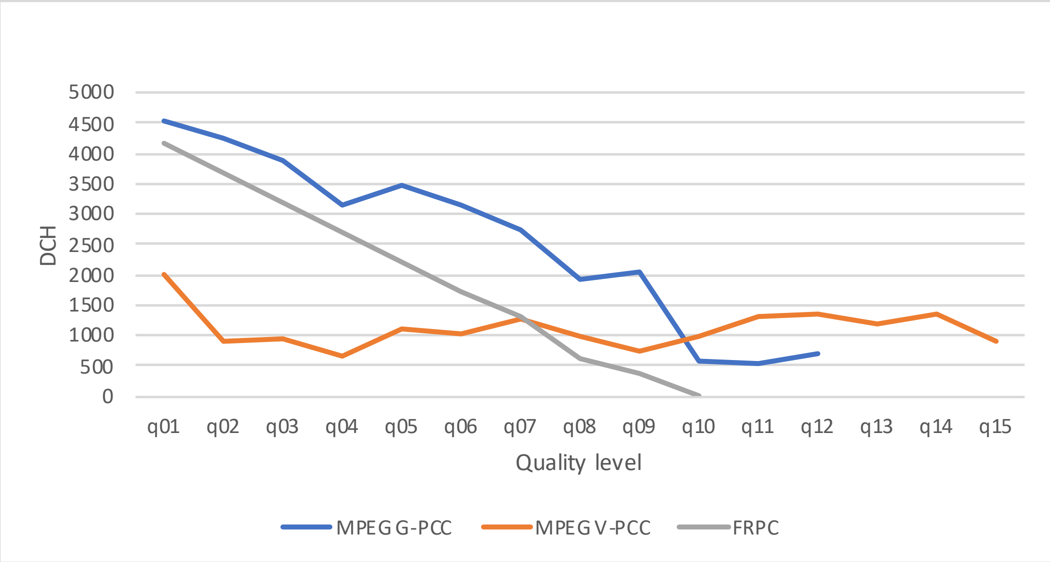
(j) Mean Shift - Experiment 2



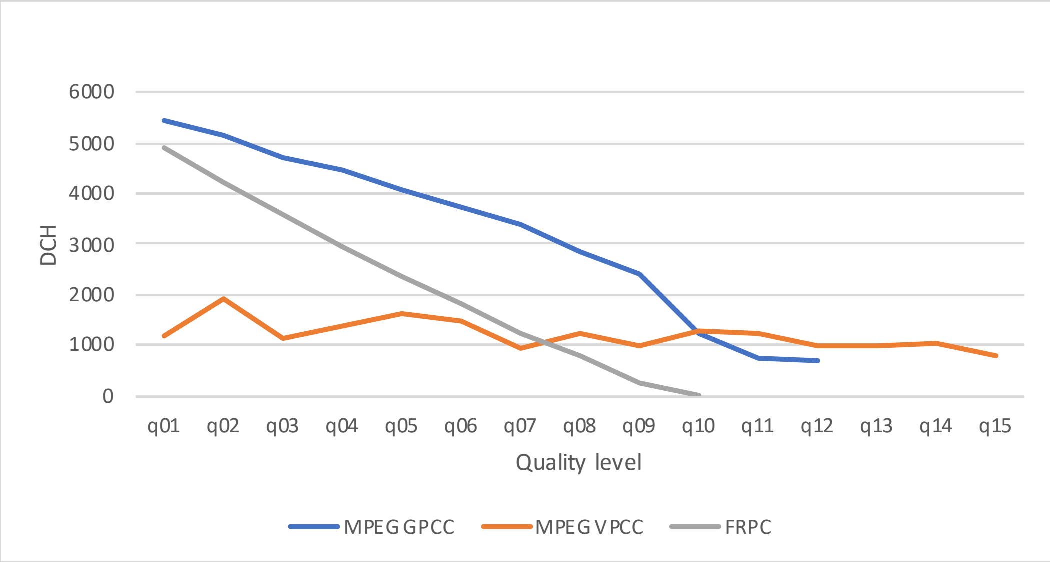
(k) Mean Shift - Experiment 3



(l) K-Medoids - Experiment 1

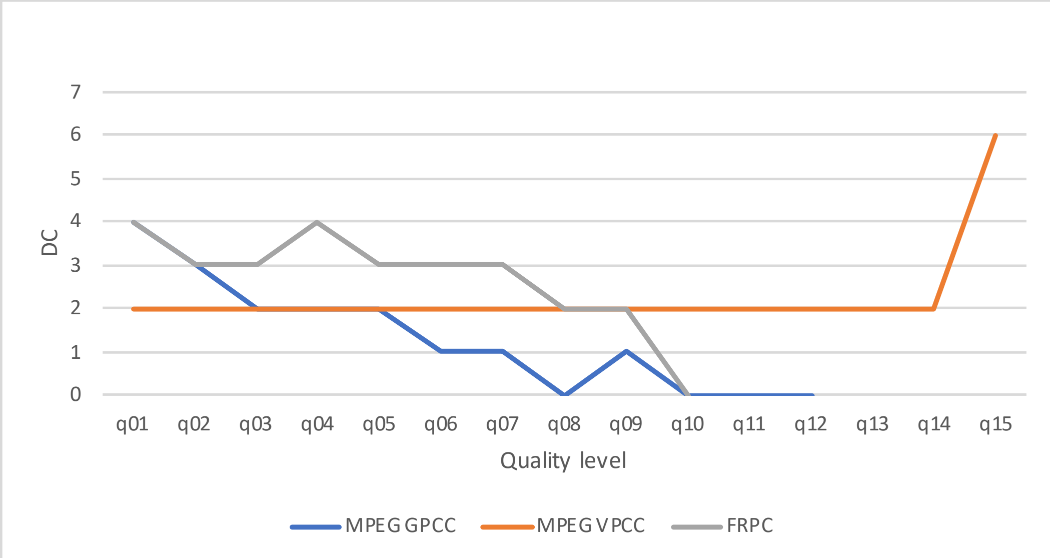


(m) K-Medoids - Experiment 2

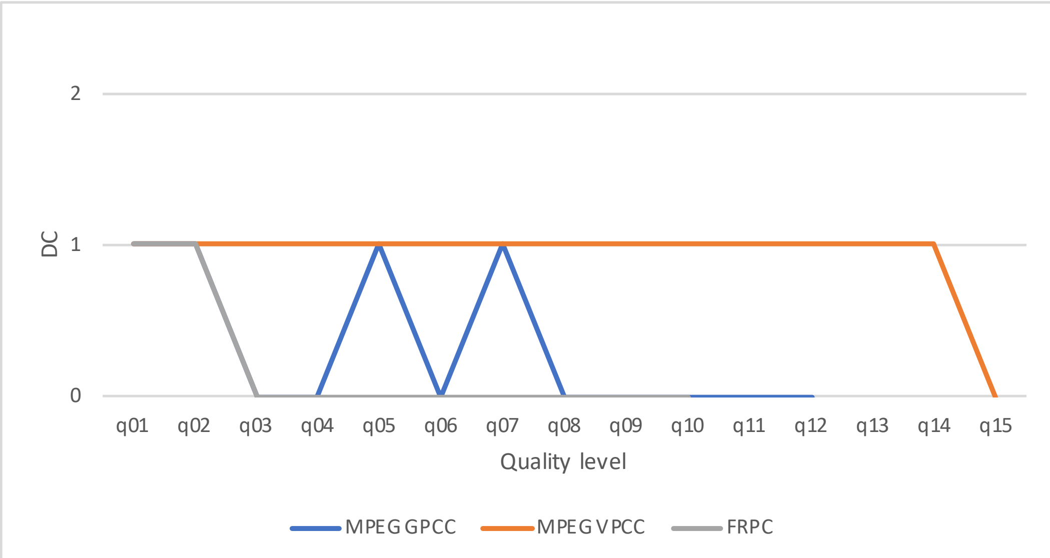


(n) K-Medoids - Experiment 3

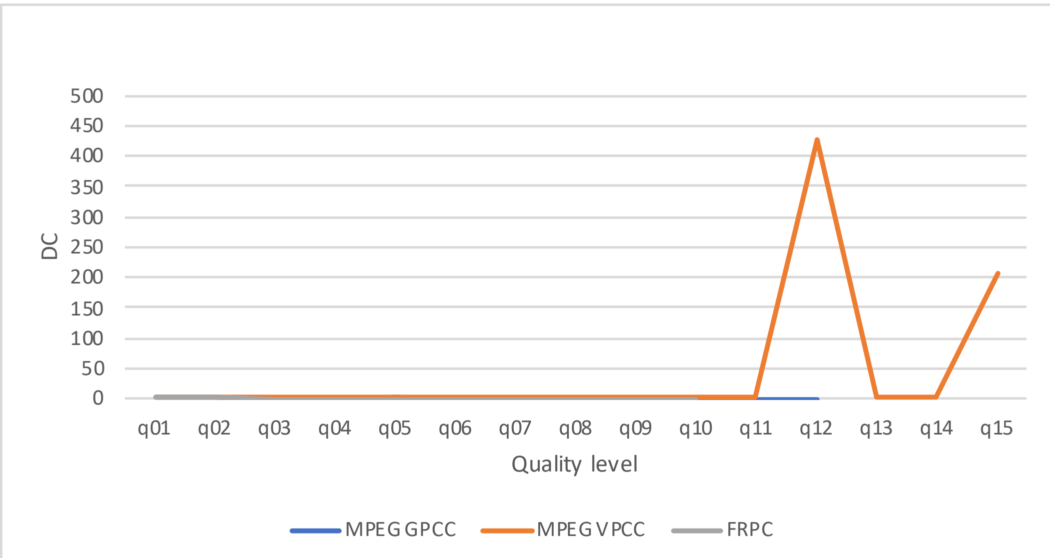
Figure B.1: DCH results.



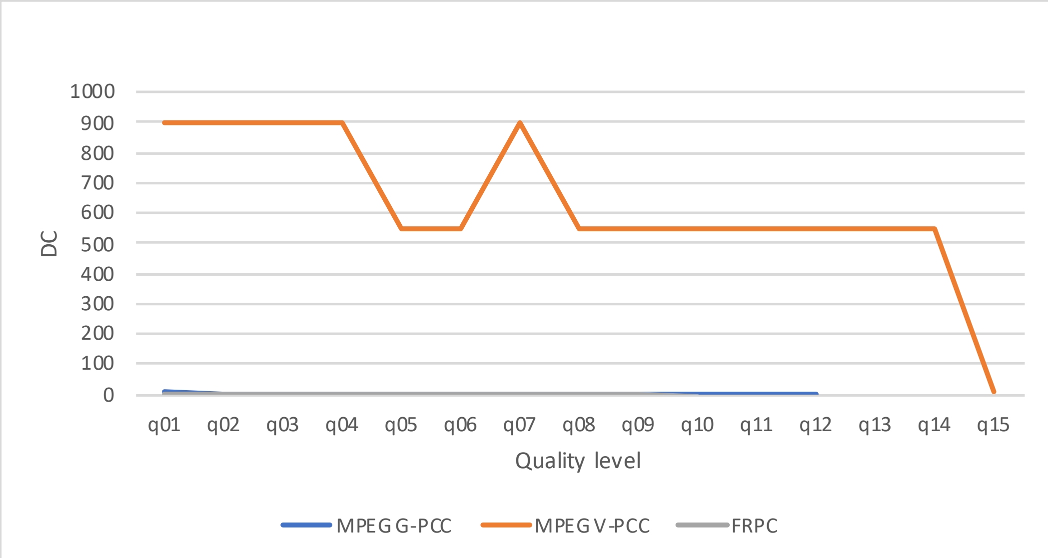
(a) Fuzzy C-Means - Experiment 1



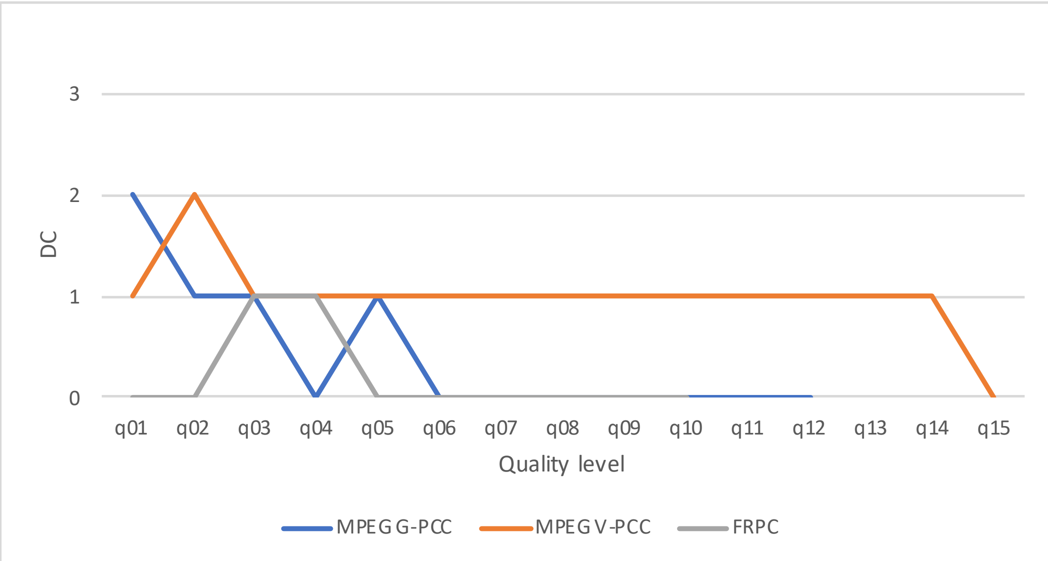
(b) Fuzzy C-Means - Experiment 2



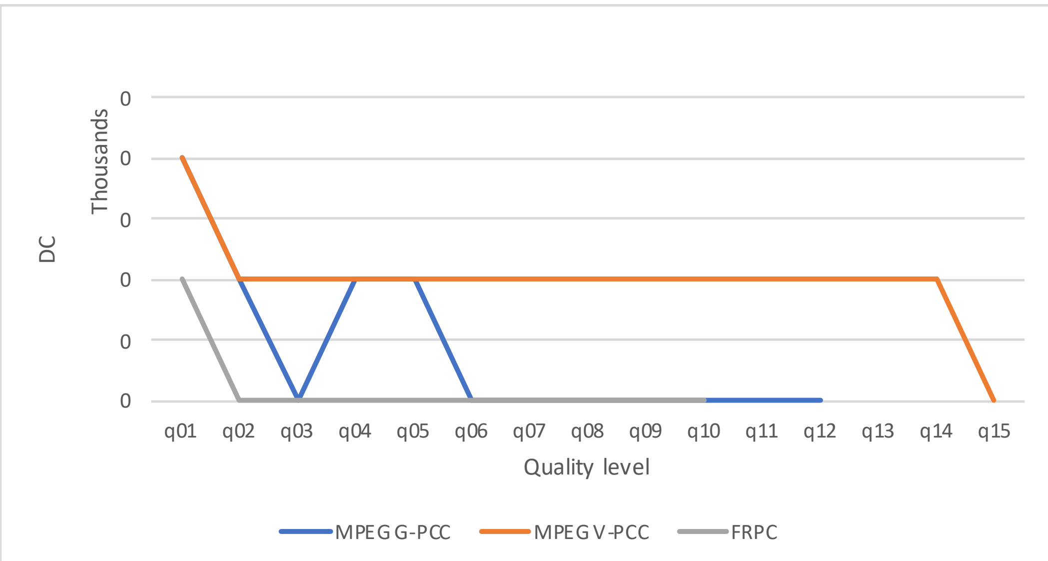
(c) Fuzzy C-Means - Experiment 3



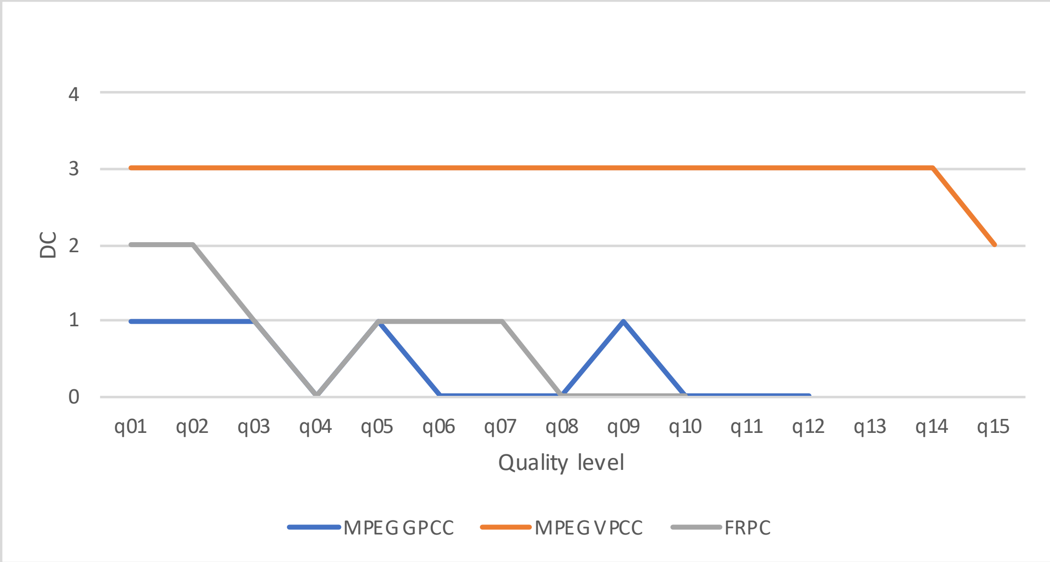
(d) Pcsegdist - Experiment 1



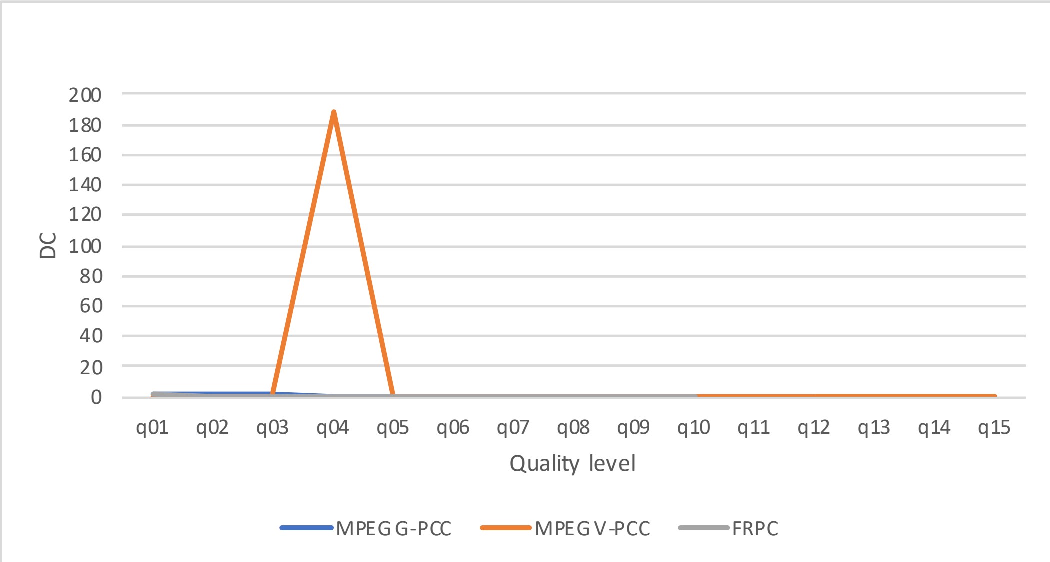
(e) Pcsegdist - Experiment 2



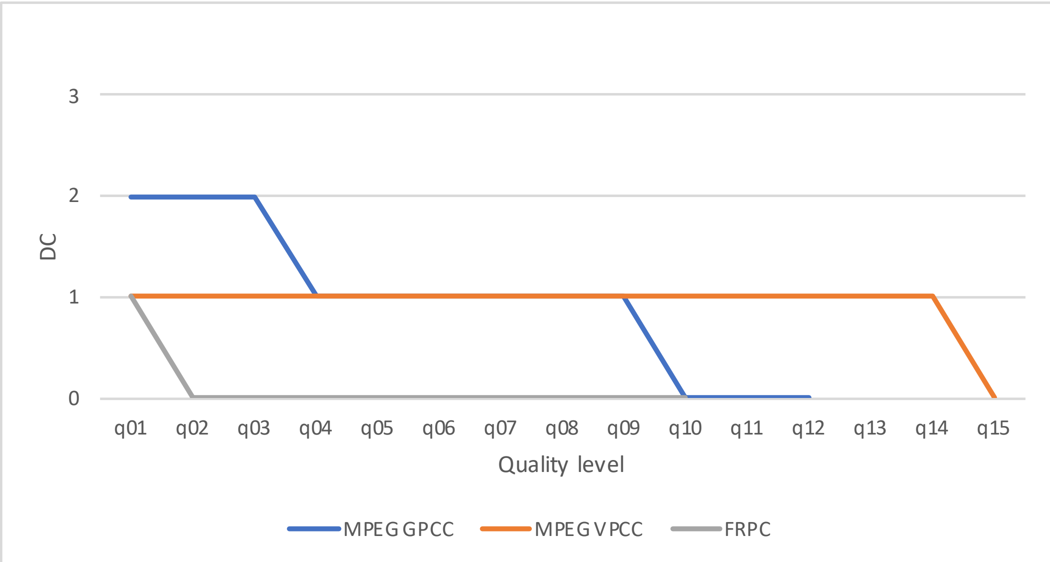
(f) Pcsegdist - Experiment 3



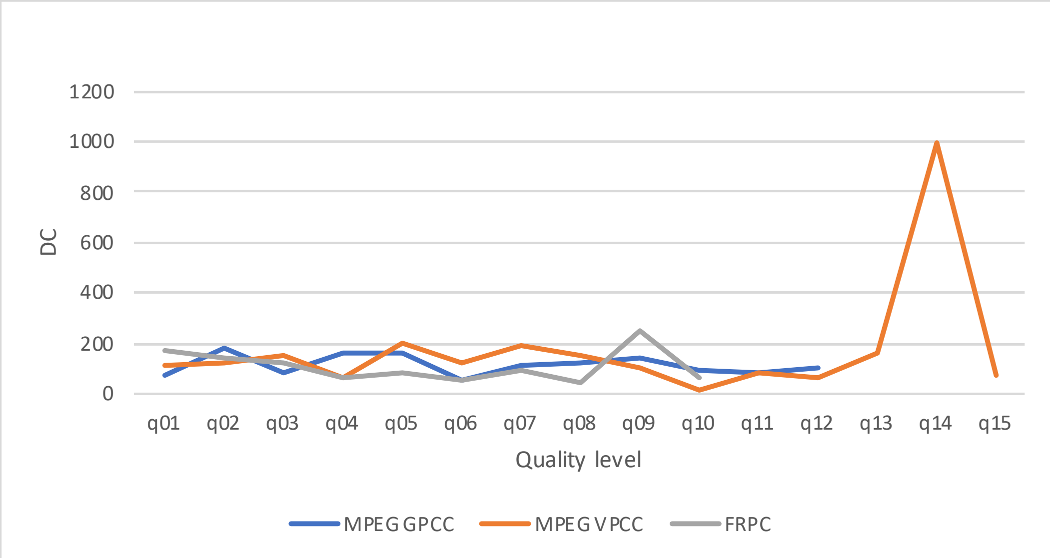
(g) Mean Shift - Experiment 1



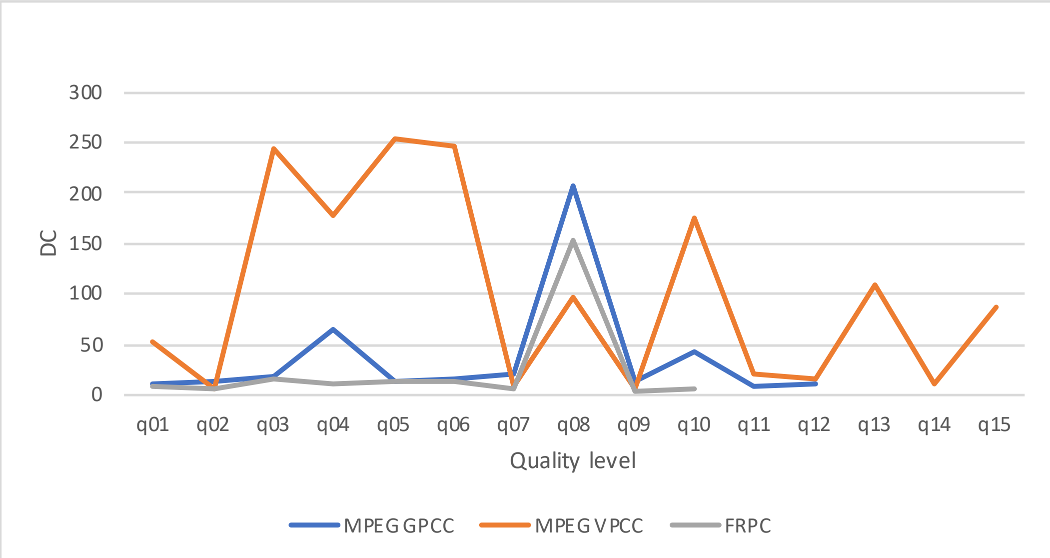
(h) Mean Shift - Experiment 2



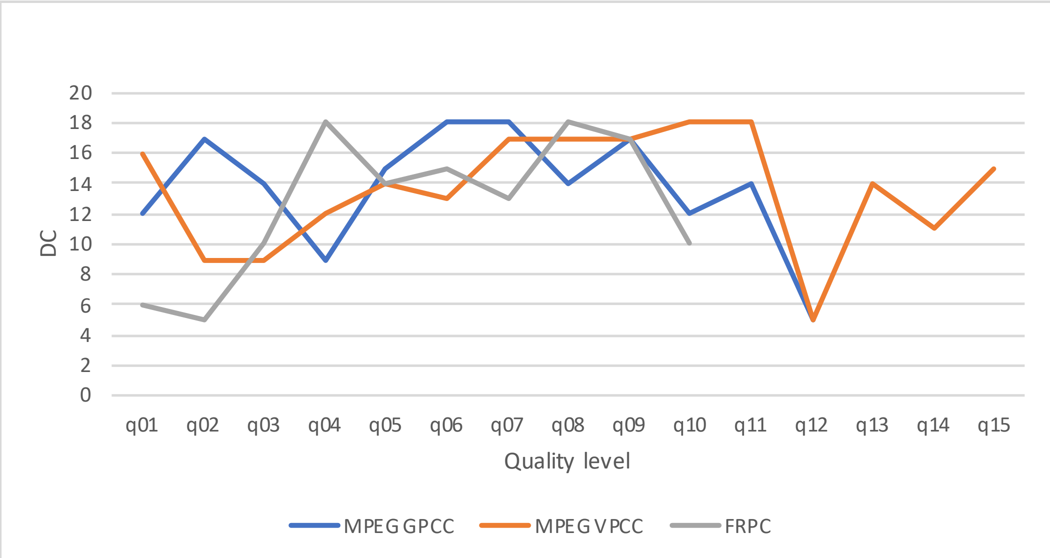
(i) Mean Shift - Experiment 3



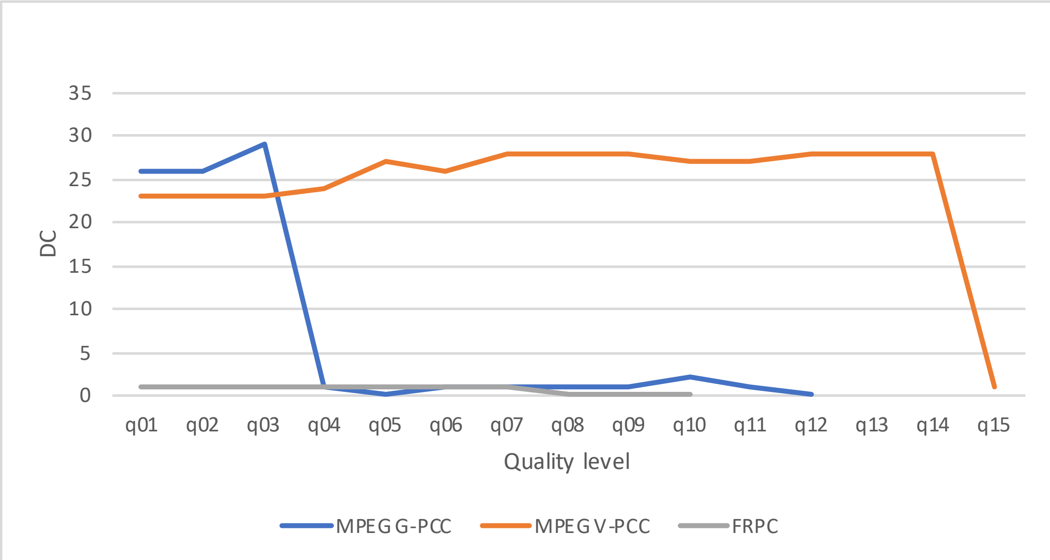
(j) K-Medoids - Experiment 1



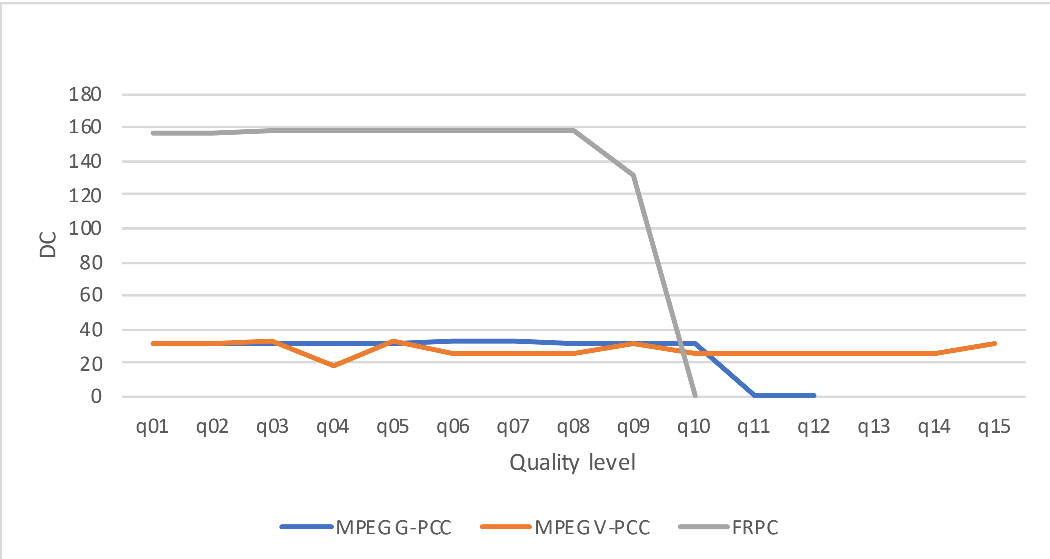
(k) K-Medoids - Experiment 2



(l) K-Medoids - Experiment 3

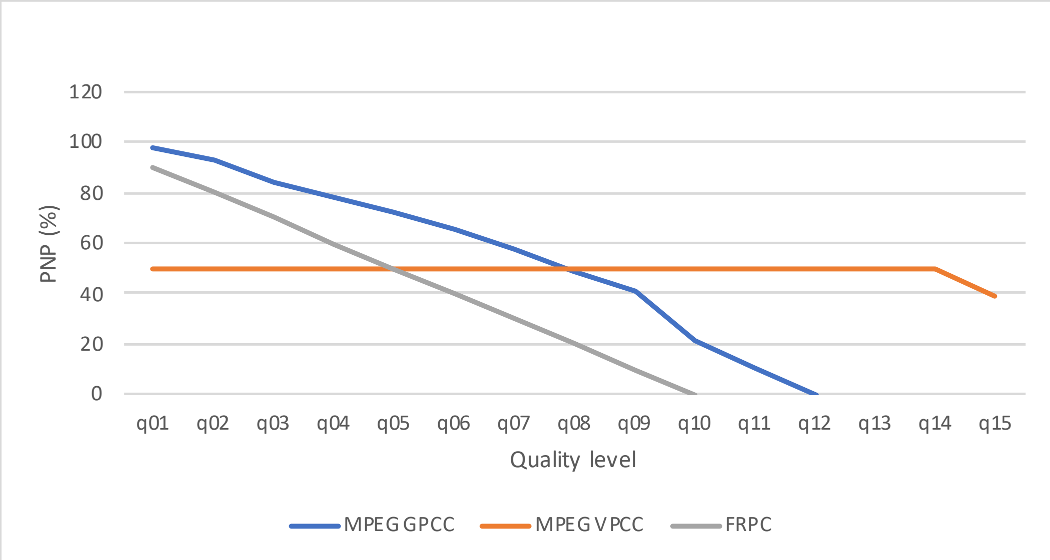


(m) Subclust - Experiment 2

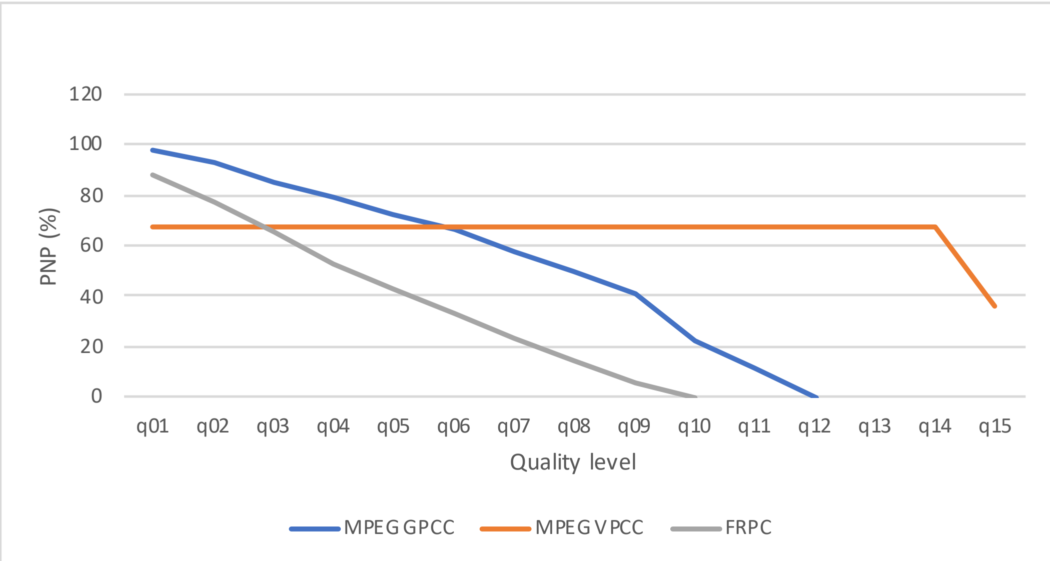


(n) Subclust - Experiment 3

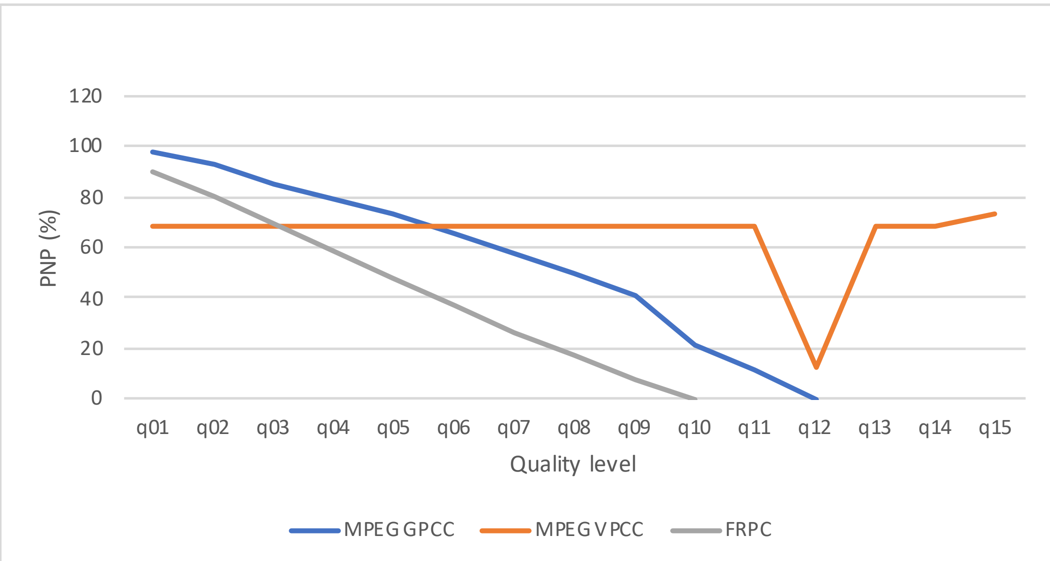
Figure B.2: DC results.



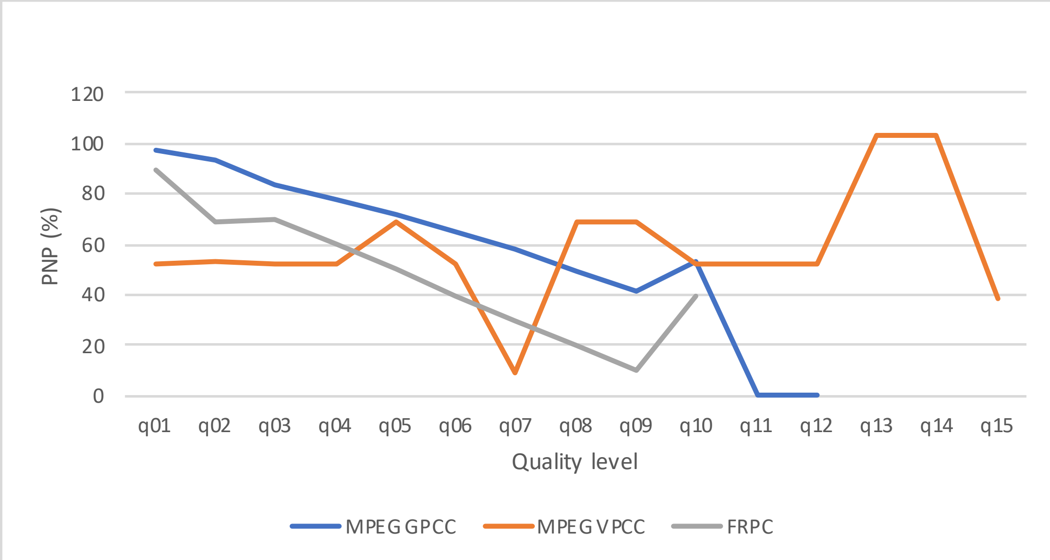
(a) Fuzzy C-Means - Experiment 1



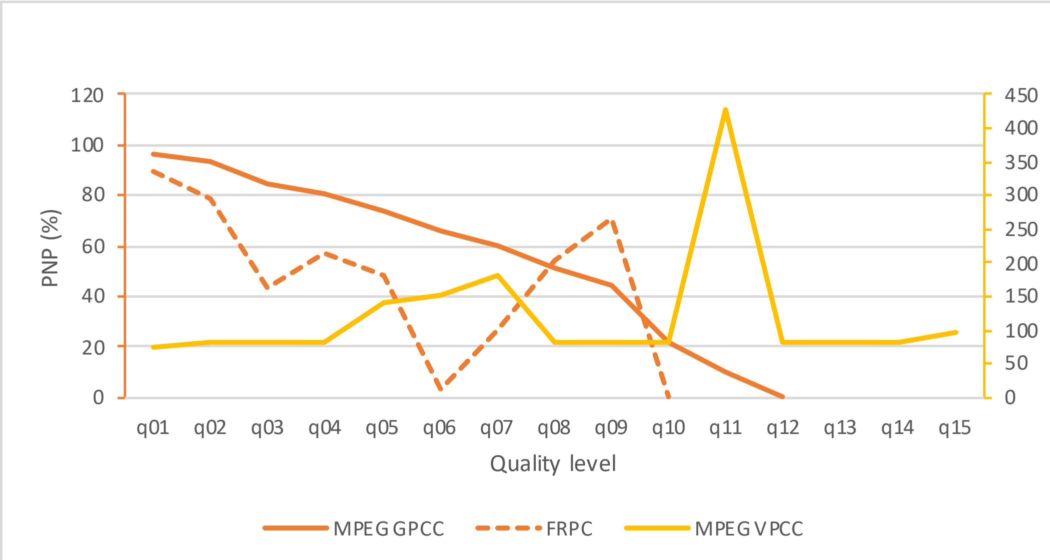
(b) Fuzzy C-Means - Experiment 2



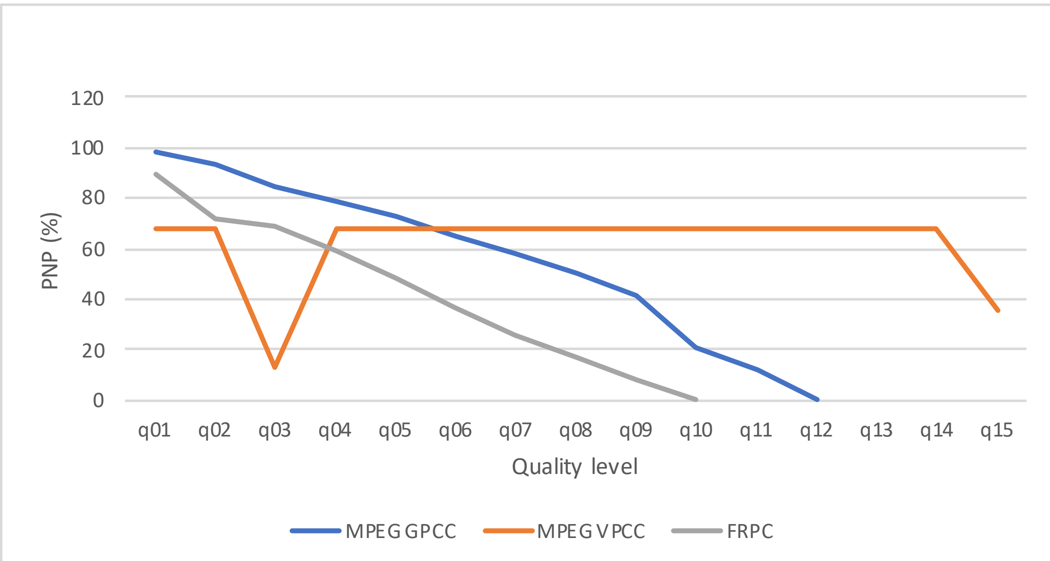
(c) Fuzzy C-Means - Experiment 3



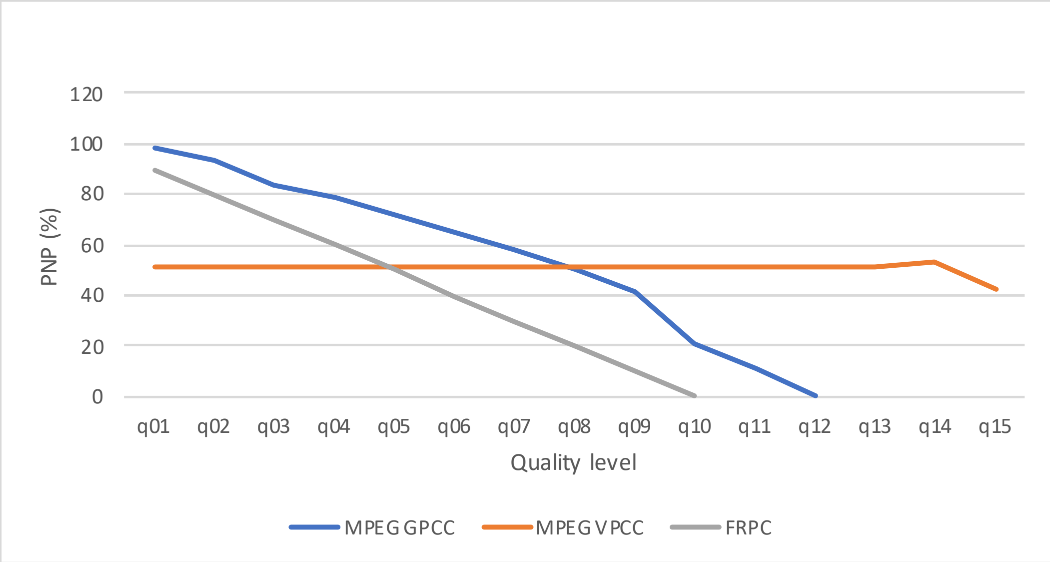
(d) K-Means - Experiment 1



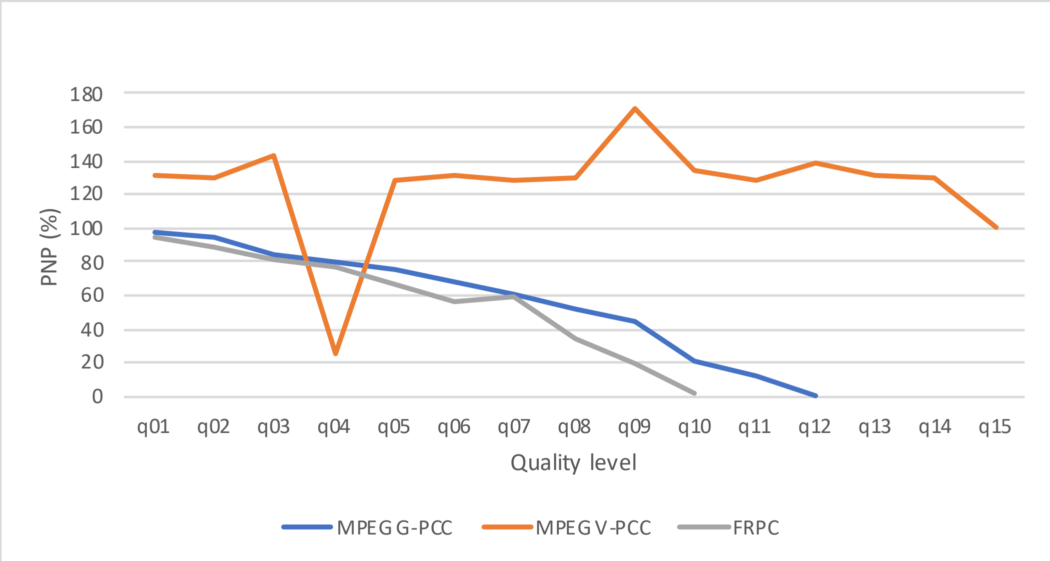
(e) K-Means - Experiment 2



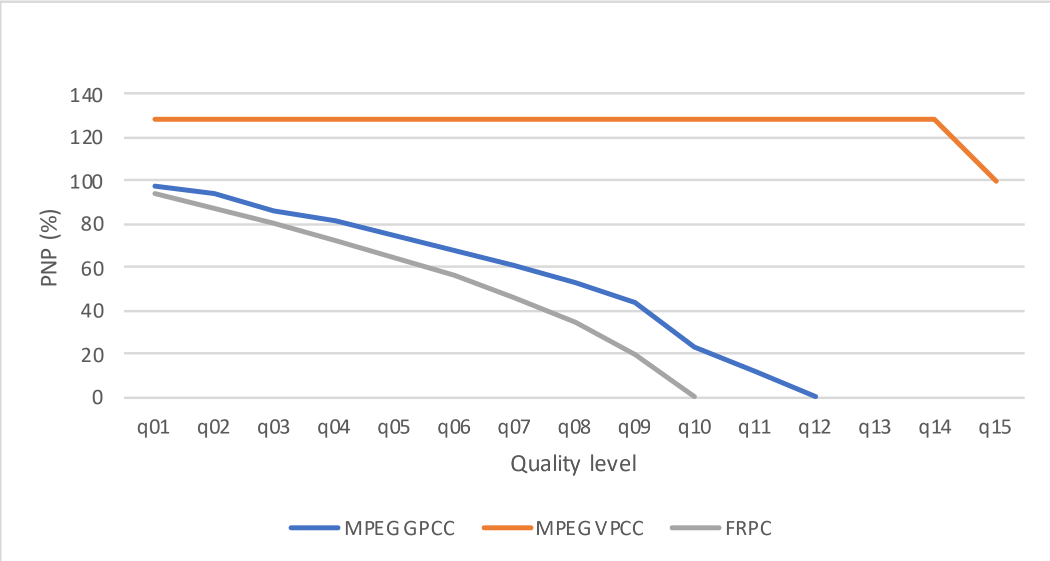
(f) K-Means - Experiment 3



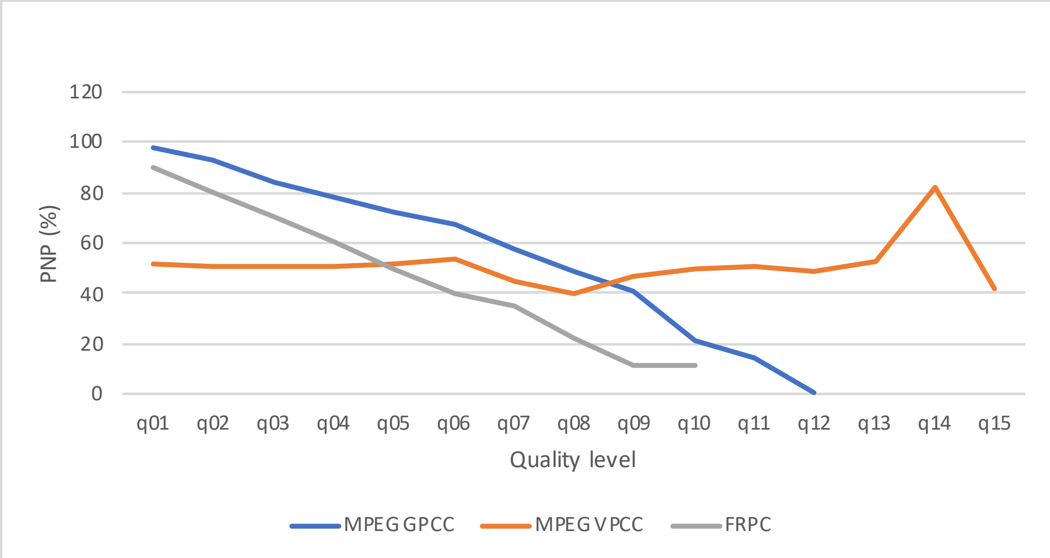
(g) Mean Shift - Experiment 1



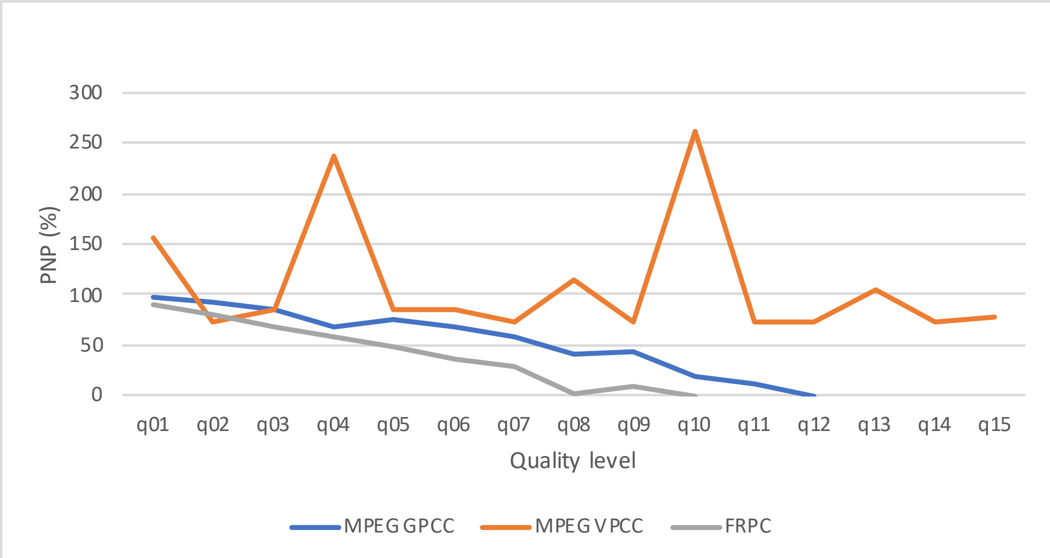
(h) Mean Shift - Experiment 2



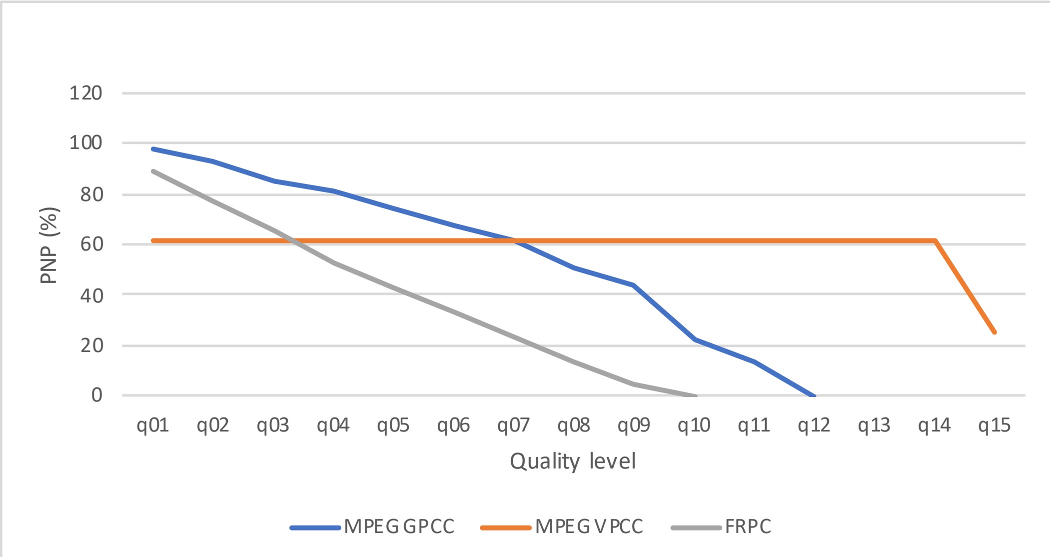
(i) Mean Shift - Experiment 3



(j) K-Medoids - Experiment 1

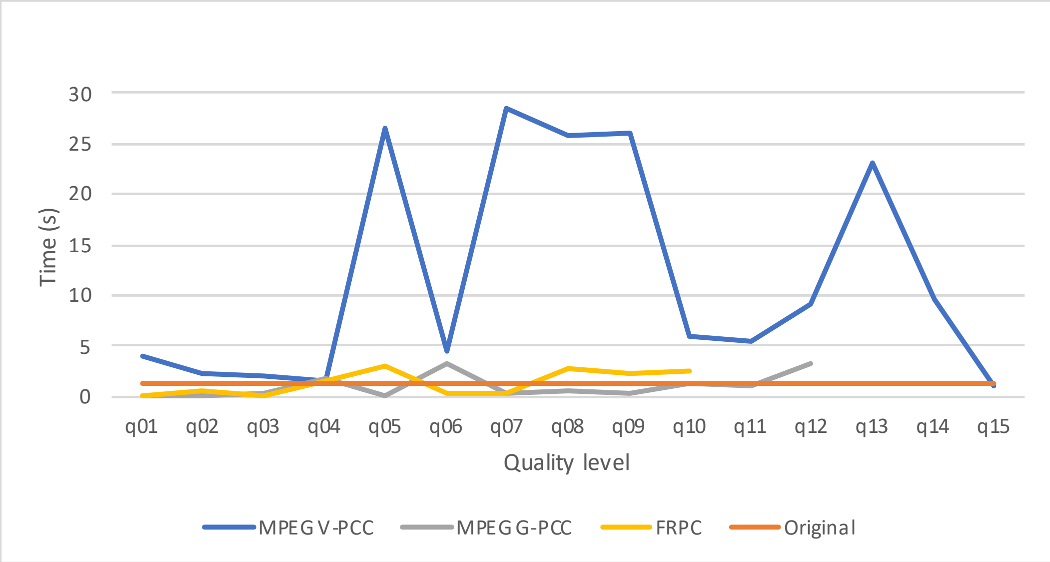


(k) K-Medoids - Experiment 2

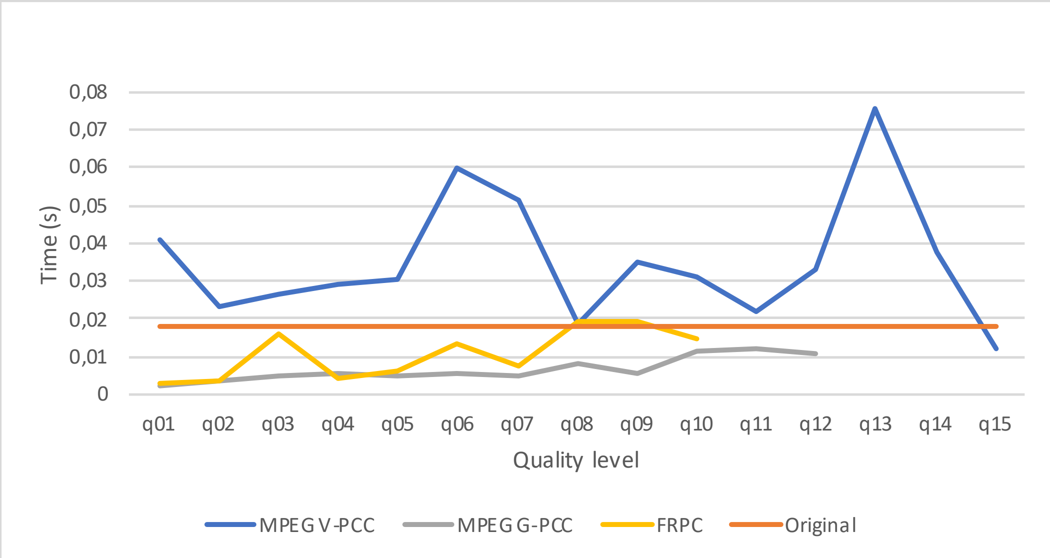


(l) K-Medoids - Experiment 3

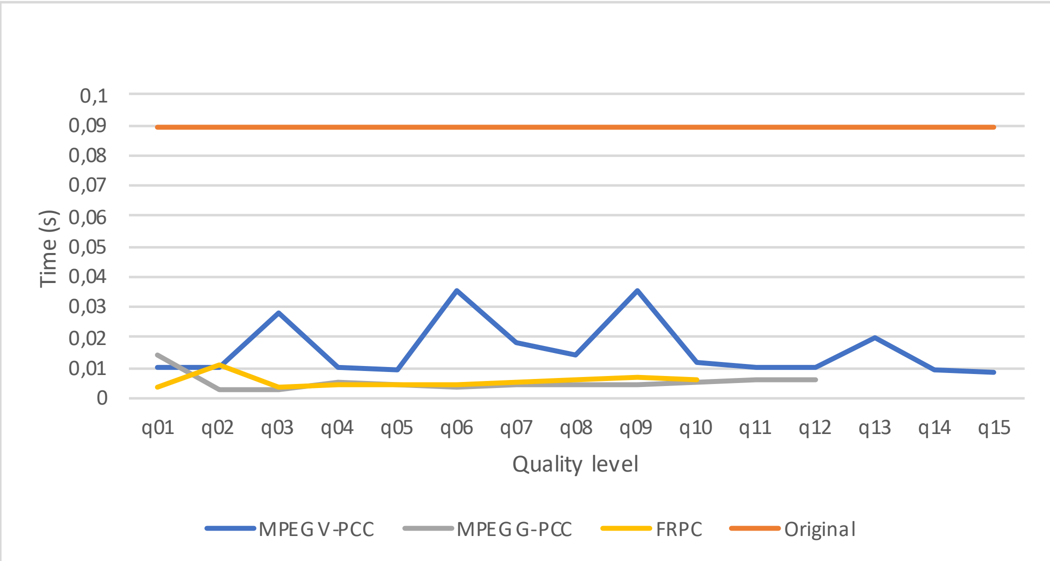
Figure B.3: PNP results.



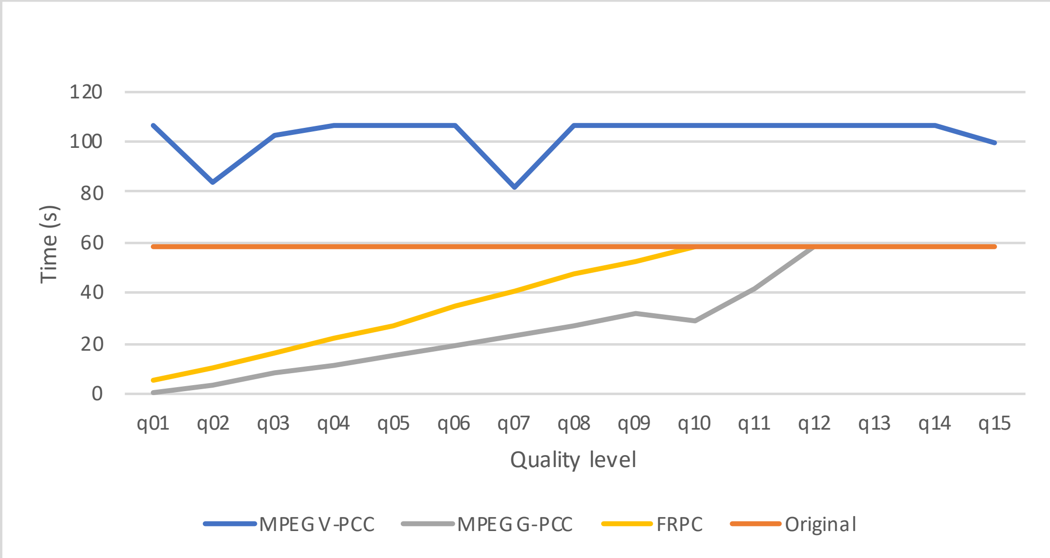
(a) K-Means - Experiment 1



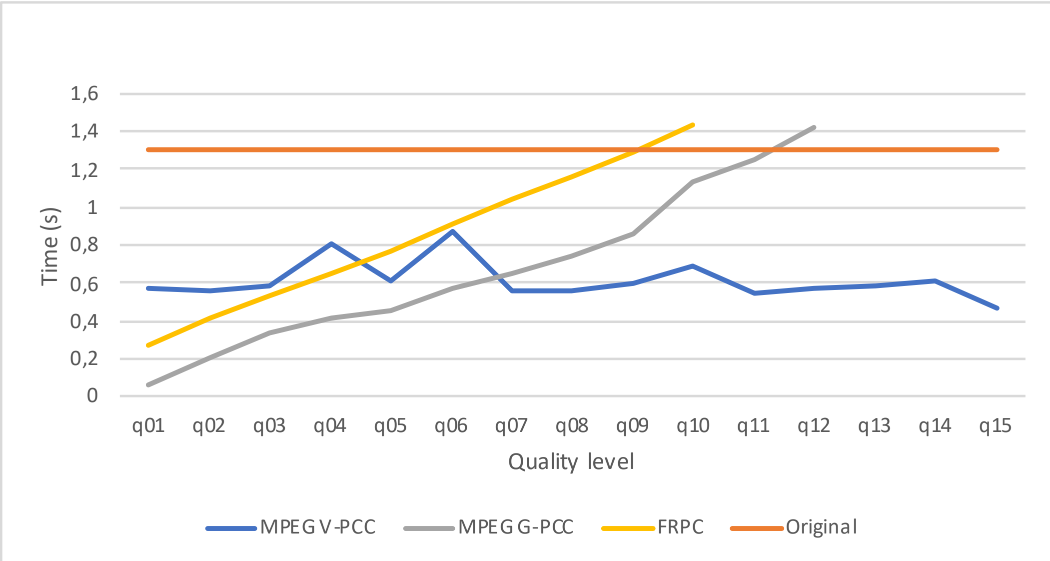
(b) K-Means - Experiment 2



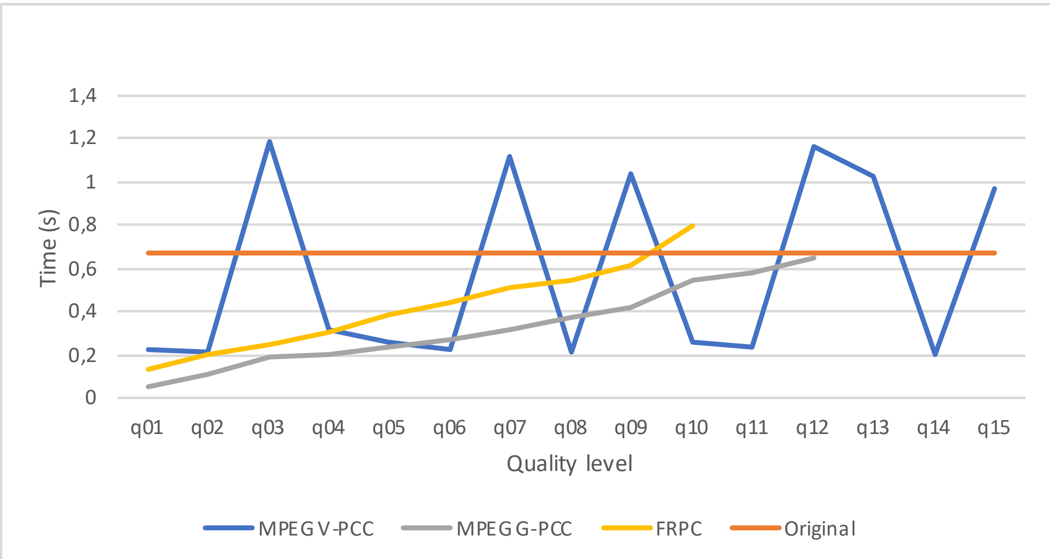
(c) K-Means - Experiment 3



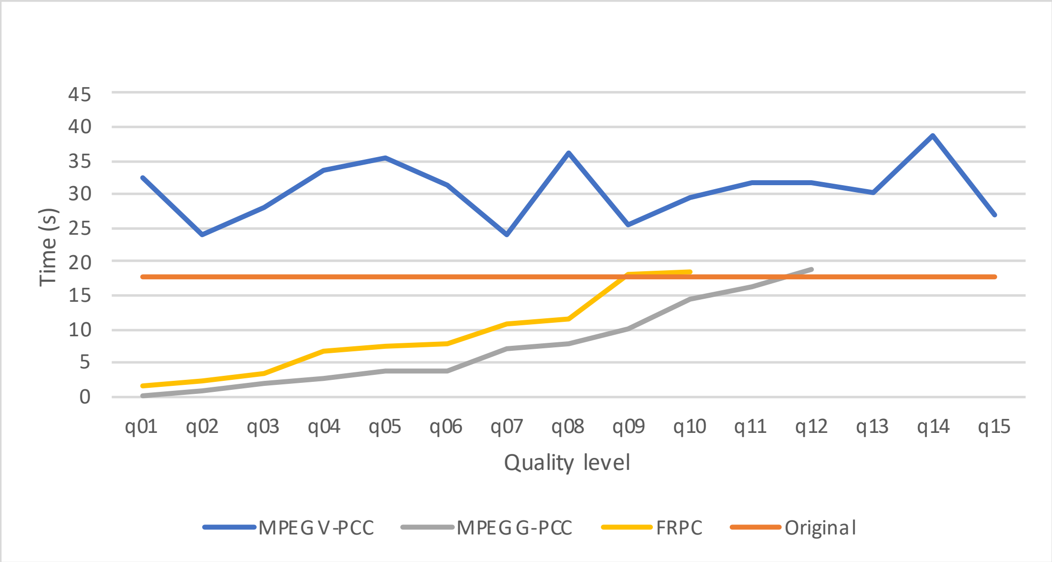
(d) Fuzzy C-Means - Experiment 1



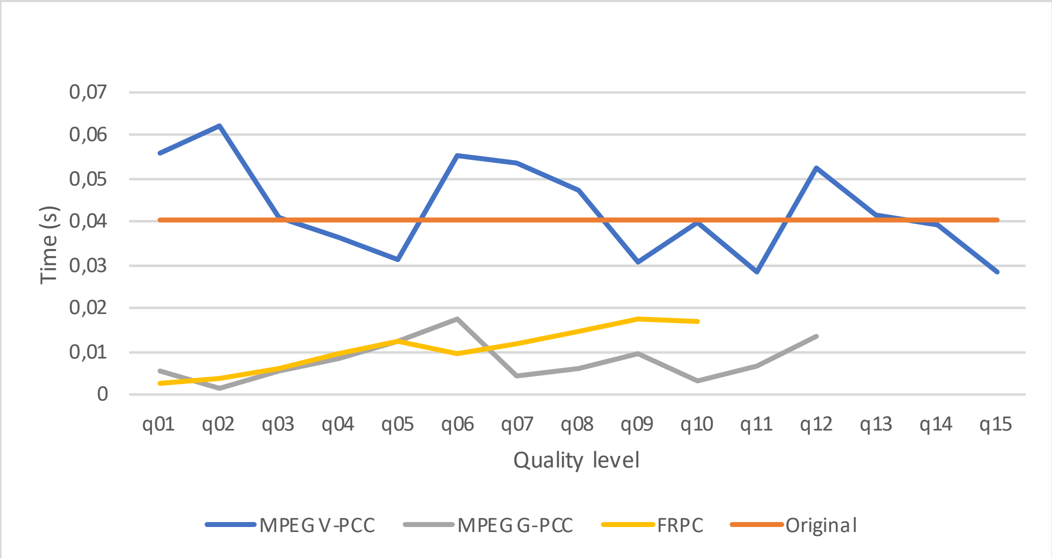
(e) Fuzzy C-Means - Experiment 2



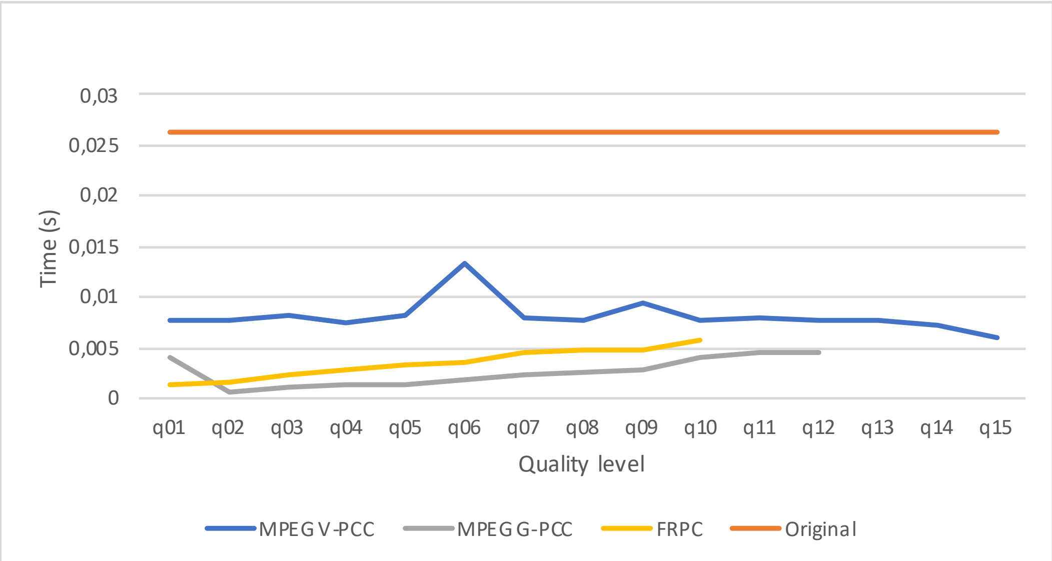
(f) Fuzzy C-Means - Experiment 3



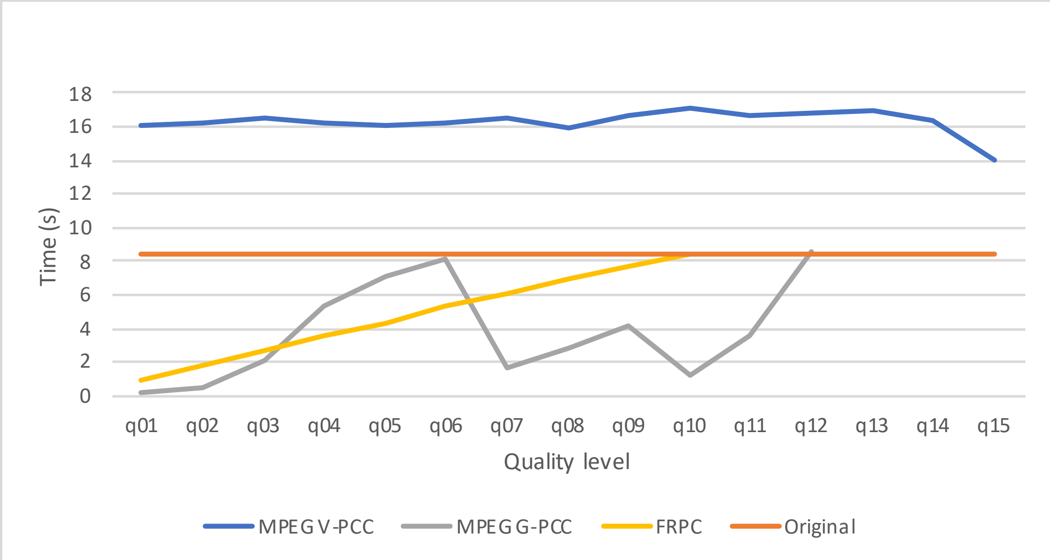
(g) Mean Shift - Experiment 1



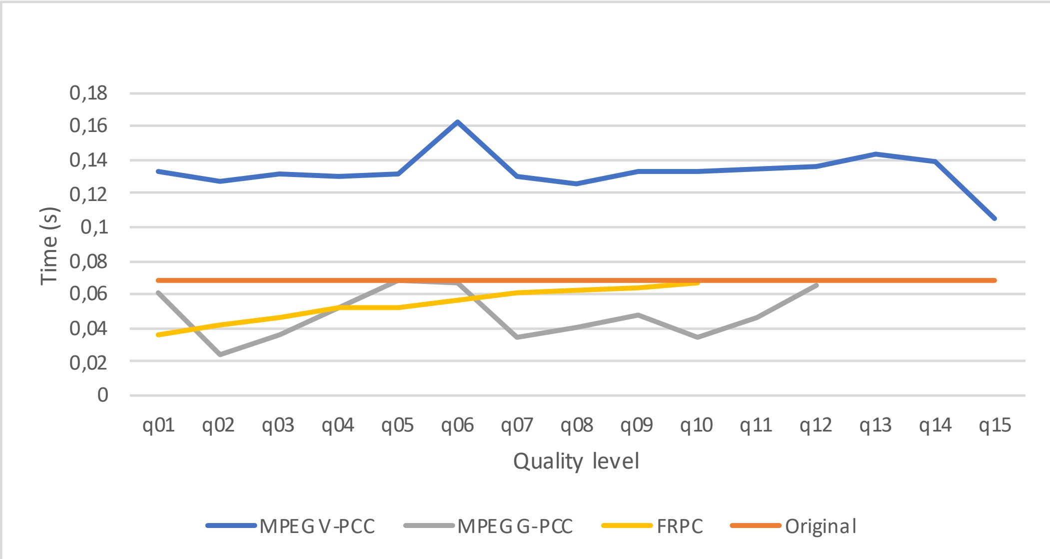
(h) Mean Shift - Experiment 2



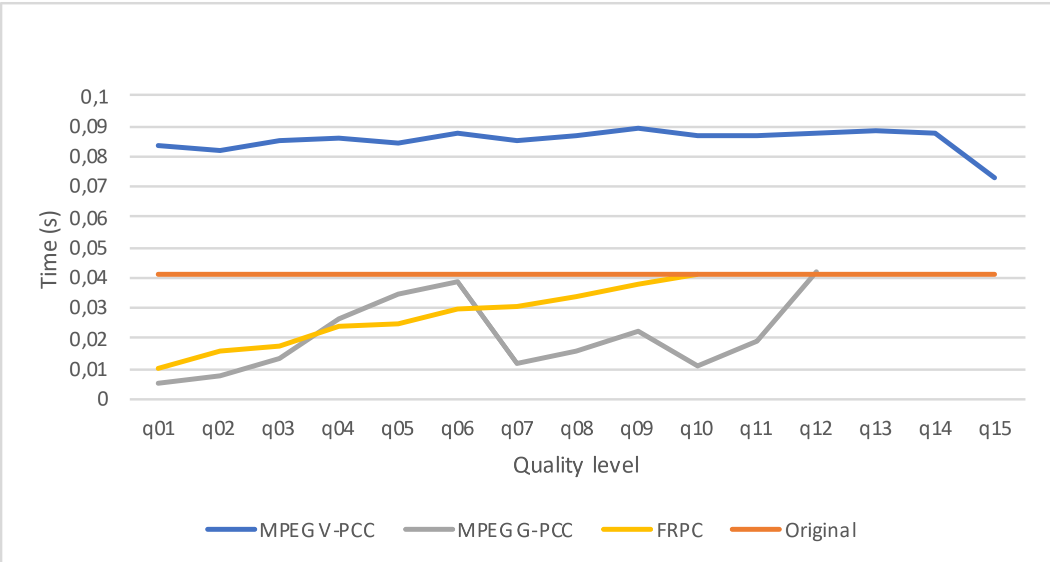
(i) Mean Shift - Experiment 3



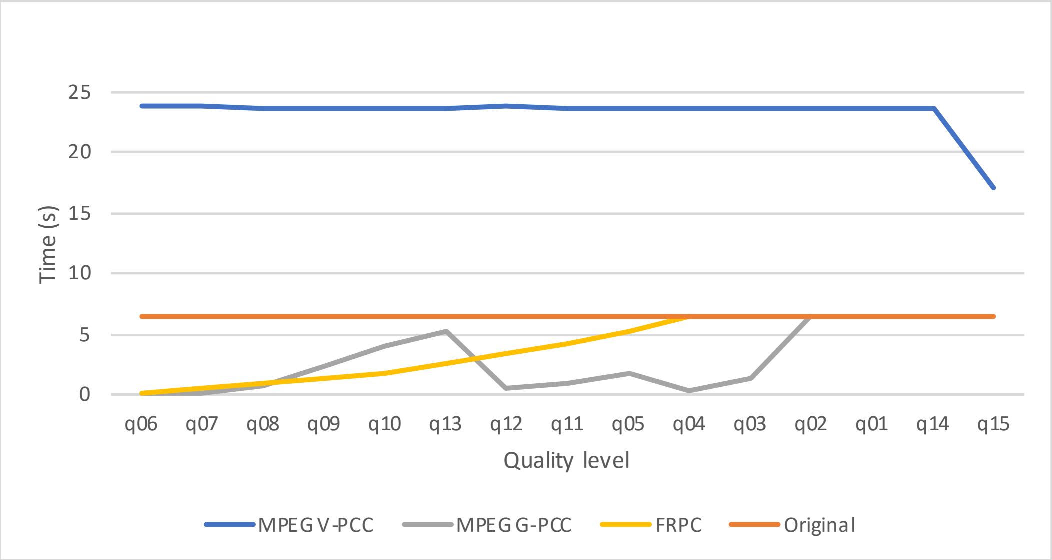
(j) Pcsegdist - Experiment 1



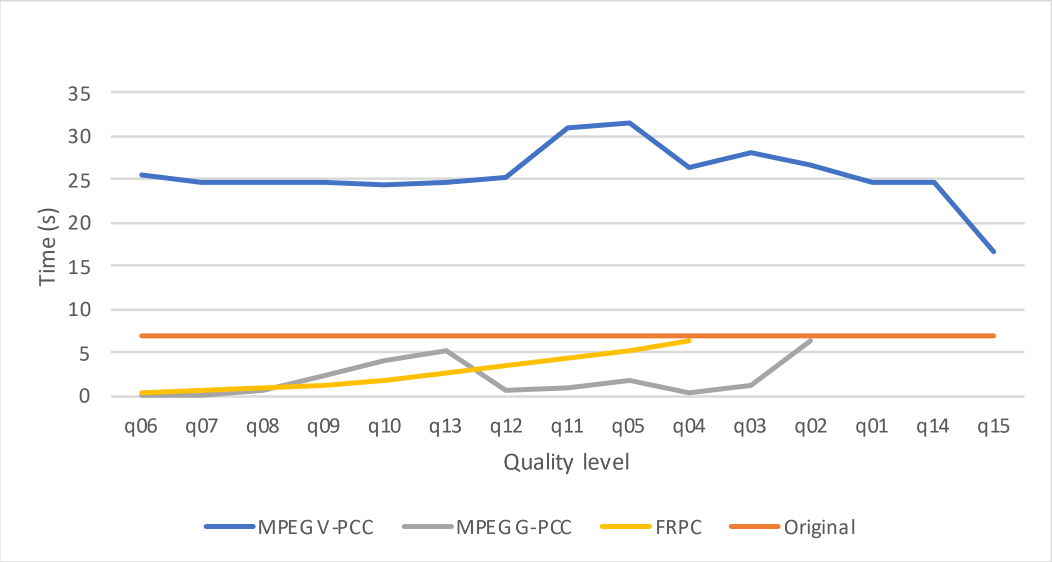
(k) Pcsegdist - Experiment 2



(l) Pcsegdist - Experiment 3



(m) Subclust - Experiment 2



(n) Subclust - Experiment 3

Figure B.4: Representation of the running time of each function according the quality level

Table B.1: Results of K-Means algorithm.

		Cluster 1						Cluster 2						Cluster 3						Cluster 4					
		C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC
Original		(1339,567,220)	0	0	0.00	0	0	(211,486,276)	0	0	0.00	0	0	(1893,524,226)	0	0	0.00	0	0	(764,510,234)	0	0	0.00	0	0
MPEG G-PCC	**q01**	(1116,553,222)	97	128947075	282.25	1272036	223	(1876,729,247)	99	13477098	35.54	1450774	1682	(1885,259,199)	99	50570351	65.22	1871932	267	(323,475,269)	97	58400882	91.17	1358633	443
	q02	(1339,568,220)	93	968547	2.12	1215964	2	(212,491,276)	93	979381	2.58	1365976	4	(1895,527,226)	93	1446151	1.87	1749359	3	(764,511,234)	93	775314	1.21	1294015	2
	q03	(1339,568,220)	84	410249	0.90	1100734	2	(211,489,276)	84	590236	1.56	1234749	3	(1894,525,226)	84	851242	1.10	1584578	2	(764,510,234)	84	541899	0.85	1173093	1
	q04	(1339,568,219)	78	385967	0.84	1027903	1	(211,489,276)	79	473647	1.25	1153188	3	(1894,526,226)	78	758759	0.98	1480410	2	(763,510,234)	79	576630	0.90	1096864	1
	q05	(1340,569,220)	72	233091	0.51	945872	2	(212,489,276)	72	293159	0.77	1062236	3	(1894,526,226)	72	480823	0.62	1362115	2	(764,511,234)	72	389787	0.61	1009601	1
	q06	(1339,568,219)	65	216104	0.47	861507	1	(211,488,276)	65	288969	0.76	956782	2	(1894,525,226)	65	527281	0.68	1233076	2	(763,510,234)	65	434820	0.68	912708	1
	q07	(1339,568,219)	58	264816	0.58	754745	2	(211,488,276)	58	247639	0.65	845248	2	(1894,525,226)	58	504953	0.65	1093255	1	(764,510,234)	58	343702	0.54	806200	1
	q08	(1339,568,219)	49	156898	0.34	647039	1	(211,488,276)	49	231695	0.61	722603	1	(1894,525,226)	49	428874	0.55	932881	1	(764,510,234)	50	331783	0.52	691785	0
	q09	(1339,568,220)	41	214327	0.47	531094	1	(211,488,276)	41	210401	0.55	591930	1	(1894,525,226)	41	343950	0.44	765102	2	(764,510,234)	41	216169	0.34	569136	1
	q10	(1885,256,199)	53	2906444	6.36	695749	629	(324,471,268)	5	82660089	218.00	201751	115	(1875,727,245)	51	27684699	35.70	961769	204	(1117,552,221)	20	107301143	167.50	200882	356
	q11	(1552,334,212)	0	61139353	133.83	244030	316	(211,487,276)	11	65223	0.17	209661	0	(1764,719,232)	19	8786588	11.33	375424	234	(764,510,234)	11	20781654	32.44	160156	0
	q12	(1338,566,219)	0	0	0.00	154469	0	(211,486,276)	0	0	0.00	112654	0	(1893,524,225)	0	0	0.00	182241	1	(763,509,233)	0	0	0.00	113518	0
MPEG V-PCC	**q01**	(1338,560,220)	52	61795982	135.27	341474	7	(212,480,276)	48	16958843	44.72	313213	7	(1894,522,227)	50	28649975	36.95	457087	3	(763,504,235)	51	58847827	91.87	540624	6
	q02	(1338,560,220)	53	59918797	131.16	279509	7	(212,480,276)	48	16465844	43.42	306592	7	(1894,522,227)	50	26890489	34.68	484015	3	(763,504,235)	51	58105400	90.71	602499	6
	q03	(1338,560,220)	52	56961349	124.68	348500	7	(212,480,276)	48	15159379	39.98	385356	7	(1894,522,227)	50	26222176	33.82	513719	3	(763,504,235)	51	51430152	80.29	432664	6
	q04	(1338,560,220)	52	52956482	115.92	299096	7	(212,480,276)	48	14253716	37.59	306856	7	(1894,522,227)	50	23139856	29.84	454320	3	(763,504,235)	51	49288444	76.94	410574	6
	q05	(1544,328,213)	69	72860619	159.48	229241	315	(212,480,276)	48	13934439	36.75	385677	7	(1767,716,233)	38	17610008	22.71	236040	230	(765,504,235)	51	71316578	111.33	403152	6
	q06	(1338,560,220)	52	46038634	100.77	221599	7	(212,480,276)	48	13632344	35.95	325222	7	(1894,522,227)	50	15362430	19.81	425270	3	(763,504,235)	51	48641322	75.93	368030	6
	q07	(1886,252,201)	9	7755083	16.98	141238	632	(282,447,272)	77	66859446	176.32	310157	81	(1828,737,244)	3	8552724	11.03	210703	223	(1058,535,224)	141	121337043	189.41	495536	296
	q08	(1531,335,213)	69	74632903	163.36	200739	301	(212,480,276)	48	13060336	34.44	325226	7	(1777,709,233)	38	18563538	23.94	253937	219	(764,505,235)	51	69187737	108.01	438619	5
	q09	(1544,328,213)	69	72643757	159.01	236292	315	(212,480,276)	48	12972966	34.21	318632	7	(1767,716,233)	38	16864356	21.75	264268	230	(765,505,235)	51	70197434	109.58	523973	6
	q10	(1337,560,220)	52	43128266	94.40	281762	7	(212,480,276)	48	12832208	33.84	303917	7	(1894,522,227)	50	970343	1.25	355722	3	(763,504,235)	51	48310298	75.42	382440	6
	q11	(1337,560,220)	52	43174410	94.50	322885	7	(212,480,276)	48	12926129	34.09	247657	7	(1894,522,227)	50	898026	1.16	298621	3	(763,504,235)	51	48342545	75.47	456849	6
	q12	(1337,560,220)	52	42781566	93.64	319371	7	(212,480,276)	48	12890388	34.00	325143	7	(1894,522,227)	50	956431	1.23	250187	3	(763,504,235)	51	48182933	75.22	571218	6
	q13	(286,448,272)	103	60611493	132.67	307470	1061	(1835,735,244)	29	33037325	87.13	191887	1643	(1064,538,223)	78	108021310	139.31	363586	829	(1886,251,201)	15	10170089	15.88	214294	1153
	q14	(1679,722,238)	103	65645825	143.69	347226	374	(212,480,276)	48	21007125	55.40	332965	7	(1882,273,196)	30	15784986	20.36	576303	253	(908,465,232)	111	108262606	169.00	434141	151
	q15	(1337,561,220)	39	0	0.00	182927	6	(211,482,275)	41	0	0.00	292400	5	(1893,519,225)	39	0	0.00	232629	6	(764,507,234)	42	0	0.00	271951	3
FRPC	**q01**	(1339,569,220)	90	443645	0.97	1178936	3	(212,489,275)	90	225154	0.59	1325979	2	(1895,528,226)	90	339909	0.44	1697475	4	(763,512,233)	90	314370	0.49	1256030	2
	q02	(328,475,267)	69	75757212	165.83	910581	1016	(1876,728,246)	84	11110729	29.30	1236509	1683	(1124,554,221)	78	92628277	119.46	1464312	770	(1885,259,199)	89	38409042	59.96	1237757	1150
	q03	(1339,569,220)	70	232254	0.51	914898	3	(211,490,276)	70	65203	0.17	1032747	3	(1894,527,226)	70	138991	0.18	1318401	3	(763,512,233)	70	141963	0.22	979453	2
	q04	(1339,569,220)	60	163590	0.36	783352	2	(211,489,276)	60	38316	0.10	884737	3	(1894,527,226)	60	72222	0.09	1129529	4	(764,512,233)	60	109915	0.17	840879	2
	q05	(1339,569,220)	50	147476	0.32	652104	2	(211,489,276)	50	29912	0.08	737422	3	(1894,527,226)	50	50032	0.06	940713	3	(764,512,234)	50	82364	0.13	701766	2
	q06	(1339,569,219)	40	91953	0.20	520460	2	(211,489,276)	40	22941	0.06	590710	3	(1894,527,226)	40	38272	0.05	751685	3	(764,512,233)	40	74086	0.12	562541	2
	q07	(1339,569,220)	30	75428	0.17	390842	2	(211,489,276)	30	7777	0.02	443575	2	(1893,527,226)	30	20246	0.03	562294	2	(764,511,234)	30	34314	0.05	423395	2
	q08	(1339,568,220)	20	65315	0.14	259134	1	(211,488,276)	20	4494	0.01	296128	2	(1893,526,226)	20	11055	0.01	374568	2	(764,511,234)	20	11934	0.02	283946	1
	q09	(1339,568,220)	10	32844	0.07	127473	1	(211,488,276)	10	410	0.00	148904	1	(1893,526,226)	10	2193	0.00	186630	2	(763,511,234)	10	2672	0.00	143750	1
	q10	(1885,255,199)	40	2868739	6.28	526608	629	(324,471,268)	34	82452920	217.45	186167	114	(1874,726,245)	38	27757982	35.80	711166	204	(1117,552,221)	53	107197580	167.34	211378	356

Table B.2: Results of the K-Medoids algorithm

		Cluster 1						Cluster 2						Cluster 3						Cluster 4					
		C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC
Original		(782,573,262)	0	0	0	0	0	(1946,586,244)	0	0	0	0	0	(119,465,236)	0	0	0	0	0	(1409,641,255)	0	0	0	0	0
MPEG G-PCC	**q01**	(844,595,237)	98	7381476	12	1371564	70	(1820,595,277)	98	23328572	30	1851604	131	(260,403,213)	98	23817450	63	1442814	156	(1380,523,293)	98	538663	1	1287645	127
	q02	(856,407,249)	93	775314	1	1294015	182	(1916,635,297)	93	22170577	29	1748607	78	(280,503,301)	93	979381	3	1365976	178	(1304,499,297)	93	1588807	3	1216690	182
	q03	(713,622,264)	84	20905748	33	1175530	84	(1900,558,245)	84	18918314	24	1584001	54	(177,414,317)	84	16799335	44	1232188	112	(1273,550,114)	84	602120	1	1101343	216
	q04	(733,417,238)	78	17362241	27	1093411	165	(1956,460,268)	78	758759	1	1480410	128	(169,540,192)	79	885833	2	1156642	100	(1350,625,261)	78	385967	1	1027903	61
	q05	(672,539,149)	72	13799155	22	1009141	161	(1878,497,239)	72	17042859	22	1361337	112	(270,601,267)	72	224713	1	1062697	206	(1264,495,261)	72	747367	2	946634	206
	q06	(738,536,266)	67	1729668	3	928916	57	(1883,407,222)	65	527281	1	1233076	191	(281,408,256)	64	18463002	49	940574	173	(1328,577,129)	65	216104	0	861507	162
	q07	(681,537,242)	58	18560580	29	815925	109	(1826,640,291)	56	33196210	43	1068843	139	(265,393,219)	57	17039557	45	835523	163	(1222,525,205)	59	5058136	11	779101	226
	q08	(689,492,254)	49	42542955	66	687098	124	(1934,580,257)	49	428874	1	932881	18	(224,573,204)	50	553897	1	727011	154	(1413,749,280)	49	95654	0	647411	111
	q09	(828,434,250)	41	216169	0	569136	147	(1793,608,250)	39	33510279	43	728639	154	(215,424,313)	41	210401	1	591930	129	(1223,526,246)	43	5417970	12	568392	219
	q10	(727,499,255)	21	15211701	24	297210	92	(1915,387,282)	21	212727	0	402244	205	(260,604,272)	22	480791	1	307995	201	(1433,482,168)	21	72431	0	278269	183
	q11	(852,591,223)	14	1471251	2	193517	82	(1911,447,267)	11	152827	0	232148	145	(257,511,241)	8	18255426	48	209661	145	(1333,677,152)	11	88098	0	217595	133
	q12	(680,552,233)	1	762874	1	72982	108	(1878,404,129)	0	0	0	182241	226	(290,442,308)	1	14001235	37	162911	187	(1303,612,254)	0	0	0	154469	110
MPEG V-PCC	**q01**	(679,571,222)	52	69285838	108	413218	111	(1870,444,248)	50	38027980	49	401631	161	(198,433,321)	47	13183341	35	376639	120	(1389,682,166)	52	57247797	125	341447	100
	q02	(860,598,172)	51	50753809	79	536903	122	(1951,610,282)	50	20272797	26	483981	45	(234,624,238)	48	23004441	61	262520	196	(1296,555,271)	52	58834409	129	341485	143
	q03	(775,430,214)	51	53782322	84	523430	151	(1789,606,231)	58	56681617	73	503369	159	(271,567,306)	47	17018313	45	306931	196	(1353,413,237)	41	61361440	134	338032	235
	q04	(738,527,261)	51	60502642	94	474102	64	(1849,458,241)	50	14446684	19	433512	161	(165,478,349)	48	9608060	25	307064	123	(1411,447,221)	52	51144840	112	291320	197
	q05	(721,379,233)	52	46675143	73	440921	205	(1910,468,242)	51	32157871	41	365080	123	(219,494,343)	46	10407507	27	299872	149	(1289,381,246)	51	57775303	126	343590	286
	q06	(675,515,246)	54	61576036	96	481354	123	(1846,371,236)	51	51592939	67	355585	237	(131,580,263)	45	2019485	5	304155	119	(1339,656,158)	51	48190453	105	281658	121
	q07	(824,390,249)	45	53885600	84	383588	188	(1975,507,223)	50	14591155	19	278957	87	(197,503,339)	48	14672601	39	329461	135	(1251,610,229)	59	51412740	113	320543	163
	q08	(764,568,106)	40	30148819	47	263479	157	(1961,615,249)	50	20795327	27	344766	33	(257,396,313)	58	25639175	68	324897	172	(1269,526,108)	52	47818450	105	275043	233
	q09	(850,645,259)	47	49979683	78	322181	99	(1851,492,241)	50	948741	1	355355	134	(236,550,355)	52	25609560	68	272355	187	(1417,532,241)	52	29426415	64	261918	110
	q10	(771,556,264)	50	49191601	77	425677	20	(1865,608,298)	46	18282487	24	295153	100	(281,485,281)	49	20209446	53	243155	169	(1412,514,139)	57	67399940	148	318364	172
	q11	(707,530,256)	51	55859826	87	411076	87	(1770,526,177)	50	898026	1	298621	198	(194,437,311)	48	9336103	25	246128	110	(1421,556,217)	52	40870010	89	322885	94
	q12	(790,632,271)	49	39004908	61	360366	60	(1926,561,226)	50	23121931	30	274242	37	(264,563,267)	50	22050595	58	257588	178	(1287,521,99)	52	44379753	97	321587	232
	q13	(685,661,171)	53	52243417	82	524807	159	(1892,494,243)	50	34506920	45	385899	107	(114,507,235)	46	8643051	23	267692	42	(1335,727,265)	52	43293544	95	281491	114
	q14	(1759,400,173)	82	70609210	110	236691	996	(824,552,233)	30	78997393	102	351952	1123	(282,377,248)	60	27081490	71	287670	186	(1831,897,289)	33	11328573	25	126376	495
	q15	(737,517,256)	42	0	0	271951	72	(1992,478,209)	39	0	0	232629	122	(216,531,361)	41	0	0	292400	171	(1389,620,198)	39	0	0	182927	64
FRPC	**q01**	(761,440,154)	90	17116232	27	1254043	173	(1825,566,235)	90	14930804	19	1697380	123	(227,611,233)	90	1967006	5	1327966	182	(1401,541,145)	90	610559	1	1178963	149
	q02	(733,443,253)	80	38675407	60	1114954	139	(1919,430,286)	80	33359867	43	1514692	164	(173,470,341)	80	106420	0	1179053	118	(1387,724,274)	80	23115111	51	1043416	88
	q03	(705,483,258)	70	14326810	22	978140	119	(1962,594,164)	70	139154	0	1318413	82	(194,536,364)	70	404548	1	1034060	164	(1401,494,211)	70	22048850	48	914873	154
	q04	(762,516,257)	60	109915	0	840879	61	(1794,525,215)	60	72222	0	1129529	166	(298,463,343)	60	38316	0	884737	209	(1343,658,260)	60	163590	0	783352	68
	q05	(835,514,245)	50	82364	0	701766	81	(1955,647,275)	50	50032	0	940713	69	(158,403,327)	50	29912	0	737422	117	(1288,593,253)	50	147476	0	652104	130
	q06	(742,540,267)	40	74086	0	562541	52	(1778,478,225)	40	38272	0	751685	201	(259,521,220)	40	22941	0	590710	152	(1408,521,226)	40	91953	0	520460	123
	q07	(702,619,238)	35	20878480	33	488979	95	(1991,484,211)	30	20246	0	562294	116	(286,373,250)	26	21098279	56	372080	191	(1309,675,248)	30	75428	0	390842	106
	q08	(771,526,261)	22	958221	2	308342	48	(1886,430,132)	20	11055	0	374568	201	(285,429,266)	18	16939391	45	260815	172	(1395,556,302)	20	65315	0	259134	98
	q09	(804,331,188)	11	186105	0	154238	254	(1909,511,232)	9	20981047	27	324471	84	(271,423,244)	10	410	0	148904	158	(1240,451,264)	10	27797238	61	182288	254
	q10	(831,608,257)	11	11262601	18	157481	60	(1893,520,227)	0	0	0	0	86	(293,401,302)	10	23160225	61	155522	197	(1367,582,151)	0	0	0	0	127

Table B.3: Results of Fuzzy C-Means algorithm

		Cluster 1						Cluster 2						Cluster 3						Cluster 4					
		C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC
Original		(1886,557,226)	0	0	0.00	0	0	(762,546,231)	0	0	0.00	0	0	(224,467,276)	0	0	0.00	0	0	(1353,553,216)	0	0	0.00	0	0
MPEG G-PCC	**q01**	(1887,561,228)	98	2976999	3.84	1851760	4	(763,543,232)	98	2127644	3.32	1369285	3	(224,477,276)	98	1991914	5.25	1445316	10	(1353,550,217)	98	2041902	4.47	1287160	3
	q02	(1887,560,227)	93	1446151	1.87	1749359	3	(763,544,232)	93	775314	1.21	1294015	2	(224,474,276)	93	979381	2.58	1365976	7	(1354,554,217)	93	968547	2.12	1215964	1
	q03	(1887,559,227)	84	851242	1.10	1584578	2	(762,544,231)	84	541899	0.85	1173093	2	(224,472,276)	84	590236	1.56	1234749	5	(1354,555,217)	84	233091	0.51	945872	3
	q04	(1887,559,227)	78	758759	0.98	1480410	2	(763,544,231)	79	576630	0.90	1096864	2	(224,472,276)	79	473647	1.25	1153188	5	(1354,555,216)	78	214327	0.47	531094	1
	q05	(1887,559,227)	72	480823	0.62	1362115	2	(763,545,232)	72	389787	0.61	1009601	1	(224,471,276)	72	293159	0.77	1062236	4	(1355,556,217)	72	72431	0.16	278269	1
	q06	(1886,558,227)	65	527281	0.68	1233076	1	(762,545,231)	65	434820	0.68	912708	2	(224,470,276)	65	288969	0.76	956782	3	(1354,555,216)	65	88098	0.19	217595	0
	q07	(1886,558,227)	58	504953	0.65	1093255	1	(762,545,231)	58	343702	0.54	806200	1	(224,470,276)	58	247639	0.65	845248	3	(1354,555,216)	58	385967	0.84	1027903	2
	q08	(1886,557,227)	49	428874	0.55	932881	0	(762,545,231)	50	331783	0.52	691785	1	(224,470,276)	49	231695	0.61	722603	3	(1354,555,216)	49	216104	0.47	861507	2
	q09	(1886,558,227)	41	343950	0.44	765102	1	(762,546,231)	41	216169	0.34	569136	0	(224,469,276)	41	210401	0.55	591930	2	(1354,554,217)	41	156898	0.34	647039	2
	q10	(1886,557,227)	21	212727	0.27	402244	0	(762,546,231)	21	128970	0.20	302342	1	(224,468,276)	21	110544	0.29	302867	1	(1354,554,216)	21	410249	0.90	1100734	2
	q11	(1886,557,227)	11	152827	0.20	232148	0	(762,546,231)	11	153854	0.24	160156	0	(224,468,276)	11	65223	0.17	209661	1	(1354,553,217)	11	264816	0.58	754745	2
	q12	(1886,557,226)	0	0	0.00	182241	0	(762,546,231)	0	0	0.00	113518	0	(224,467,276)	0	0	0.00	112654	0	(1353,553,216)	0	0	0.00	154469	0
MPEG V-PCC	**q01**	(1886,559,228)	50	26684139	34.41	457092	2	(763,548,233)	51	58258640	90.95	572202	3	(227,456,276)	48	18933531	49.93	377069	11	(1352,540,217)	52	46977497	102.83	250357	13
	q02	(1886,559,228)	50	22094379	28.49	483993	2	(763,548,233)	51	54904800	85.71	505446	3	(227,456,276)	48	18550413	48.92	262839	11	(1352,540,217)	53	44723471	97.90	280850	13
	q03	(1886,559,228)	50	23913546	30.84	513648	2	(763,548,233)	51	51277606	80.05	451413	3	(227,456,276)	48	17247335	45.49	307700	11	(1352,540,217)	52	40884169	89.49	282027	13
	q04	(1886,559,228)	50	20015139	25.81	454327	2	(763,548,233)	51	48311873	75.42	530289	3	(227,456,276)	48	16145675	42.58	263366	11	(1352,540,217)	52	40483633	88.61	261991	13
	q05	(1886,559,228)	50	18163784	23.43	365080	2	(763,548,233)	51	48487551	75.69	452524	3	(227,456,276)	48	15917658	41.98	336020	11	(1352,540,217)	52	41803463	91.50	318364	13
	q06	(1886,559,228)	50	10494224	13.53	446324	2	(763,548,233)	51	47212836	73.70	430722	3	(227,456,276)	48	15747486	41.53	289603	11	(1352,540,217)	52	41142393	90.06	197462	13
	q07	(1886,559,228)	50	9634132	12.43	355201	2	(763,548,233)	51	49615561	77.45	304473	3	(227,456,276)	48	15302308	40.36	321606	11	(1352,540,217)	52	41221227	90.23	319371	13
	q08	(1886,559,228)	50	7531506	9.71	357220	2	(763,548,233)	51	46905298	73.22	483937	3	(227,456,276)	48	15615344	41.18	311474	11	(1352,540,217)	52	41744663	91.38	322885	13
	q09	(1886,559,228)	50	948741	1.22	355355	2	(763,548,233)	51	47059336	73.46	512655	3	(227,456,276)	48	15135881	39.92	321581	11	(1352,540,217)	52	51888055	113.58	320361	13
	q10	(1886,559,228)	50	970343	1.25	355722	2	(763,548,233)	51	46484418	72.56	433261	3	(227,456,276)	48	14794253	39.02	258712	11	(1352,540,217)	52	53019062	116.05	291341	13
	q11	(1886,559,228)	50	898026	1.16	298621	2	(763,548,233)	51	46567383	72.69	434164	3	(227,456,276)	48	15146350	39.94	246178	11	(1352,540,217)	52	56943597	124.64	368020	13
	q12	(1886,559,228)	50	956431	1.23	250187	2	(763,548,233)	51	46916074	73.24	408586	3	(227,456,276)	48	14951596	39.43	302774	11	(1352,540,217)	52	61685689	135.02	341496	13
	q13	(1886,559,228)	50	945772	1.22	283254	2	(763,548,233)	51	46349420	72.35	444547	3	(227,456,276)	48	14929768	39.37	251607	11	(1352,540,217)	52	62129416	136.00	344840	13
	q14	(1886,559,228)	50	973840	1.26	282387	2	(763,548,233)	51	46756002	72.99	472871	3	(227,456,276)	48	15038497	39.66	325185	11	(1352,540,217)	52	40841250	89.40	212421	13
	q15	(1885,552,226)	39	0	0.00	232629	6	(763,549,231)	42	0	0.00	271951	3	(225,459,276)	41	0	0.00	292400	8	(1352,543,217)	39	0	0.00	182927	10
FRPC	**q01**	(1887,561,227)	90	339909	0.44	1697475	4	(762,545,231)	90	314370	0.49	1256030	2	(225,470,275)	90	225154	0.59	1325979	3	(1354,557,217)	90	443645	0.97	1178936	4
	q02	(1887,560,227)	80	183259	0.24	1507806	3	(762,544,231)	80	197730	0.31	1118252	2	(224,472,276)	80	106420	0.28	1179053	5	(1354,557,216)	80	266295	0.58	1046883	4
	q03	(1887,560,227)	70	138991	0.18	1318401	3	(762,545,231)	70	141963	0.22	979453	2	(224,472,276)	70	65203	0.17	1032747	5	(1354,557,217)	70	232254	0.51	914898	4
	q04	(1887,561,227)	60	72222	0.09	1129529	4	(762,545,231)	60	109915	0.17	840879	1	(224,472,276)	60	38316	0.10	884737	5	(1354,556,217)	60	163590	0.36	783352	4
	q05	(1887,560,227)	50	50032	0.06	940713	3	(762,545,231)	50	82364	0.13	701766	1	(224,472,276)	50	29912	0.08	737422	5	(1354,556,216)	50	147476	0.32	652104	3
	q06	(1887,560,227)	40	38272	0.05	751685	3	(762,545,231)	40	74086	0.12	562541	1	(224,471,276)	40	22941	0.06	590710	4	(1354,556,216)	40	91953	0.20	520460	4
	q07	(1887,560,227)	30	20246	0.03	562294	3	(762,545,231)	30	34314	0.05	423395	1	(224,471,276)	30	7777	0.02	443575	4	(1354,556,216)	30	75428	0.17	390842	3
	q08	(1887,559,227)	20	11055	0.01	374568	2	(762,545,231)	20	11934	0.02	283946	1	(224,470,276)	20	4494	0.01	296128	3	(1354,555,216)	20	65315	0.14	259134	2
	q09	(1887,559,227)	10	2193	0.00	186630	2	(762,546,231)	10	2672	0.00	143750	1	(224,469,276)	10	410	0.00	148904	2	(1354,555,216)	10	32844	0.07	127473	2
	q10	(1886,557,226)	0	0	0.00	0	0	(762,546,231)	0	0	0.00	0	0	(224,467,276)	0	0	0.00	0	0	(1353,553,216)	0	0	0.00	0	0

Table B.4: Results of the Mean Shift algorithm

		Cluster 1						Cluster 2						Cluster 3						Cluster 4						Cluster 5					
		C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC
Original		(771,614,232)	0	0	0.00	0	0	(204,544,277)	0	0	0.00	0	0	(1895,676,232)	0	0	0.00	0	0	(1335,587,211)	0	0	0.00	0	0	0	0	0	0	0	0
MPEG G-PCC	**q01**	(772,613,233)	98	1970375	3.08	1369342	1	(205,559,278)	98	11460330	30.22	1445020	14	(1896,679,234)	98	2976999	3.84	1851760	4	(1335,583,212)	98	2041902	4.47	1287160	4	0	0	0	0	0	0
	q02	(771,615,233)	93	775314	1.21	1294015	1	(205,552,278)	93	979381	2.58	1365976	7	(1896,680,233)	93	1446151	1.87	1749359	4	(1335,588,211)	93	968547	2.12	1215964	1	0	0	0	0	0	0
	q03	(771,613,232)	84	541899	0.85	1173093	1	(204,549,277)	84	590236	1.56	1234749	4	(1896,678,233)	84	851242	1.10	1584578	3	(1335,588,211)	84	410249	0.90	1100734	2	0	0	0	0	0	0
	q04	(771,613,232)	79	576630	0.90	1096864	0	(204,549,277)	79	473647	1.25	1153188	5	(1896,678,233)	78	758759	0.98	1480410	2	(1335,588,211)	78	385967	0.84	1027903	2	0	0	0	0	0	0
	q05	(771,614,232)	72	355192	0.55	1010204	1	(205,548,278)	72	9413632	24.83	1059057	3	(1896,678,233)	72	480823	0.62	1362115	3	(1336,590,211)	72	233091	0.51	945872	3	0	0	0	0	0	0
	q06	(770,613,232)	65	434820	0.68	912708	0	(204,547,277)	65	288969	0.76	956782	2	(1896,678,232)	65	527281	0.68	1233076	2	(1335,589,210)	65	216104	0.47	861507	2	0	0	0	0	0	0
	q07	(771,613,232)	58	343702	0.54	806200	0	(204,548,277)	58	247639	0.65	845248	3	(1895,677,232)	58	504953	0.65	1093255	1	(1336,590,210)	58	264816	0.58	754745	3	0	0	0	0	0	0
	q08	(771,613,232)	50	331783	0.52	691785	0	(204,546,277)	49	231695	0.61	722603	2	(1895,676,232)	49	428874	0.55	932881	0	(1335,587,211)	49	156898	0.34	647039	1	0	0	0	0	0	0
	q09	(771,614,232)	41	216169	0.34	569136	1	(204,547,277)	41	210401	0.55	591930	2	(1895,677,232)	41	343950	0.44	765102	1	(1335,588,211)	41	214327	0.47	531094	1	0	0	0	0	0	0
	q10	(771,613,232)	21	128970	0.20	302342	0	(204,545,277)	21	110544	0.29	302867	1	(1895,675,232)	21	212727	0.27	402244	1	(1335,588,211)	21	72431	0.16	278269	1	0	0	0	0	0	0
	q11	(771,614,232)	11	153854	0.24	160156	0	(204,545,277)	11	65223	0.17	209661	1	(1895,676,232)	11	152827	0.20	232148	0	(1335,588,211)	11	88098	0.19	217595	1	0	0	0	0	0	0
	q12	(771,613,232)	0	0	0.00	113518	0	(204,543,277)	0	0	0.00	112654	1	(1895,674,231)	0	0	0.00	182241	2	(1335,586,211)	0	0	0.00	154469	1	0	0	0	0	0	0
MPEG V-PCC	**q01**	(771,616,234)	51	81184473	126.73	513850	3	(204,540,277)	29	16240561	42.83	432421	5	(1895,678,233)	50	29435194	37.96	457084	2	(1334,581,211)	52	57887335	126.71	344816	6	(209,383,280)	0	0	0.00	0	0
	q02	(771,615,234)	51	59085171	92.24	599804	3	(204,541,277)	8	6878743	18.14	176685	3	(1895,678,233)	50	28786937	37.13	484013	2	(1334,582,211)	53	61484349	134.58	281785	5	(209,382,280)	0	0	0.00	0	0
	q03	(771,616,234)	51	59300398	92.57	514406	3	(204,541,277)	11	1585096	4.18	190186	3	(1895,678,233)	50	30248111	39.01	487542	2	(1334,582,211)	52	52294949	114.47	271874	5	(209,383,280)	0	0	0.00	0	0
	q04	(771,615,234)	51	56747418	88.59	426489	3	(204,541,277)	11	2276917	6.00	199394	3	(1895,678,233)	50	27009674	34.83	454289	2	(1334,582,211)	52	52100550	114.04	325767	5	(209,382,280)	0	0	0.00	0	0
	q05	(771,615,234)	51	62636503	97.78	524779	3	(204,541,277)	10	9454189	24.93	191158	3	(1895,678,233)	50	31796664	41.01	362959	2	(1334,581,211)	52	46132792	100.98	345246	5	(209,382,280)	0	0	0.00	0	0
	q06	(771,615,234)	51	51916421	81.04	514486	3	(204,541,277)	41	19855775	52.36	613579	3	(1895,677,233)	50	16439204	21.20	425270	2	(1334,582,211)	52	43965167	96.24	275622	5	(209,382,280)	0	0	0.00	0	0
	q07	(771,615,234)	51	56801104	88.67	537322	3	(204,541,277)	55	23302265	61.45	805550	3	(1895,678,233)	50	4587181	5.92	353803	2	(1334,582,211)	52	45196025	98.93	281865	5	(209,383,280)	0	0	0.00	0	0
	q08	(771,615,234)	51	47306198	73.85	360755	3	(204,541,277)	12	2740225	7.23	168269	3	(1895,678,233)	50	7531506	9.71	357220	2	(1334,582,211)	52	48633470	106.45	331476	5	(209,382,280)	0	0	0.00	0	0
	q09	(771,615,234)	51	43466683	67.85	452797	3	(204,542,277)	17	12727042	33.56	256333	3	(1895,678,233)	50	948741	1.22	355355	2	(1334,582,211)	52	51549035	112.84	318367	5	(209,382,280)	0	0	0.00	0	0
	q10	(771,615,234)	51	48068739	75.04	418553	3	(204,542,277)	55	23222096	61.24	805670	3	(1895,678,233)	50	970343	1.25	355722	2	(1334,582,211)	52	44710265	97.87	282414	5	(209,383,280)	0	0	0.00	0	0
	q11	(771,615,234)	51	61076473	95.34	472357	3	(204,542,277)	28	16915350	44.61	420002	3	(1895,678,233)	50	898026	1.16	298621	2	(1334,582,211)	52	42172626	92.31	325509	5	(209,382,280)	0	0	0.00	0	0
	q12	(771,615,234)	51	51606726	80.56	454691	3	(204,542,277)	11	3180145	8.39	168529	3	(1895,678,233)	50	956431	1.23	250187	2	(1334,582,211)	52	42917647	93.94	281767	5	(209,382,280)	0	0	0.00	0	0
	q13	(771,615,234)	51	55250272	86.25	535488	3	(204,542,277)	18	13432631	35.43	271173	3	(1895,678,233)	50	945772	1.22	283254	2	(1334,582,211)	52	41273671	90.34	197766	5	(209,382,280)	0	0	0.00	0	0
	q14	(771,615,234)	53	61052360	95.31	530656	3	(204,542,277)	12	2769802	7.30	168672	3	(1895,678,233)	50	973840	1.26	282387	2	(1334,582,211)	52	42558712	93.16	212398	5	(209,382,280)	0	0	0.00	0	0
	q15	(771,615,232)	42	0	0.00	271951	2	(206,465,279)	41	0	0.00	292400	80	(1893,663,228)	39	0	0.00	232629	13	(1333,579,211)	39	0	0.00	182927	8	0	0	0	0	0	0
FRPC	**q01**	(770,612,231)	90	314370	0.49	1256030	2	(205,547,277)	90	225154	0.59	1325979	2	(1896,678,233)	90	339909	0.44	1697475	3	(1335,589,211)	90	443645	0.97	1178936	3	0	0	0	0	0	0
	q02	(770,612,232)	80	197730	0.31	1118252	2	(205,549,277)	80	106420	0.28	1179053	4	(1896,679,233)	80	183259	0.24	1507806	4	(1335,590,211)	80	266295	0.58	1046883	3	0	0	0	0	0	0
	q03	(770,613,232)	70	141963	0.22	979453	1	(204,548,277)	70	65203	0.17	1032747	4	(1896,679,233)	70	138991	0.18	1318401	3	(1335,589,211)	70	232254	0.51	914898	3	0	0	0	0	0	0
	q04	(771,613,232)	60	109915	0.17	840879	0	(204,549,277)	60	38316	0.10	884737	4	(1896,679,233)	60	72222	0.09	1129529	4	(1335,589,211)	60	163590	0.36	783352	2	0	0	0	0	0	0
	q05	(771,613,232)	50	15758960	24.60	697898	1	(204,549,277)	50	739942	1.95	741289	4	(1896,679,233)	50	50032	0.06	940713	4	(1335,589,211)	50	147476	0.32	652104	3	0	0	0	0	0	0
	q06	(770,613,232)	40	74086	0.12	562541	1	(204,548,277)	40	22941	0.06	590710	4	(1896,679,233)	40	38272	0.05	751685	4	(1335,590,211)	40	91953	0.20	520460	3	0	0	0	0	0	0
	q07	(770,613,232)	30	34314	0.05	423395	1	(204,548,277)	30	7777	0.02	443575	4	(1896,679,233)	30	20246	0.03	562294	3	(1335,589,211)	30	75428	0.17	390842	3	0	0	0	0	0	0
	q08	(771,613,232)	20	11934	0.02	283946	0	(204,547,277)	20	4494	0.01	296128	3	(1896,678,233)	20	11055	0.01	374568	3	(1335,589,211)	20	65315	0.14	259134	2	0	0	0	0	0	0
	q09	(770,613,232)	10	2672	0.00	143750	0	(204,547,277)	10	410	0.00	148904	2	(1895,677,233)	10	2193	0.00	186630	2	(1335,588,211)	10	32844	0.07	127473	2	0	0	0	0	0	0
	q10	(771,614,232)	0	0	0.00	0	0	(204,544,277)	0	0	0.00	0	0	(1895,674,232)	0	0	0.00	0	1	(1335,587,211)	0	0	0.00	0	0	0	0	0	0	0	0

Table B.5: Results of the Pcsegdist

		Cluster 1						Cluster 2						Cluster 3						Cluster 4					
		C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC	C	PNP	AVD	PAVD	DCH	DC
Original		(211,486,276)	0	0	0	0	0	(763,510,234)	0	0	0	0	0	(1339,567,219)	0	0	0	0	0	(1894,524,226)	0	0	0	0	0
MPEG GPCC	**q01**	(212,492,276)	98	1991914	5	1445316	5	(764,511,234)	98	2127644	3	1369285	2	(1339,566,220)	98	2041902	4	1287160	1	(1895,526,227)	98	2976999	4	1851760	3
	q02	(212,490,276)	93	979381	3	1365976	4	(764,511,234)	93	775314	1	1294015	2	(1339,568,220)	93	968547	2	1215964	1	(1895,526,226)	93	1446151	2	1749359	2
	q03	(211,489,276)	84	590236	2	1234749	3	(764,510,234)	84	541899	1	1173093	1	(1339,568,220)	84	410249	1	1100734	2	(1894,526,226)	84	851242	1	1584578	2
	q04	(211,489,276)	79	473647	1	1153188	3	(764,510,234)	79	576630	1	1096864	1	(1339,568,219)	78	385967	1	1027903	1	(1894,525,226)	78	758759	1	1480410	1
	q05	(212,489,276)	72	293159	1	1062236	3	(764,511,234)	72	389787	1	1009601	1	(1340,569,220)	72	233091	1	945872	2	(1895,526,226)	72	480823	1	1362115	2
	q06	(211,488,276)	65	288969	1	956782	2	(764,510,234)	65	434820	1	912708	1	(1339,568,219)	65	216104	0	861507	1	(1894,525,226)	65	527281	1	1233076	1
	q07	(211,488,276)	58	247639	1	845248	2	(764,510,234)	58	343702	1	806200	1	(1339,568,219)	58	264816	1	754745	1	(1894,525,226)	58	504953	1	1093255	1
	q08	(211,488,276)	49	231695	1	722603	1	(763,510,234)	50	331783	1	691785	0	(1339,568,219)	49	156898	0	647039	1	(1894,525,226)	49	428874	1	932881	1
	q09	(211,488,276)	41	210401	1	591930	1	(764,510,234)	41	216169	0	569136	1	(1339,568,220)	41	214327	0	531094	1	(1894,525,226)	41	343950	0	765102	1
	q10	(211,487,276)	21	110544	0	302867	1	(764,510,234)	21	128970	0	302342	0	(1339,567,220)	21	72431	0	278269	0	(1894,525,226)	21	212727	0	402244	0
	q11	(211,487,276)	11	65223	0	209661	0	(763,510,234)	11	153854	0	160156	0	(1339,567,220)	11	88098	0	217595	0	(1894,524,226)	11	152827	0	232148	0
	q12	(211,486,276)	0	0	0	112654	0	(763,510,234)	0	0	0	113518	0	(1339,567,219)	0	0	0	154469	0	(1894,524,226)	0	0	0	182241	0
MPEG VPCC	**q01**	(1107,516,239)	514	582347313	1536	459493	897	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	q02	(1107,516,239)	514	577474179	1523	345941	897	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	q03	(1107,516,239)	514	570991222	1506	414716	897	(1576,590,143)	100	64056582	100	1394727	822	0	0	0	0	0	0	0	0	0	0	0	0
	q04	(1106,516,239)	514	567634229	1497	414811	897	(1582,587,144)	100	64059022	100	1394760	827	0	0	0	0	0	0	0	0	0	0	0	0
	q05	(753,513,244)	324	319220571	842	391433	544	(1894,522,227)	102	15126004	24	361749	1131	0	0	0	0	0	0	0	0	0	0	0	0
	q06	(753,513,244)	324	313942642	828	356665	544	(1894,522,227)	102	14600700	23	396597	1131	(1550,587,144)	100	45684946	100	1311205	226	0	0	0	0	0	0
	q07	(1106,516,239)	514	563770119	1487	399135	896	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	q08	(753,513,244)	324	305033386	804	372412	544	(1894,522,227)	102	14473122	23	355211	1131	0	0	0	0	0	0	0	0	0	0	0	0
	q09	(753,513,244)	324	302494862	798	372809	544	(1894,522,227)	102	14427707	23	353017	1131	0	0	0	0	0	0	0	0	0	0	0	0
	q10	(753,513,244)	324	303488504	800	360180	544	(1894,522,227)	102	14449309	23	353876	1131	0	0	0	0	0	0	0	0	0	0	0	0
	q11	(753,513,244)	324	303625725	801	354727	544	(1894,522,227)	102	14376991	22	278278	1131	0	0	0	0	0	0	0	0	0	0	0	0
	q12	(753,513,244)	324	303254813	800	358497	544	(1894,522,227)	102	14435397	23	257686	1131	0	0	0	0	0	0	0	0	0	0	0	0
	q13	(753,513,244)	324	302890223	799	358862	544	(1894,522,227)	102	14424737	23	257687	1131	0	0	0	0	0	0	0	0	0	0	0	0
	q14	(753,513,244)	324	302904791	799	364830	544	(1894,522,227)	102	14452806	23	257686	1131	0	0	0	0	0	0	0	0	0	0	0	0
	q15	(211,482,275)	41	0	0	292400	5	(764,507,234)	42	0	0	271951	3	(1337,561,220)	39	0	0	182927	6	(1893,519,225)	39	0	0	232629	6
FRPC	**q01**	(212,489,275)	90	225154	1	1325979	2	(763,512,233)	90	314370	0	1256030	2	(1339,569,220)	90	443645	1	1178936	3	(1895,528,226)	90	339909	0	1697475	4
	q02	(211,489,276)	80	106420	0	1179053	3	(763,512,233)	80	197730	0	1118252	2	(1339,569,219)	80	266295	1	1046883	2	(1895,527,226)	80	183259	0	1507806	3
	q03	(211,490,276)	70	65203	0	1032747	3	(763,512,233)	70	141963	0	979453	2	(1339,569,220)	70	232254	1	914898	3	(1894,527,226)	70	138991	0	1318401	3
	q04	(211,489,276)	60	38316	0	884737	3	(764,512,233)	60	109915	0	840879	2	(1339,569,220)	60	163590	0	783352	2	(1894,527,226)	60	72222	0	1129529	3
	q05	(211,489,276)	50	29912	0	737422	3	(764,512,234)	50	82364	0	701766	2	(1339,569,220)	50	147476	0	652104	2	(1894,527,226)	50	50032	0	940713	3
	q06	(211,489,276)	40	22941	0	590710	3	(764,512,233)	40	74086	0	562541	2	(1339,569,219)	40	91953	0	520460	2	(1894,527,226)	40	38272	0	751685	3
	q07	(211,489,276)	30	7777	0	443575	2	(764,511,233)	30	34314	0	423395	2	(1339,569,219)	30	75428	0	390842	2	(1894,527,226)	30	20246	0	562294	2
	q08	(211,488,276)	20	4494	0	296128	2	(764,511,234)	20	11934	0	283946	2	(1339,568,219)	20	65315	0	259134	1	(1894,526,226)	20	11055	0	374568	2
	q09	(211,488,276)	10	410	0	148904	1	(763,511,234)	10	2672	0	143750	1	(1339,568,219)	10	32844	0	127473	1	(1894,526,226)	10	2193	0	186630	2
	q10	(211,486,276)	0	0	0	0	0	(763,510,234)	0	0	0	0	0	(1339,567,219)	0	0	0	0	0	(1894,524,226)	0	0	0	0	0