COPPE
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
UFRJ

COMPARATIVE PERFORMANCE OF HEVC-BASED CODECS FOR STATIC LINEARLY-ARRANGED LIGHT FIELDS

Luiz Gustavo Cardoso Tavares

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Eduardo Antônio Barros da Silva
Fernando Manuel Bernardo Pereira

Rio de Janeiro
Setembro de 2016

COMPARATIVE PERFORMANCE OF HEVC-BASED CODECS FOR STATIC LINEARLY-ARRANGED LIGHT FIELDS

Luiz Gustavo Cardoso Tavares

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Prof. Lisandro Lovisolo, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2016

Tavares, Luiz Gustavo Cardoso

Comparative Performance of HEVC-based codecs for Static Linearly-Arranged Light Fields / Luiz Gustavo Cardoso Tavares. – Rio de Janeiro: UFRJ/COPPE, 2016.

XV, 82 p.: il.; 29, 7cm.

Orientadores: Eduardo Antônio Barros da Silva
Fernando Manuel Bernardo Pereira

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2016.

Referências: p. 79 – 82.

1. Image Compression. 2. Multiview Coding. 3. Light Field. I. Silva, Eduardo Antônio Barros da *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

À minha família e amigos.

Agradecimentos

Gostaria de agradecer a Deus por guiar em mais esta etapa e por me dar forças para superar os momentos difíceis que nela apareceram.

A minha família, agradeço por todo o apoio, paciência e compreensão durante esses anos de estudo. Sem esse apoio, não teria sido possível alcançar nenhum dos objetivos que almejei.

Agradeço ao professor Eduardo pela confiança e por toda a paciência. Sua orientação e conselhos ao longo desses anos foi muito importante para o meu aprendizado e crescimento profissional. Agradeço ao professor Fernando Pereira pelas orientações e sugestões ao trabalho.

A todos os professores, amigos e colegas do SMT em especial a Felipe Ribeiro, Felipe Barboza, Jonathan, Tadeu, Renam, Gabriel, Markus e Anderson. Um muito obrigado ao José Fernando por todo o auxílio prestado. Agradeço também em especial ao suporte do amigo Luís Lucas, sem o qual este trabalho não teria êxito. Foi um grande prazer poder trabalhar e contar com vocês.

Agradeço também aos amigos e colegas do CTEx pelo apoio durante o término desse trabalho. Meus agradecimentos também aos meus amigos Yoann e Leonardo por todo o incentivo em mais esta etapa. Agradeço por fim ao leitor desta obra, que ao lê-la prestigia todo o trabalho realizado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ESTUDO COMPARATIVO DE DESEMPENHO DE *CODECS* SIMILARES AO HEVC PARA CAMPOS DE LUZ LINEARMENTE ARRANJADOS

Luiz Gustavo Cardoso Tavares

Setembro/2016

Orientadores: Eduardo Antônio Barros da Silva
Fernando Manuel Bernardo Pereira

Programa: Engenharia Elétrica

Nos últimos anos, a importância das imagens por campos de luz tornaram-se destaque na comunidade científica dadas as suas numerosas e facinantes aplicações. Um campo de luz é uma forma de representar informações luminosas provenientes de qualquer direção, em qualquer momento. Normalmente, dada a enorme quantidade de dados produzida, em muitos casos trabalha-se com uma versão reduzida dos campos de luz: as imagens multivistas. Estas são adquiridas através de um conjunto de câmeras calibradas.

O principal propósito deste trabalho é investigar o quanto as ferramentas atuais em compressão de imagens conseguem explorar a redundância entre-vistas deste conteúdo a fim de reduzir a quantidade de dados gerada codificador. Este trabalho também estuda como as técnicas padrões de compressão agem em diferentes tipos de seqüência multivista, real ou sintética, densa ou esparsa. Cada uma das ferramentas é avaliada de acordo com o seu desempenho taxa-distorção a fim de encontrar a melhor solução para cada tipo diferente seqüências multivistas.

Resultados experimentais mostram que o desempenho de compressão é altamente dependente de como a estrutura de predição entre-vistas se aproveita da densidade de vistas. Além disso, resultados mostram que quando a densidade angular de vistas é muito alta, o modo mais eficiente de explorar a redundância entre-vistas é descartar pontos de vista no codificador e utilizar interpolação de vistas no decodificador. Esse método, quando aplicado a seqüências multivistas com alta densidade angular supera a aplicação direta de perfis *inter* do HEVC. Isso sugere que as ferramentas de codificação do HEVC não são capazes de explorar a alta redundância inerente a campos de luz com alta densidade angular.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

COMPARATIVE PERFORMANCE OF HEVC-BASED CODECS FOR STATIC LINEARLY-ARRANGED LIGHT FIELDS

Luiz Gustavo Cardoso Tavares

September/2016

Advisors: Eduardo Antônio Barros da Silva
Fernando Manuel Bernardo Pereira

Department: Electrical Engineering

In the last years, the importance of light field imaging has grown together with its numerous and fascinating applications. A light field is a form of representing the light coming for every direction at any time. However, the scientific community, given the large amount of generated data, in many cases works with a simplification of light fields, called multiview imaging, in which images are acquired through a set of calibrated cameras.

The main purpose of this work is to investigate how well the current image compression techniques can exploit the natural inter-view redundancy of such content in order to reduce the total amount of data generated by the encoder. This work also studies how standard compression techniques act in different types of multiview sequences, real or synthetic, dense or sparse. Each of the tested compression methods was evaluated by its rate-distortion performance in order to find out the best coding solution for compressing different multiview sequences.

Experimental results show that the compression performance is highly dependent on how the inter-view prediction structure takes advantage of the sequence view density. In addition, our results point to the fact that, when the angular density of views is very high, the most efficient way to exploit their inter-view redundancy is by discarding viewpoints at the encoding, together with view interpolation at the decoder. Such method, when applied to multiview sequences with high density of views outperforms the direct application of HEVC inter profiles. This suggests that the HEVC coding tools are not able to fully exploit the high redundancy inherent to light fields with high angular density.

Contents

List of Figures

List of Tables

Lists of Abbreviations

3D-TV	3D Television, p. 1
FTV	Free-viewpoint Television, p. 1
PBRT	Physically Based Rendering, p.11
HEVC	High Efficiency Video Coding, p. 22
MPEG	Moving Picture Experts Group, p. 22
VCEG	Video Coding Experts Group, p. 22
H.264/AVC	Advanced Video Coding, p. 22
CTU	Coding Tree Unit, p. 22
CTB	Coding Tree Block, p. 22
CU	Coding Unit, p. 22
CB	Coding Block, p. 22
PU	Prediction Unit, p. 22
PB	Prediction Block, p. 22
TU	Transform Unit, p. 22
TB	Transform Block, p. 22
MV	Motion Vector, p. 23
DST	Discrete Sine Transform, p. 24
DCT	Discrete Cosine Transform, p. 24
QP	Quantization Parameter, p.30
SAO	Sample Adaptive Offset, p.24
CABAC	Context Adaptive Binary Arithmetic Coding, p.24
MV-HEVC	Multiview extension for HEVC, p.26
3D-HEVC	3D extension for HEVC, p.26
GOV	Group of Views, p. 26
GOP	Group of Pictures, p. 26
PSNR	Peak Signal-to-Noise Ratio, p.37
MSE	Mean Squared Error, p.37
RD	Rate-distortion, p.37
BD	Bjøntegaard delta, p.37
IST-MCFI	Motion Compensation Frame Interpolation from Instituto Superior Técnico, p.35

Chapter 1

Introduction

In recent years, there has been an increasing need to reproduce more accurately the human vision through multimedia contents. Nowadays, new improvements in visual contents are being developed in order to improve the human visual experience.

For example, photography can now acquire electromagnetic information from other frequencies than the visible light. It can also register from tiny objects as molecules to enormous structures as galaxies.

Furthermore, the current enhancements on video technology and computational power have made possible the use of more sophisticated acquisition methods that provide a more realistic experience to the users. The necessity of emulating the human vision led to the development of the 3D video technology, which adds depth perception to videos.

Another recent innovation made to enhance the video experience is to add the possibility of a viewpoint change. With this technology, the user can watch the same content from whichever position in the scene he/she wants.

1.1 Motivation

If multiple viewpoints in the same scene are necessary, several cameras must be disposed around the scene to obtain the desired content. The resulting set of visual information is called a multiview sequence.

Among the many applications of multiview sequences, there are two important applications currently being studied and developed by the scientific community: the 3D Television (3D-TV) [3] and the Free-viewpoint Television (FTV)[4].

The 3D-TV main focus is to provide an immersion into the contents by delivering to the user depth information. Using at least two cameras to simulate the stereoscopic human vision, depth is calculated and presented to the user through specific technologies on displays and glasses.

Using dozens of views more views than 3D-TV, Free-viewpoint Television has as its main purpose to provide an environment where the users can freely choose the position where they want to watch the video content. However, it may require a very large number of cameras. In order to avoid this issue, the desired content may be generated by a so called virtual camera: a model of camera created by geometric combinations of other cameras that can simulate the video content as if a physical camera was really there.

Other important applications of multiview imaging are:

1. **Scene segmentation and composition:** Multiple views of the same scene can help to introduce or remove objects from a scene as the object dimensions can be calculated and occlusions issues can be solved;

2. **Surveillance:** Detection and tracking of abnormalities in a scene are easier with multiple cameras, solving for example, problems as occlusions;

3. **Light field cameras:** Introducing arrays of microlenses in a camera, the user can acquire individual light rays from a scene. By selectively combining these, different images can be rendered (e.g. multiple viewpoints or multiple focal distances);

4. **Cinematography:** Multiple views can aid in creating some scene effects such as the Bullet Effect, seen in The Matrix (1999) and the Dolly Zoom Effect (change in the cameras parameters, such as field of view, giving the zoom effect) as seen in Vertigo (1958).

1.2 Objective

The main objective of this work is to study how current technology is capable of exploiting the natural redundancy present in multiview sequences. This redundancy is present between the views of such a sequence, and it is related to the proximity of the cameras. The closer are two adjacent cameras, the largest will be the inter-view redundancy.

This works proposes to study different HEVC coding structures, coding parameters and redundancy removal techniques in order to effectively exploit such inter-view redundancy. It is also crucial to analyse how much these techniques are able to remove the inter-view redundancy when compressing different types of multiview sequences.

1.3 Organization

This work is structured as follows: Chapter 2 presents a mathematical model for multidimensional imaging focusing in the multiview approach as well as the description of the material used in the compression experiments. Chapter 3 makes experiment proposals with the objective to understand the inter-view redundancy and different manners to reduce it. Chapter 4 describes all the methods, parameters and tools used in this work to compress multiview sequences as well as to compare results. Chapter 5 analyses the results of the proposed experiments and compare the compression approaches. Chapter 6 presents the conclusions of the experiments and what can be learned from them.

Chapter 2

Multiview imaging

This chapter will introduce some basic concepts behind multiview imaging. For its proper understanding, some background concepts will be presented concerning the plenoptic function and multiview acquisition.

2.1 The plenoptic function and multiview sequences

The plenoptic function, first introduced by Adelson in [5], is a multidimensional function which describes, in a given point in space, all luminous information from the surrounding environment. In other words, this function can describe exactly the light spectrum that comes from every direction at any time instant.

A plenoptic function can be mathematically explained considering a point located in (O_x,O_y,O_z) which can register an intensity of light varying with time $I(t)$. On can also represent the direction from where the light comes by the vector in spherical coordinates $\vec{d} = (\theta, \phi)$ and the light spectrum $S(\lambda)$, where λ is the wavelength. A representation of the plenoptic function is depicted in Figure 2.1.

In a concise form, the plenoptic function at a point can be represented considering all listed variables and functions and can be written as

$$I = f(O_x, O_y, O_z, \theta, \phi, \lambda, t). \tag{2.1}$$

A plenoptic function, as seen in Equation 2.1, would generate an enormous amount of data to be processed or stored. Fortunately, as the human visual system is limited, and it is not necessary to store all luminous data described by $I(O_x, O_y, O_z, \theta, \phi, \lambda, t)$.

First of all, the human eye has only three types of cells responsible for color vision [6]. These cells are called cones and cannot distinguish the full spectral density of light, but just the combination of three spectral responses.

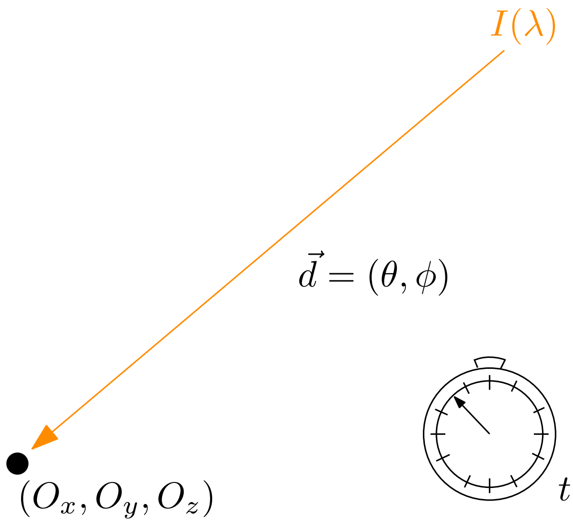


Figure 2.1: *Visual representation of a plenoptic function in the Equation 2.1 form.*

In addition, the total number of eye cells is finite, meaning that humans cannot perceive luminous information from every direction [7]. Correspondingly, modern cameras store the data which only stimulates a finite number of light sensitive elements, called *pixels*.

It also is not possible to arrange an infinite number of cameras in a limited spatial volume, restricting the choices of (O_x,O_y,O_z). Instead, one should use a limited number of cameras in this volume.

In order to further simplify the plenoptic function $I(O_x, O_y, O_z, \theta, \phi, \lambda, t)$ one should consider how the light is captured by the camera. One of the possibilities is the pinhole camera model, presented in Figure 2.2.

This camera model has a pinhole located in (C_x, C_y, C_z) (also called camera centre) where the light rays should pass through. It also has a plane called *camera plane* that registers all the luminous information from the outside of the camera and that passes through the pinhole. It is considered here that the image has $M \times N$ pixels, and that a pixel is square with unit side. Each pixel can be represented by a pair of values (u, v) indicating its location on the image plane, with $0 \leq u \leq M - 1$, $0 \leq v \leq N - 1$.

Be an instantaneous monochromatic light ray with intensity $L(x, y, \gamma)$ that intercepts the camera plane on the point (x, y) having travelled a distance *Gamma* from the light source (or from a reflexive object) to the centre of the camera. Then, the total luminous intensity received by a pixel (u, v) is

$$I(u, v) = \int_{u}^{u+1} \int_{v}^{v+1} \int_{0}^{\Gamma} L(x, y, \gamma) \, du \, dv \, d\gamma, \tag{2.2}$$

where Γ is the maximum distance from the camera centre to the reflexive opaque object or light source.

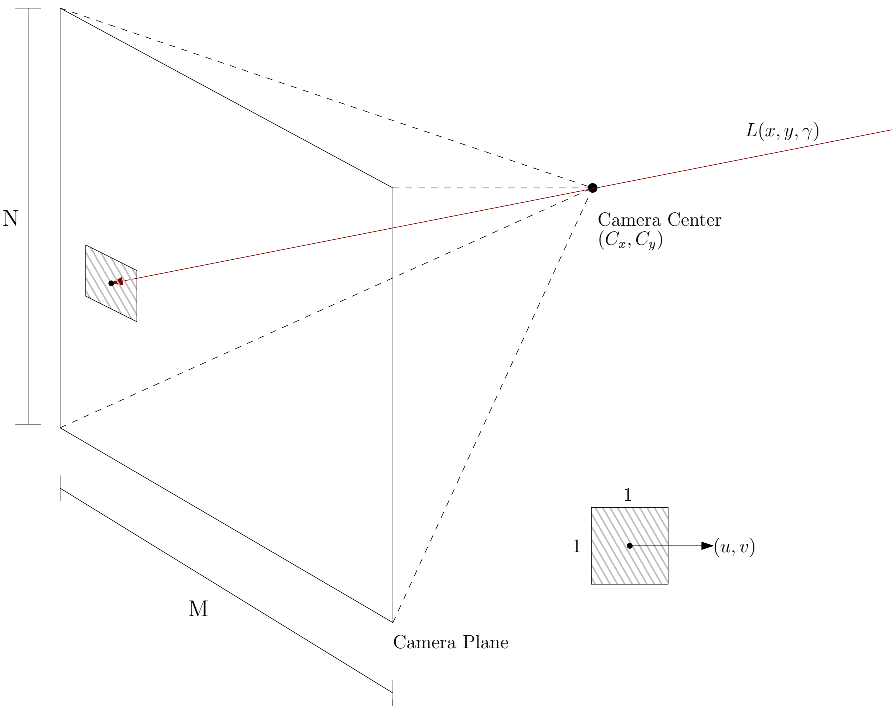


Figure 2.2: *Pinhole camera model and its pixel representation.*

Note that the pinhole restricts the light that intercepts the camera plane. Therefore, the maximum angle Θ_{max} in front of the camera with respect to the vector normal to the camera plane is called *field of view.* One can see the field of view represented in Figure 2.3 for the the x-axis. Θ_{max} can be written as

$$\Theta_{max} = 2 \arctan\left(\frac{M}{2f}\right), \tag{2.3}$$

where f, the distance between the focal plane and the camera centre is called *focal length.* Analogously, the maximum vertical angle Φ_{max} is

$$\Phi_{max} = 2 \arctan\left(\frac{N}{2f}\right). \tag{2.4}$$

Considering that the light varies in time, then Equation 2.2 can be rewritten as

$$I(u, v, t_0) = \int_u^{u+1} \int_v^{v+1} \int_0^{\Gamma} \int_t L(x, y, \gamma, t) W_{t_0}(t)\, du\, dv\, d\gamma\, dt. \tag{2.5}$$

$W_{t_0}(t)$ in Equation 2.2 refers to the exposure window from the moment t_0. The function $W(t)$ depends on the shutter response and the material which the film is made of. The time interval when the shutter is kept open is called *exposure time.*

Equation 2.5 can be extended to a video sequence, which starts at a time instant agreed to be zero in both sequences and captures K frames at a rate of R frames per second. This extension can be seen in Equation 2.6.

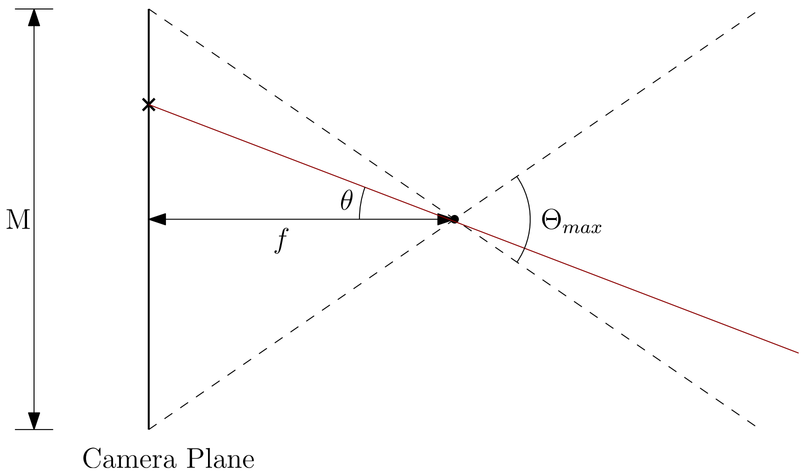


Figure 2.3: *Upper view of a pinhole camera model and its focal length f and field of view Θ_{max}.*

$$I(u,v,n) = \int_u^{u+1}\int_v^{v+1}\int_0^{\Gamma}\int_t L(x,y,\gamma,t)W_{nR}(t)\,du\,dv\,d\gamma\,dt, \tag{2.6}$$

being n in this equation called frame index.

One can notice that Equation 2.6 is only valid for monochromatic light rays. However, usually the light ray carries different wavelengths λ, with intensity of $L(x,y,\gamma,t,\lambda)$.

An image which has multiple wavelengths cannot be described in the $I(u,v)$ format. A common solution, derived from the fact that human beings have three cones, is to represent the visible spectrum of colours in a three-component basis (P_1,P_2,P_3). Each of these basis is represented by a spectrum response $S_{P_i}(\lambda)$. Therefore, a channel P_i of an image $I(u,v)$, considering the visible spectrum situated between 350 nm and 780 nm, can be written as

$$I_{P_i}(u,v,n)=\int_u^{u+1}\int_v^{v+1}\int_0^{\Gamma}\int_t\int_{350\,nm}^{780\,nm} L(x,y,\gamma,t,\lambda)W_{nR}(t)S_{P_i}(\lambda)\,du\,dv\,d\gamma\,dn\,d\lambda. \tag{2.7}$$

In order to make $L(x,y,\gamma,t,\lambda)$ be a good representation of the plenoptic function $f(O_x,O_y,O_z,\theta,\phi,\lambda,t)$, it is also necessary to determine if each of the dimensions are sampled with a high enough density.

For example, for a video sequence, the time is well represented if the number of frames per second is enough to cause, from the human point of view, a fluid transition between frames.

In addition, as pointed out in the discussion leading to the Equations 2.5 and 2.6, a pinhole camera structure does not allow the registration of all θ and ϕ angles. This angle depends on the focal length f and the size of the camera plane.

All the above considerations are valid for a single camera, i.e. they do not take into consideration the measurement of the plenoptic function for several camera centre positions (O_x, O_y, O_z). It can be approximated by placing multiple cameras in the space.

Clearly is not possible to acquire a full plenoptic function, over all the desired space. By limiting the number of cameras and defining the scene of interest, there are many possible camera arrangements. Cameras can be placed randomly around the scene of interest, but it would be cumbersome to handle the resulting images. One simple arrangement consists of cameras being placed at points belonging to a 3-D Cartesian grid.

An image, belonging to a 3-D Cartesian grid can be represented by $I^{a,b,c}_{P_1,P_2,P_3}(u,v)$ being a, b and c the index of the image in the grid. Another possible arrangement is a two-dimensional Cartesian grid. Figure 2.4 exemplifies this arrangement.

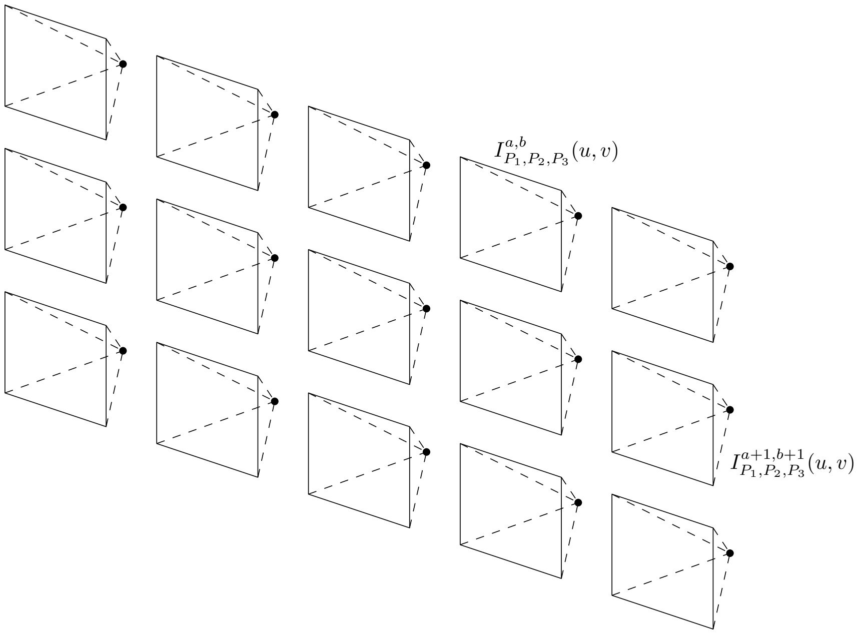

$I^{a,b}_{P_1,P_2,P_3}(u,v)$
$I^{a+1,b+1}_{P_1,P_2,P_3}(u,v)$

Figure 2.4: *Example of a two-dimensional camera array.*

All these multidimensional simplifications of the plenoptic function are called *light-fields* [8].

A set of images $I^{a,b,c}_{P_1,P_2,P_3}(u,v)$ in a Cartesian grid is given the name of *5D-light field* as one needs five coordinates to represent it.

Similarly, $I^{a,b}_{P_1,P_2,P_3}(u,v)$ is known as *4D-light field*. Light fields are families of images which recently have become popular by their innovative capabilities such as refocusing, reduction of exposure time, computation of synthetic images from different viewpoints, etc.

Recent devices, called *light-field cameras* [1] (see Figure 2.5), were developed specifically to acquire light fields and allow the user to produce images with some desired properties. Examples of these properties are the focus (as exemplified in Figure 2.6), the exposure, the position of the camera, etc.

Figure 2.5: *A light field camera. Image taken from [1].*

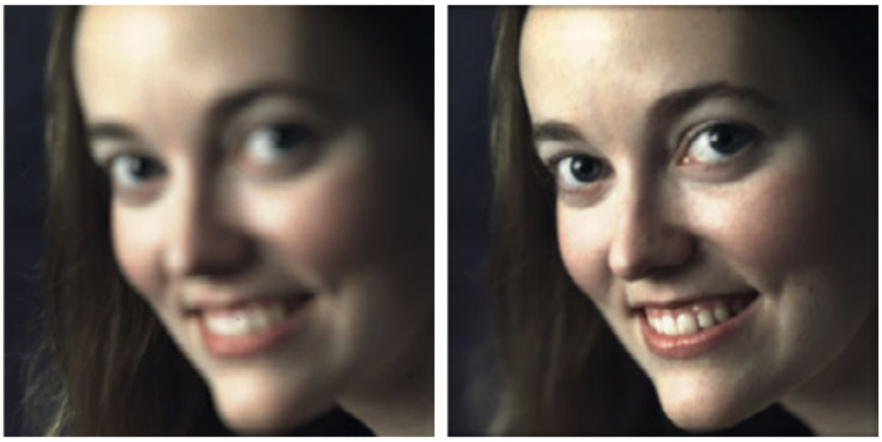


Figure 2.6: *An example of a post-refocusing of an image captured by a light field camera. Image taken from [1].*

When acquired by a camera array, a 3D-light field is often referred to as a *multiview sequence*. They are represented as $I^{a}_{P_1,P_2,P_3}(u, v)$ function.

Nowadays, there are mainly two ways to acquire a multiview sequence. The next section explains these and presents the arguments for and against each one of them.

2.2 Multiview acquisition

As explained in Section 2.1, a light field can be sampled by a 1-D, 2D or 3D array of cameras. This array can be implemented using two methods depending on the scene of interest: physical cameras with centres placed pre-established points for real environments, or virtual cameras placed in computer modelled scenes.

2.2.1 Camera grid

Placing physical cameras on a grid in order to record a scene is not an easy task. For example, it would be difficult to construct a 3D-array of cameras as the cameras cannot be placed in front of each other to avoid interfering with the scene acquisition.

Two-dimensional and one-dimensional arrays are easier to construct, but multi-view sequences with small distance between cameras may not be built due to the physical camera sizes.

An alternative solution would be placing a single camera which moves into different pre-established positions. However, this method would not allow recording a scene where the objects are moving (non-static scene) as the image correction for misplacement and misalignment would be difficult.

Instead, multiple cameras can compose the grid, as shown in Figure 2.7. The problems of this solution are that it is very difficult to achieve perfect synchronization between cameras and consistently adjust the various camera parameters, e.g. focal length, brightness, and exposure time. In addition, the number of cameras to be used is limited.

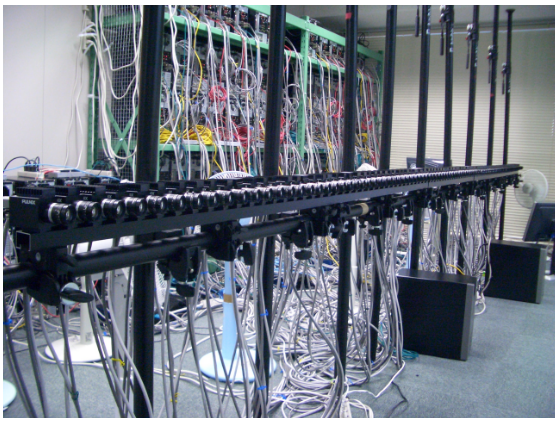


Figure 2.7: *1D-Camera arrangement designed by Nagoya University, Japan [2]*

After the acquisition, it is also necessary to synchronize all cameras and perform camera calibration to correct brightness and compensate misalignment and misplacement.

In spite of being fast, once the cameras and the environment are ready, any kind of scenarios can be recorded. However, the physical camera array can be very costly. For that reason, the rendering process presented in the next section is an alternative often preferred.

2.2.2 Sequence rendering

An alternative for placing real cameras to sample a light field is to get images from a computationally created environment. One of the most popular rendering algorithms that can be used for this purpose, and that provides a reliable description of the scene, is the so-called ray-tracing algorithm [9].

The ray-tracing method uses the light properties as well as its path through the environment to compute the pixel values as in Equation 2.7. The result is an image quite realistic as the light, materials and its interactions like reflectance, absorption, transparency etc, are all physically modelled.

With the use of ray-tracing algorithms, there are no restrictions associated with the camera position and distance between cameras. On the other hand, modelling a realistic scene can be very difficult and the rendering time can be very large, naturally always depending on the available computational resources.

This work will use different sequences acquired using both of the above described methods in its tests. The next section describes these sequences.

2.3 Description of the used multiview sequences

For a better understanding of how multiview sequences will behave when subject to different compression tools, it is essential to know each sequence main characteristics and structure as they can play an important role in the compression performance.

In order not to produce biased results, ten different sequences are chosen to be part of the subsequent experiments. These sequences include both light fields captured by camera arrays and rendered light fields using the Physically Based Ray-Tracer software (PBRT) [10].

In total, there are ten sequences, created in four different places, being five of them static (only one time instant) and the other five video sequences with a number of views varying from 7 to 850 views. The sequences *San Miguel-UHasselt*, *Kendo*, *Pantomime*, *Balloons*, *Champagne Tower* and *Dog* are also used as multiview test sequences by ISO/IEC JTC1/SC29/WG11, the MPEG standardization group.

Format
All multiview sequences are stored in the YUV 4:2:0 raw format. This format uses a colour space called YUV ($\{P_1, P_2, P_3\} = \{Y, U, V\}$), where the Y channel carries the *luminance* information and the U and V channels carry the chrominance (hue and saturation information).

For each channel, YUV 4:2:0 format uses 8 bits per pixel for each channel. This format also attributes only a half of the total image resolution (width and length) to U and V channels.

Then, there are four times more Y pixels then U pixels (see Figure 2.8).

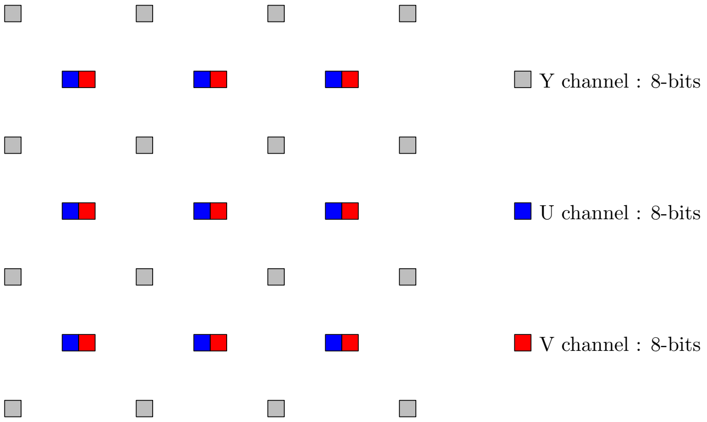


Figure 2.8: *Arrangement of luminance and chrominance in the YUV 4:2:0 format*

The Appendix A presents and gives a brief description of each of these multiview sequences.

Chapter 3

Proposals

As introduced in Chapter 1, multiview imaging is a form of video content which brings extra visual perceptions to the users. These perceptions are extracted from multiple cameras spread around the scene, giving to the users a more complete overview of the environment they are watching.

However, these extra perceptions come with a price. The extra viewpoints are very costly in terms of bits to broadcasting services and to computational resources, such as storage.

Due to this large amount of data involving the work with multiview sequences, it becomes essential to develop new compression techniques to make it viable to be largely used by people.

As the multiple viewpoints in a multiview sequence always record a same scene, there are common aspects to the scenes in different cameras, which lends them a significant inter-view redundancy. This redundancy depends on how the cameras are placed, the distance between them, their configuration, etc.

3.1 Literature review

Many efforts have been made in order to reduce the large amount of data of multiview contents. Among the most popular types of multiview content are the images from light-field cameras and the images from an array of cameras (real or synthetic).

Content generated by specific light-field cameras, like the Lytro Camera[11] is then object of study on the multiview compression subject. The array of microlenses placed inside the camera acts to split the light according to its incident angle. The resulting image is a composition of many micro-images, each one depicting the scene as it was a different camera.

As each micro-image in a light-field image displays a slightly different point of view, the micro-images carry strong similiarities among them. Different approaches have been studied in order to improve the compression performance on such content. This similarity between diferent micro-images can be explored if an algorithm is capable of searching contents of one micro-images in others. Some solutions, as the algorithm presented in [12] and [13], called *Self-Similarity Compensated Prediction* use block-matching used inside a light-field image in order to overperform compression results from HEVC intra-coding[14], saving up to 66.22% of the bitrate.

A combined approach was investigated in [15]. This approach adds to the Self-Similarity Prediction of [12] and [13] variations of the HEVC prediction modes. These prediction modes use the information of blocks that were already encoded to estimate a block to be encoded. It uses the idea that a view can be estimated by a combination of other nearby views. Results show that the proposed combination can reduce the bitrate in 31.86%, in average, when compared with the HEVC intra-mode.

For multiview sequences generated by camera arrays, there are also many compression techniques in the literature. One approach to multiview compression involve the usage of depth maps. Depth maps are images that inform the depth corresponding to each pixel in the scene. With the help of depth maps, the 3D geometry of the scene is known and it is then possible to synthesize new viewpoints for the sequence. This type of algorithms, called Depth-image-based rendering (DIBR) [16], permits the use of more sparse sequences, since intermediate views can be synthesized. However, these algorithms depend heavily on the quality of the depth map in order to calculate the 3D correspondences correctly.

For multiview sequences without depth information, view synthesis algorithms, as in [17] and [18] are capable of synthesizing intermediate viewpoints by searching blocks that are present in two views and shifting the block to a position proportional to the desired viewpoint. These synthesized views, in replacement of some original views, can reduce the total bitrate of the sequence.

As a complement for these works, an objective metric for measuring the quality of the horizontal-parallax effect on multiview sequences was tested in [19]. It considers that the quality of the interaction with the sequence consists in a weighting of the intra-picture quality and the parallax quality, that is based on the optical flow [20]. Results show that the correlation with subjective scores is high for sequences with high density of views.

3.2 Proposed experiments

As one can see, the approaches for either light-field images or multiview sequences share some similarities. Both of lines of research try to find similarities between different viewpoints: micro-images for light-field images or views for an array of cameras.

Taking as example a linear arrangement of cameras, one can divide the redundancy that is present in a multiview content in three parts:

- **Temporal redundancy:** Similarities in content between different time instants;
- **Intra-view redundancy:** Similarities between parts of the same image due to the scene construction and composition;
- **Inter-view redundancy:** Similarities shared between different viewpoints due to the scene composition and the cameras positioning.

The main focus of this work is the inter-view redundancy, thus, only static multiview sequences (each view has only one time frame) will be used. Therefore, the redundancy present in the multiview sequence are only the inter and intra-view, where the inter-view redundancy comes from its multiview nature.

In order to study how one can store data from multiview content, it is necessary to relate this redundancy to the multiview settings and parameters. This relation should exclude the influence of intra-view redundancy, once it is not related to the multiview compression issue. The intra-view compression issue is out of scope of this work, being well developed in the most recent codecs [21].

This work will study the relation of the multiview redundancy with some of the multiview sequences main parameters and compare different coding solutions to find out good settings for multiview compression.

This study is divided in three parts:

1. **Simulcasting coding:**

   Simulcasting, a word blending for simultaneous broadcasting, is an independent viewpoint compression, ignoring the multiview nature of the content. This study is necessary to estimate the effect of intra-view redundancy in the compression of each tested sequence.

   Once the intra-view redundancy contribution to the compression of each sequence is known, it will be possible to calculate how each method estimates the inter-view redundancy when the multiview compression is made.

The simulcast coding also can indicate how complex is the content of the sequence since the simulcast performance depends only on the content present in a single view.

2. **Multiview coding:**

   Study of some techniques to reduce the total amount of bits of a sequence using the information shared by more than simply one view. The multiview compression will consider the relation among different views under different coding conditions.

   In this experiment, parameters such as distance between views, density of views, as well as size and type of the compression structures, are evaluated in terms of compression performance. All these results are written in comparison with the simulcast performance in order to measure the inter-view redundancy reduction;

   From these results, one can evaluate which set of parameters produces better results in inter-view redundancy reduction. These results can then be used for future multiview compression works.

3. **Multiview coding with redundancy removal:**

   This experiment will study a form of redundancy removal using the results and parameters of the multiview coding experiment. This removal will be made by discarding a certain number of views, in order to diminish the total amount of data, and replace them by views synthesized from adjacent encoded views.

   The experiment will continue its analysis by verifying how the removal of those views affected the compression performance. If it provides negatives results, it means that this removal had discarded important inter-view information that could be properly exploited by the studied compression methods (HEVC in this case). If the compression performance is improved, it means that only redundant information was discarded and then this removal technique is effective.

   The diagram of Figure 3.1 presents the experiment idea and each constituent part of it

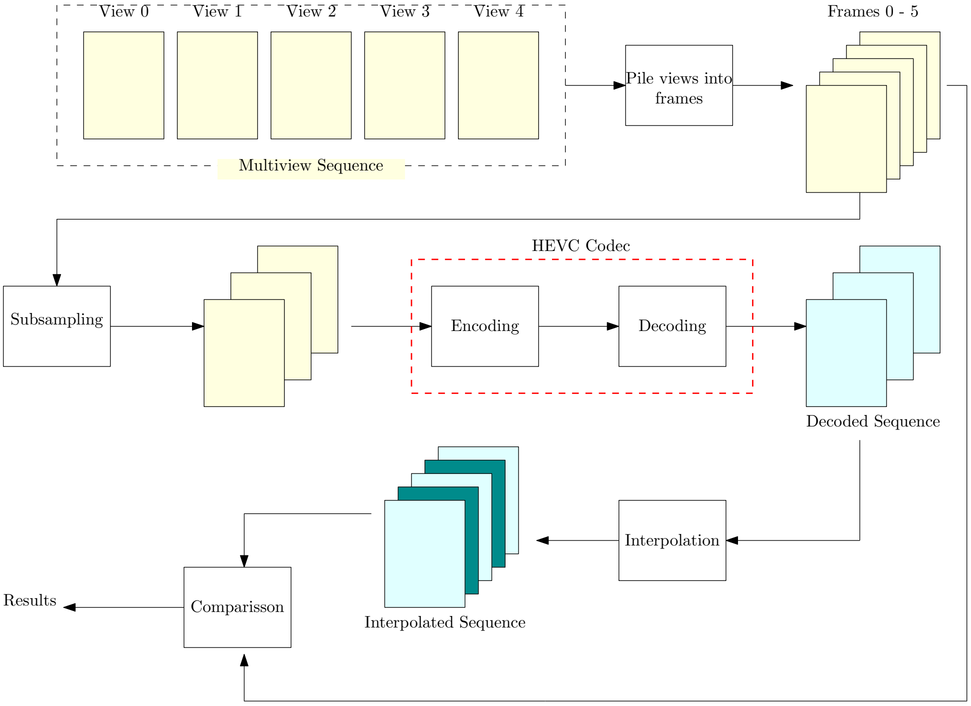


Figure 3.1: *View decimation and decoder interpolation experiment schematic.*

The next chapter will explain all the methodologies used in these experiments as well as the used performance metrics and performance comparison methods among different compression results.

Chapter 4

Methodology

This chapter introduces the methodology used for carrying out the main proposals of this work as detailed in Chapter 3.

The first step of this study is to understand how the views arrangement can affect the multiview compression performance and how to make it measurable. The most important parameter related to the views arrangement to be studied in this work is the view density. Section 4.1 introduces the view density concept and describes the method used to measure it.

After that, this chapter will also introduce the coding tools which all sequences will be subjected to in the compression experiments as well as explain all important parameters and metrics used in those tests.

4.1 View density

Aside from the characteristics of each of one the views that composes a multiview sequence, like resolution, bit depth, colour space, etc; there is another important feature to be considered: the camera arrangement.

Cameras usually can be arranged linearly (as a row), in an arc, as a two-dimensional array, spread all over the environment, etc.

In this work, only linearly-arranged multiview sequences will be used. Linearly-arranged sequences are the simplest form of a multiview sequence and also the one for which there is more available content for research.

Among linearly-arranged multiview sequences, there is an important factor that differentiates one arrangement from another: the distance between adjacent cameras.

A good value for this distance, from the point of view of the user interaction, depends on how the views are disposed. In other words the distance between views also depends on the room and where they are displayed on the room. It depends also on the monitor size and model, disposition of the monitor in the room, distance between the monitor and the viewer, etc.

This distance can influence on how the user can interact with a multiview sequence. In the room setting used for this work, the user interacts with the multiview sequence by moving around the room, with his head position being captured by a tracking camera placed above the monitor. The quantity of views the user can perceive per degree inside the tracking camera coverage is called *view density*.

The procedures performed to measure the view density took place in a room located at the SMT Laboratory from COPPE/UFRJ (see Figure 4.1). A 3D monitor JVC 463D10U 46' with circular polarization was used for displaying the sequences and an OptiTrack V:120 Duo camera was used for determining the viewer position. For example, a multiview sequence that has many views in a small angle of view provides a smooth transition between views. A multiview sequence with less views per angle of view can provide a harsh transition even though it would consume less storage space.

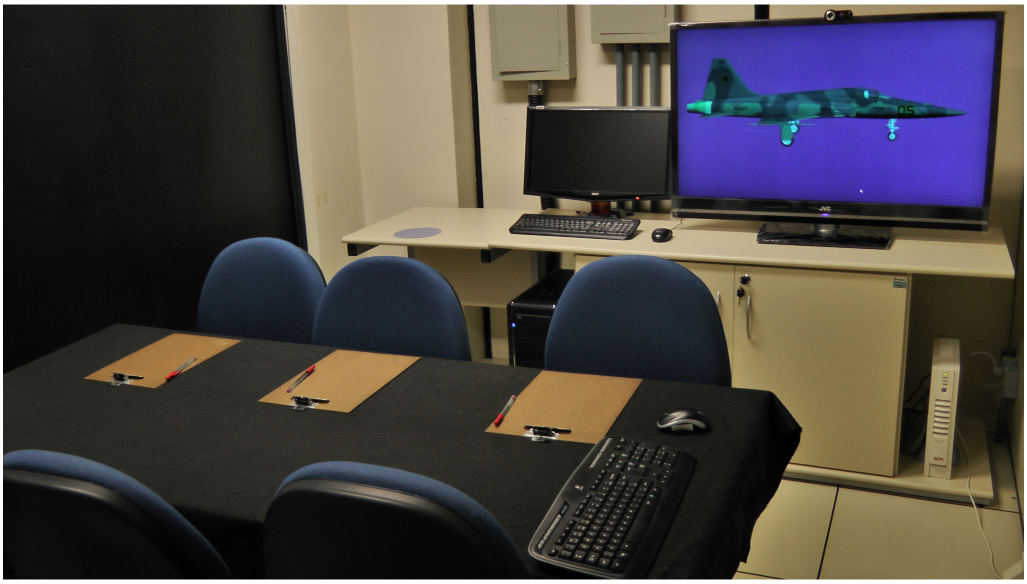


Figure 4.1: *Room used for view density measurements.*

In the tested configuration, the viewer can only move along a line parallel to the screen in order to navigate through the scene between different points of view.

We start by establishing the leftmost camera image to be shown when the viewer is at the very left of the room and the rightmost image to be shown when the viewer is at the very right of the room. The intermediate viewpoints will be equally distributed in the spaced between the rightmost and the leftmost viewpoints. Using these settings, the room can be modelled as seen in Figure 4.2.

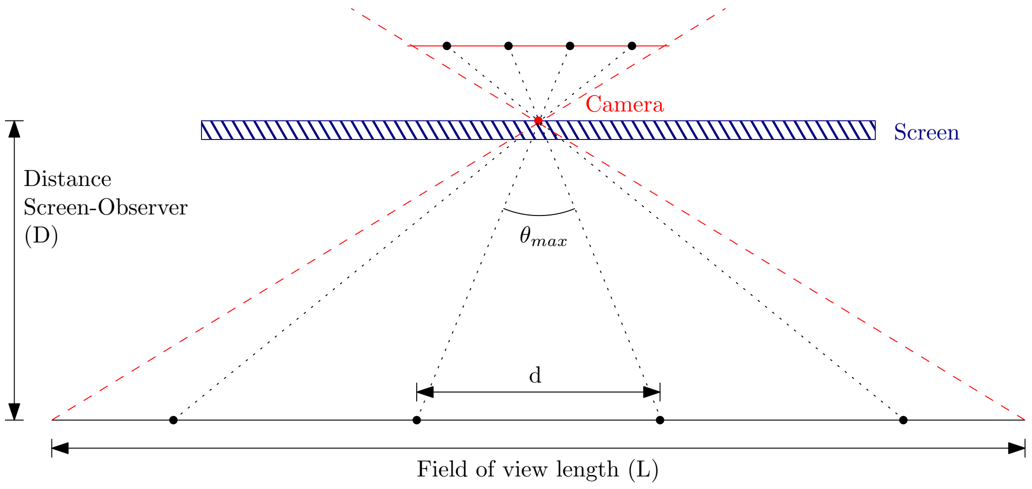


Figure 4.2: *Room model for view density measurements.*

The measurements to be obtained in order to find the view density of each sequence are:

- **The field of view length** (L)**:** It is the distance between the left and right walls. In the used room,
$$L = 270\,cm$$
- **Screen-observer distance (D):** It is the distance between the center of the screen and the usual position of observation right in front of the screen. In this set-up,
$$D = 203\,cm$$

From Figure 4.2 one notices that the maximum angle θ_{max} between the two central adjacent views can be calculated using the angle formed between two corresponding points from the two most central views and the tracking camera. Thus, this angle can be calculated by:

$$\theta_{max} = 2\arctan\left(\frac{d}{2D}\right), \tag{4.1}$$

where d is the distance between two points of view. Once the vision range of the tracking camera was set as the room width, d can be represented as $d = \frac{L}{N_{views}}$ and θ_{max} as

$$\theta_{max} = 2\arctan\left(\frac{L}{2N_{views}D}\right). \tag{4.2}$$

This formula can relate the view density with the number of views of each multiview sequence. As one can see, the view density is an angular measure and it can be defined as is the amount of views the user can perceive within an angle in front of the tracking camera or, in other words, it is measured in views/degree.

4.2 Multiview sequence classification

Once the angular view density formula was derived in 4.2, one can find the angular view density of each sequence to be tested. High values of angular view density (or from now on angular density) means that the sequence displays a denser grid of views than sequences with low values of angular density. Table 4.1 presents the results for each sequence.

Table 4.1: *Angular view density (in views/degrees) for multiview sequences*

Sequence Name	View Density (views/degrees)
San Miguel - UFRJ	11.1539
Audi TT	8.3983
San Miguel - UHasselt	2.6245
Champagne Tower	1.0498
Pantomime	1.0498
Dog	1.0498
Balloons	0.0921
Kendo	0.0921
Elephant	6.0363
Train	6.5611

From the results above, it is possible to distinguish at least three groups of angular view densities:

- Sequences *San Miguel - UFRJ*, *Audi TT*, *Train* and *Elephant* can be classified as high-density multiview sequences, or supermultiview sequences, as they have the largest density values;

- Sequences *San Miguel - UHasselt*, *Champagne Tower*, *Pantomime* and *Dog* can be called medium-density multiview sequences as they present intermediate values of angular density;

- Sequences *Baloons* and *Kendo* can be called low-density multiview sequences as they have very low angular density values.

The angular density, as a measurement inversely proportional to the distance between the cameras, is related to the correlation between multiview images. Hence, it is fundamental to consider this measure as an important feature impacting on the coding efficiency behaviour of a multiview sequence.

4.3 Coding tools

4.3.1 High Efficiency Video Coding (HEVC)

In order to perform the simulcasting as well as the inter-view compression, the most recent video compression standard will be used: the High Efficiency Video Coding (HEVC).

Recently, the HEVC standard, developed by the ISO/IEC JTC 1/SC 29/WG 11 Moving Picture Experts Group (MPEG), jointly with ITU-T SG16/Q.6 Video Coding Experts Group (VCEG) is considered the current state of the art video codec [22]. HEVC, also known as MPEG-H Part 2 and ITU-T H.265, is the successor of H.264 / MPEG-4 AVC , being published on November 25, 2013.

The main goal of HEVC is to save at least 50% of the bitrate over H.264 under the same quality level. HEVC has brought many new coding tools suited to coding new emerging contents, e.g. new high-quality formats as Ultra-HD, formats for mobile devices and multiview contents. HEVC also made improvements on parallel processing tools allowing a reduction in encoding and decoding time.

Although HEVC presents new features and improved techniques, its basic coding architecture follows the well-established architecture from previous video codecs, as for example H.264. A simplified diagram of the HEVC architecture is presented in Figure 4.3.

At a first sight, this architecture can be confused with the H.264 architecture. However, inside each block there are many changes that allow HEVC to have a superior coding performance. Those differences are presented in the next paragraphs.

Signal Partitioning

Instead of H.264 partitioning, where each frame is partitioned in 16 × 16 macroblocks, HEVC presents the Coding Tree Unit (CTU). The CTU is a composition of three Coding Tree Blocks (CTBs) – two CTBs for chrominance and one for luminance – which can have the sizes of 16 × 16, 32 × 32 or 64 × 64 pixels.

Each CTU is a root node of a quadtree structure where each children node is called Coding Unit (CU) which, following the example of the CTU, consists in two chrominances and one luminance Coding Blocks (CB).

From the CU tree level, two structures emerge: the Prediction Unit (PU) with its three Prediction Blocks (PB) and its partitions to be used in the prediction stage of the coding process, as well as the Transform Unit (TU) and its Transform Blocks (TB), which will be submitted individually to the Transform block.

All these changes in partitioning, with the addition of larger block structures and the differentiation between prediction and transform structures, allow a better

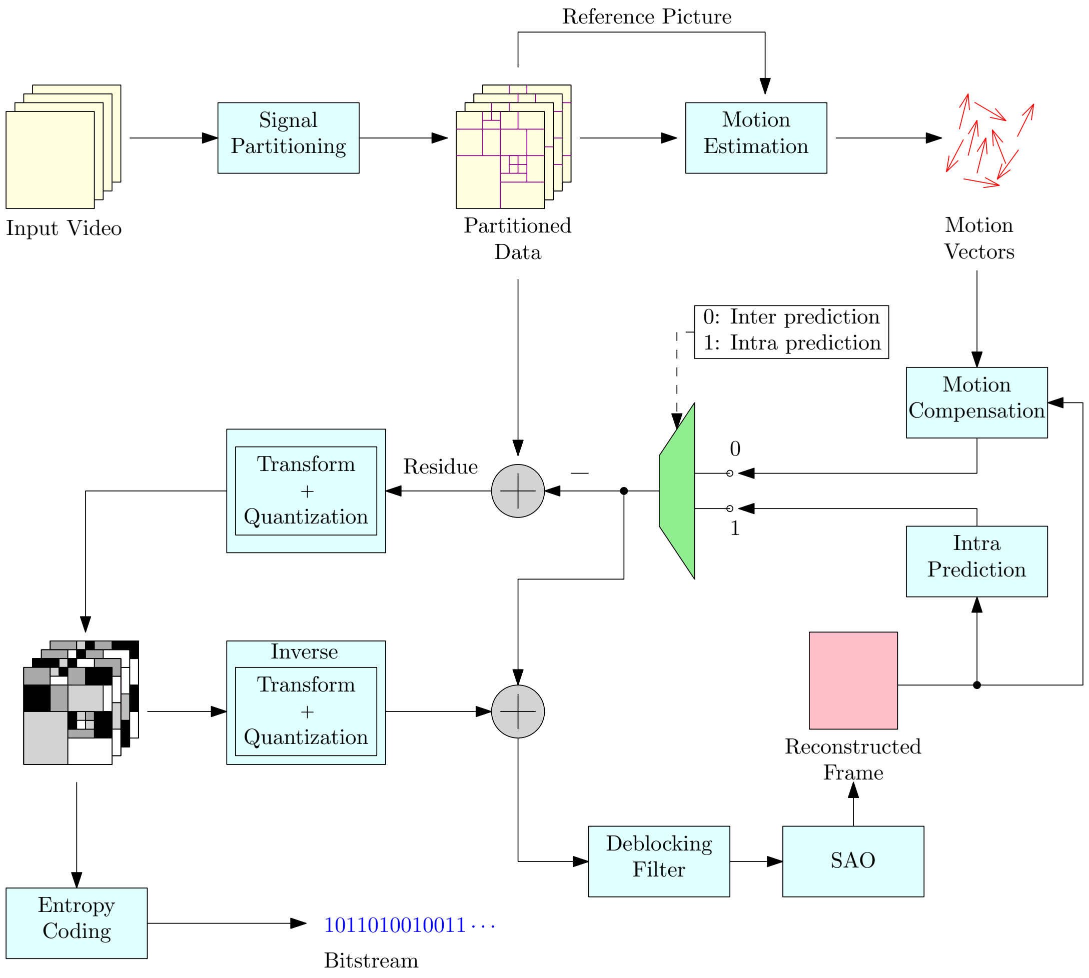


Figure 4.3: *HEVC encoder block diagram*

performance comparing with the H.264 in compressing contents with larger resolutions.

Motion Estimation

There were significant differences in Motion Estimation from H.264 standard. Aside of the H.264 motion estimation methods, two methods were added to the HEVC: The Advanced Motion Vector mode, which uses adjacent prediction blocks to estimate candidates for a Motion Vector (MV); and the Merge Mode where motion vectors can be directly borrowed from estimated adjacent blocks.

Motion Compensation

The HEVC motion compensation is very similar to the H.264 motion compensation by using a quarter-pixel precision for motion vectors. The interpolation for fractional pixels however, uses an 8-tap or 7-tap filters instead of the 6-tap filter used by the H.264.

Intra Prediction

The HEVC uses up to 33 directional intrapicture prediction modes compared with the 8 modes from H.264 plus a surface fitting (planar) and a flat (DC) prediction.

Transform

The HEVC standard has added Transform Blocks of sizes 16 × 16 and 32 × 32 to the existing 4 × 4 and 8 × 8 blocks from H.264 to increase the coding performance over high resolution contents. The transform block uses a new integer transform, similar to the Discrete Sine Transform (DST) for 4 × 4 luminance blocks and a function similar to the Discrete Cosine Transform (DCT) basis functions for the other TB cases.

Quantization

For the quantization block, HEVC adds the support to new TB sizes to the H.264 quantization design and keeps the variable which controls the output quality, the quantization parameter (QP), in the range of 0 to 51 for eight-bit depth contents.

SAO Filter

While the H.264 standard uses a 4 × 4 deblocking filter, the HEVC uses a filter version with the size of 8 × 8. After deblocking, HEVC compensates the deblocking filter by adding an offset to the decoded pixel value. This block is called Sample Adaptive Offset (SAO). In the HEVC standard, a new amplitude mapping was added using statistical analysis of the signal in order to improve the deblocking operation;

Entropy Coding

The HEVC uses a Context Adaptive Binary Arithmetic Coding (CABAC) which is very similar to the Entropy coder used in H.264 except for parallelization routines.

Other Structures

Two entirely new structures were added to HEVC standard to provide new ways to deal with synchronisation and parallelism: tiles and slices.

Tiles are regions of the content which can be decoded independently in order to provide some parallelism to the coding and decoding processes.

Slices are sets of sequential CTUs created with the purpose of synchronisation in case of data loss and to serve as a limit to the influence of the prediction block. A common choice for a slice is an entire video frame or a view in case of multiview sequences. Slices are classified in three types:

- **I-slice**: Slice such that CUs contained in it slice can be only be predicted by other CUs of the same picture;

- **P-slice**: Slice such that only uses motion prediction from one other picture per Prediction Block (uniprediction);
- **B-slice**: Slice such that can use motion predictions from other two pictures per Prediction Block (biprediction).

In the HEVC standard, usually all three types of frames are organized periodically in time. The size of its period is called GOP (Group of Picture) size. The bigger the GOP size, the more is the temporal redundancy between frames exploited.

4.3.2 Multiview compression

An inter-structure prediction is defined as a specific type of prediction in which data from another similar structure is used to estimate the current one in order to reduce the total amount of data to be encoded. For example, if components of an image block are used to predict another block, this operation can be referred to as an inter-block prediction.

Regarding video sequences, prediction can as well be carried out in the time dimension, diminishing the temporal redundancy contained within video frames. This redundancy is related basically to the usual video content, in which variations between frames are usually small, except in occasional changes of scene, rapid camera movements, etc.

In this work, another type of prediction will be explored: the inter-view prediction. This prediction uses the visual information present in one camera to estimate the data of another camera. The justification for the use of this operation is similar to the one for the use of inter-frame prediction: small variations between its structures due to the natural redundancy present in time and space.

Nevertheless, there are some differences between inter-view and temporal redundancies. Whereas temporal contents can change completely from one frame to another, the visual information is usually consistent between two adjacent cameras. For example, consider that the images of a linearly-aligned multiview sequence are displayed sequentially as frames in a video sequence following and ordering based on the camera position. In this case the resulting video sequence would be similar to the one generated by a single camera moving along the positions of the cameras. Such a movement would not generate any abrupt change in its contents. In this case, although some object occlusions can clearly occur, their occurrence would be consistent with the camera movement.

For these reasons, one might expect that a multiview sequence, when submitted to the same encoding process of temporally-acquired video frames, can have its inter-view redundancy adequately explored as it was inter-frame redundancy.

Concerning the multiview coding, the Joint Collaborative Team on 3D Video Coding Extension (JCT-3V) created two extensions for the HEVC standard where prediction tools are employed in order to reduce inter-view redundancy.

One of them, called Multiview-HEVC (MV-HEVC) , is the HEVC extension created for multiview video. For a given quality, it tries to reduce the total bitrate by exploring not only the spatio-temporal redundancy within each view, but also by exploring the redundancy present among different camera views.

The other, called 3D-HEVC, deals basically with multiview sequences containing depth maps. In this extension, depth information is used to estimate the position of an object in a predicted view. Besides depth information, camera parameters must also be encoded in this extension.

Unfortunately, several multiview sequences used in this work do not present any depth information or camera parameters. In addition, as this work concerns only static multiview sequences, there is no need to use neither the 3D-HEVC nor the MV-HEVC. The HEVC software using the inter-frame tools along the views to perform the inter-view prediction will suffice to our goals.

In video sequences, the inter-frame prediction implies that the nth frame, denoted by $\phi[n]$ can be predicted by a set of other frames resulting in an estimated frame $\tilde{\phi}[n]$. The closest $\tilde{\phi}[n]$ is to $\phi[n]$, the best can be the prediction and the less information is required to represent the difference between $\tilde{\phi}[n]$ and $\phi[n]$. This difference is what will be added to the encoded file increasing the total bitrate, so it is necessary to find the best ways to predict a $\tilde{\phi}[n]$ in order to reduce that difference.

Consider that $\tilde{\phi}[n]$ is a result of an inter-frame (or inter-view) prediction in which Mframes were used in the process. This set is called a Group of Pictures (GOP) for video sequences or Group of Views (GOV) of size M for multiview sequences.

Regarding the choice of which frames/views are the best to be part of this GOP/GOV, it is not a bad assumption to consider that the closest frames/views in time or space are the ones which can provide a better prediction than the distant ones. For example, frames within a specific second of video are most probably to be similar while a frame a minute later, probably is totally different from these ones. The choice of views, for a GOV follows the same logic: spatially closest views are the best candidates to be part of a GOV then the distant ones.

Regarding the choice of the number of composing elements in the GOP/GOV, one can say that few elements are more probable to produce a poorer estimation $\tilde{\phi}[n]$ than a larger group. However, it is not true to consider that if we expand the group as we want, a better approximation $\tilde{\phi}[n]$ will be achieved. It is expected that the performance of the predictor will stop increasing significantly after a reasonable and significant group of frames/views has already been added to the GOP/GOV. Another important factor to be considered is the computational complexity involved in predicting a frame/view. The largest the group of elements, the more complex is the prediction process.

Considering that the contents of a video sequence can often change drastically, usually the GOP size should not be very large since uncorrelated frames are not helpful. On the other hand, multiview sequences always depict the same scene except from the camera position. The changes between views depend only on the quantity of occlusions and the distance between the cameras. Hence, GOV sizes can be large, since there is much less probability that uncorrelated views belong to the group.

It is also necessary to define the role of each view inside a GOV. A GOV should be organized so that each view inside it can be predicted by other views inside the same GOV. The HEVC standard (as well as its predecessor, the H.264 standard) have established that each view can in principle use all the other views from inside the same GOV as long as view type restrictions for prediction are respected.

In this work, the type of a given view will be referred as the type of the slices contained in the view, as we will assume one slice type per view as has been done in Section 4.3. Thus, there are three main types of views:

- **I-views:** Views that use I-slices. These views are not predicted by other views from the GOP and can be seen as references for the other predictions;

- **P-views:** Views that use P-slices. In this work, P-views are always predicted by another P-view or an I-view;

- **B-views:** Views that use B-slices. These are the views predicted both by I-views , P-views or even other B-views.

Using these three types of views, the GOV prediction structures can be diagrammed. In this work, four structures were chosen to organize the inter-view prediction.

- **IIIII:** All views are encoded without prediction of other views. This is the encoding structure used in the simulcast experiment;

- **IPPPP:** This structure organizes the views in order to be predicted only by the previously predicted view (except for the first I-view). This scheme predicts the view considering only one direction, for example, from the view immediately in the left. This structure can be observed in Figure 4.4;

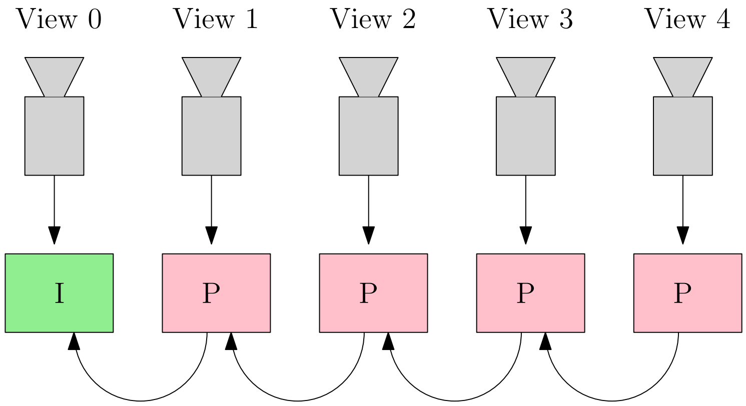


Figure 4.4: *IPPPP inter-view prediction structure.*

- **IBBBI:** This prediction structure is also called a dyadic GOV and creates layers of prediction (hierarchical layers) starting with two I-views and separated by $M-1$ views. These two I-views composes the prediction of the view equidistant from both of them creating a first layer. This view in the first layer predicts two more views equidistant to both I-views, creating a second layer. This process repeats until there are no more views within the GOV to be encoded. A dyadic GOV is depicted in Figure 4.5.

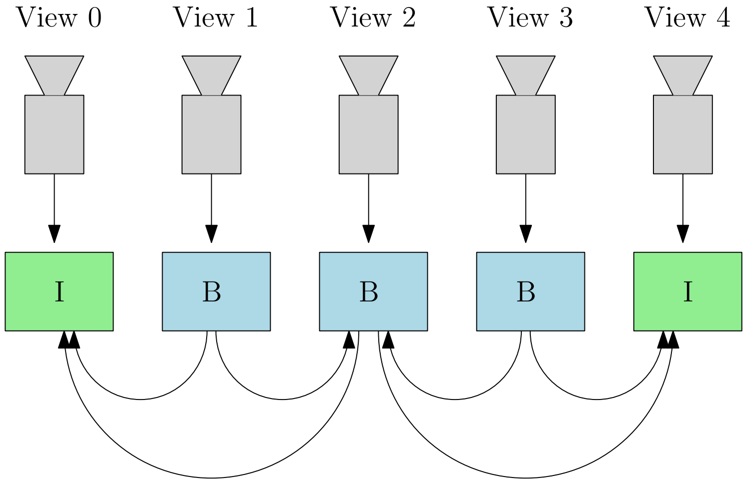


Figure 4.5: *IBBBI inter-view prediction structure.*

- **IBBBP:** Similar to the IBBBI prediction structure, this structure uses as the last view in the GOV an P-view instead of an I-view. Then, the P-view is used as the first view in the next GOP, if it exists.

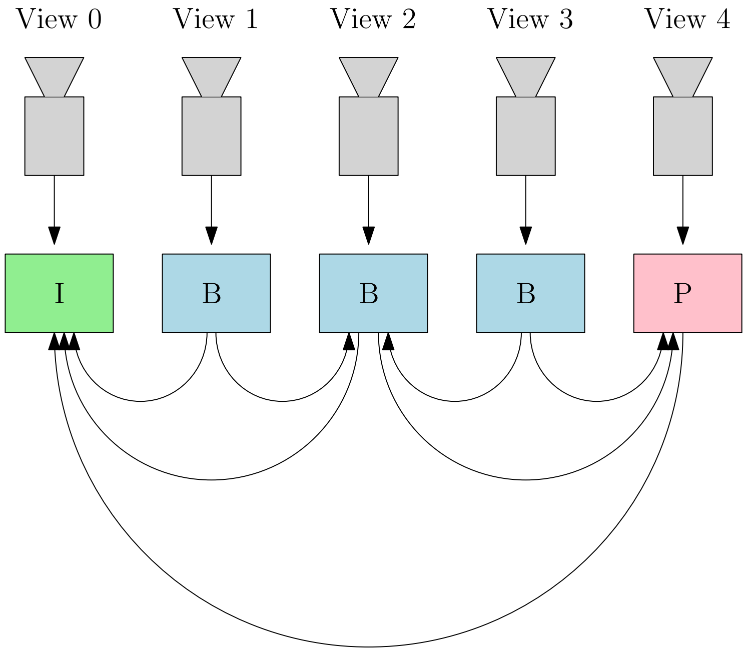


Figure 4.6: *IBBBP inter-view prediction structure.*

These structures were chosen to be part of this work for the following considerations:

- A GOV must begin with an I-view or a P-view if it is not the first GOV;
- A GOV cannot contain I-views except in the first and in the last positions;
- As the viewpoints in the tested multiview sequence are linearly spaced, it is reasonable to choose prediction structures for B-views where the two views chosen for its prediction have the same distance to the predicted B-view. For this reason, the IBBBI and the IBBBP structures were chosen to be hierarchical, which fulfil this consideration.

By taking into account these above-mentioned considerations, one can define the methodology and the conditions necessary to perform the experiments proposed on Chapter 3.

4.4 Methodology description

4.4.1 Test parameters

In this section, it will be presented the methodology conditions used in this work to carry out the experiments described in Chapter 3.

Codec

The HEVC reference software HM-16.0 [23], provided by the Fraunhofer Heinrich Hertz Institute, was used in this work to encode the multiview sets. They were all coded using the HEVC main profile level 6.2.

Views disposition

As all multiview sets in this work are static, one can use the temporal prediction of the HEVC encoder by assigning each viewpoint to a video frame. For example, the Kendo sequence has 5 views. This sequence is then transformed into a video file with five frames, being the leftmost view corresponding to the first frame and the rightmost view corresponding to the last frame.

The advantage of formatting a multiview set into a single video file is the power to apply not just the intra-view prediction tools to all views at once but also choose the inter-view tools to be used among them.

Quality levels

One important parameter in the HEVC codec, that can be modified to change the quality of the coded sequence is the *Quantization Parameter*(QP) . The Quantization Parameter is a rate control parameter that tells the quantizer how much spatial detail is discarded. The QP value is proportional to the quantization step applied to the encoder transform coefficients. Hence, the greater the QP, the more details are discarded by the quantization process, resulting in an encoded image with poorer quality and smaller rate. On the other hand, a small QP value preserves more details resulting in an encoded image with better quality at the cost of a higher rate. For an eight-bit sequence, the QP value can assume values between 0 and 51.

In this work, we will test a total of eight quality levels, corresponding to eight QP values. Those values are :

$$QP = \{10, 15, 20, 25, 30, 35, 40 \text{ and } 45\}.$$

View density levels

Another parameter that will be tested in this work is the compression behaviour according to the angular density of the sequence. One can change the angular density of a sequence by changing its number of views. A simple way to decrease the number of views is by keeping equally spaced views. For example, if one discards three views in each group of four views, we refer to this operation as a subsampling factor of four. In this work, a sequence of subsampling-by-two operations will be used to implement these factors, as seen in Figure 4.7. For this reason, all chosen values of subsampling factors are all powers of two from 1 up to 512.

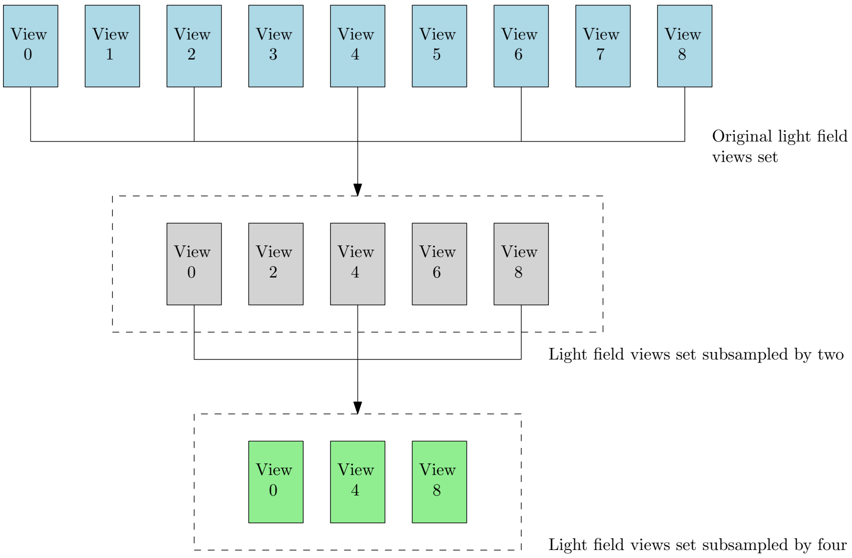

View 0
View 1
View 2
View 3
View 4
View 5
View 6
View 7
View 8
Original light field views set
View 0
View 2
View 4
View 6
View 8
Light field views set subsampled by two
View 0
View 4
View 8
Light field views set subsampled by four

Figure 4.7: *View Subsampling scheme in a hypothetical multiview sequence with eight views.*

However, the maximum subsampling factor value depends on the number of views of the sequence as this factor must not exceed the number of views.

This operation is performed by skipping one of two points of view in the multiview set with N points of view resulting in a new set with $(N + 1)/2$ points of view. Table 4.2 shows the effect of subsampling operation for different factors on the number of views of in the tested sequences.

Table 4.2: *Number of views of multiview sets for each subsampling level.*

Sequence	Subsampling Factor								
	1	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**
San Miguel - UFRJ	769	385	193	97	49	25	13	7	4
Audi TT	513	257	129	65	33	17	9	5	3
San Miguel - UHasselt	193	97	49	25	13	7	4	-	-
Champagne Tower	65	33	17	9	5	3	-	-	-
Pantomime	65	33	17	9	5	3	-	-	-
Dog	65	33	17	9	5	3	-	-	-
Balloons	5	3	-	-	-	-	-	-	-
Kendo	5	3	-	-	-	-	-	-	-
Elephant	385	193	97	49	25	13	7	4	-
Train	385	193	97	49	25	13	7	4	-

Table 4.3 presents the maximum used subsampling factor for each multiview sequence.

Table 4.3: *Maximum chosen subsampling factor for multiview sequences.*

Sequence Name	Maximum Subsampling Level
San Miguel - UFRJ	256
Audi TT	256
San Miguel - UHasselt	64
Champagne Tower	32
Pantomime	32
Dog	32
Balloons	2
Kendo	2
Elephant	64
Train	64

Table 4.4 presents the corresponding angular density for each subsampling factor.

Table 4.4: *Angular density (views/degree) of multiview sets for each subsampling level.*

Model	**Subsampling Factor**								
	1	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**
San Miguel - UFRJ	11.1539	5.5770	2.7885	1.3943	0.6972	0.3486	0.1744	0.0874	0.0441
Audi TT	8.3983	4.1991	2.0996	1.0498	0.5249	0.2625	0.1314	0.0660	0.0336
San Miguel - UHasselt	2.6245	1.3122	0.6562	0.3281	0.1642	0.0823	0.0416	0.0217	-
Champagne Tower	1.0498	0.5249	0.2625	0.1314	0.0660	0.0336	-	-	-
Pantomime	1.0498	0.5249	0.2625	0.1314	0.0660	0.0336	-	-	-
Dog	1.0498	0.5249	0.2625	0.1314	0.0660	0.0336	-	-	-
Balloons	0.0921	0.0465	-	-	-	-	-	-	-
Kendo	0.0921	0.0465	-	-	-	-	-	-	-
Elephant	6.0363	3.0181	1.5091	0.7546	0.3773	0.1888	0.0946	0.0477	-
Train	6.5611	3.2806	1.6403	0.8202	0.4101	0.2052	0.1028	0.0518	-

4.4.2 Adjustments on the test set

Before testing the multiview sequences described in Appendix A it is necessary to put all sequences under specific conditions that allow us to perform the proposed tests. Four main adjustments were made: discarding of video frames, cropping and view selection. These adjustments were made depending on the sequence in question.

The list below explains how each of these adjustments was made:

1. **Discarding of video frames:**

    - *Motivation:* As this work will focus on static multiview compression, only one frame per viewpoint is necessary;
    - *Method:* The first frame (frame 0) of each viewpoint was kept while the subsequent frames were discarded;
    - *Sequences:* Champagne Tower, Pantomime, Dog, Balloons and Kendo, as those five sequences contain more than one frame per view.

2. **Cropping:**

    - *Motivation:* Encoders do not always accept every possible image dimensions. Hence, adjustments on the sequences to an acceptable input size are needed;

- *Method:* Cropping the sequence to the closest acceptable input format (see Table 4.5). The size of cropped part on the top of each viewpoint was equal to the size of the cropped part on the bottom of it. Similarly, the size of cropped part on the left of each viewpoint was equal to the size of the cropped part on the right of it;
- *Sequences:* Train and Elephant.

Table 4.5: *Original and cropped size for Elephant and Train sequences*

Sequence Name	**Original Resolution**	**Cropped Resolution**
Elephant	1280 × 853	1280 × 720
Train	1255 × 473	1216 × 448

3. **GOV size:**

- *Motivation:* In this work, we are also interested on how the size of the prediction structure affects the multiview compression. This size can be parametrized by the GOV size. The GOV size (M) in this work can assume the following powers of two:

$$M = \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$$

- *Method:* In order to divide the sequence in an integer number of GOVs, the number of views of each sequence was reduced to the closest power of two number M. It was made by equally discarding views in the beginning and in the end of the sequence. Then, the number he number of viewpoints N_{views} obtained for each sequence was:

$$N_{views} = max\{M\} + 1$$

  For example, the sequence *Pantomime* has 80 views and its maximum GOV size is 64. Then, the first eight and the last eight points of view were excluded from the sequence, making the *Pantomime* work with only 64 views.

  The list of the chosen number of viewpoints can be seen in the Table 4.6;

- *Sequences:* All multiview sequences.

Table 4.6: *Chosen viewpoints for each mutiview sequence*

Sequence Name	Chosen Views	Number of Views	Maximum GOV size
San Miguel - UFRJ	169 to 682	513	512
Audi TT	64 to 577	513	512
San Miguel - UHasselt	4 to 197	193	128
Champagne Tower	7 to 72	65	64
Pantomime	7 to 72	65	64
Dog	7 to 72	65	64
Balloons	2 to 7	5	4
Kendo	2 to 7	5	4
Elephant	38 to 423	256	128
Train	58 to 443	256	128

4.4.3 View Synthesis

For the view synthesis to be performed on the *multiview coding with redundancy removal* described on Chapter 3, it will be used an interpolation algorithm described in the Content Adaptive Wyner-Ziv Video Coding called IST-MCFI (Motion Compensation Frame Interpolation from Instituto Superior Técnico, Portugal) proposed by [17].

This algorithm was chosen over other view synthesis algorithms due to the lack of depth maps for all sequences.

The IST-MCFI frame interpolation uses information from two adjacent frames in the sequence (called Key-frames), to generate M frames (called Wyner-Ziv frames) between them. An interpolated frame Z between two frames X_i and X_{i+1} is estimated using the following steps:

1. **Average interpolation:** The first approximation of Z is to compute the average of X_i and X_{i+1};

2. **Forward Motion Estimation:** This step searches vectors which estimate the motion between X_i and X_{i+1}. A low pass filter is applied first to both key-frames in order to avoid bad motion vector candidates.

3. **Bidirectional Motion Estimation:** Improves the motion vectors accuracy from the forward motion estimation by using a bidirectional motion estimation (weighed estimations from frames X_i and X_{i+1}) better handling covered and uncovered regions of Z.

4. **Spatial Motion Smoothing:** This step removes bad candidates from the motion estimation steps.

These steps enumerate how the algorithm estimates only one view. The interpolator can estimate M views between X_i and X_{i+1}, repeating the same steps but replacing a key-view by Z or other subsequent interpolated views (see Figure 4.8). One example of this algorithm in action can be seen in Figure 4.9.

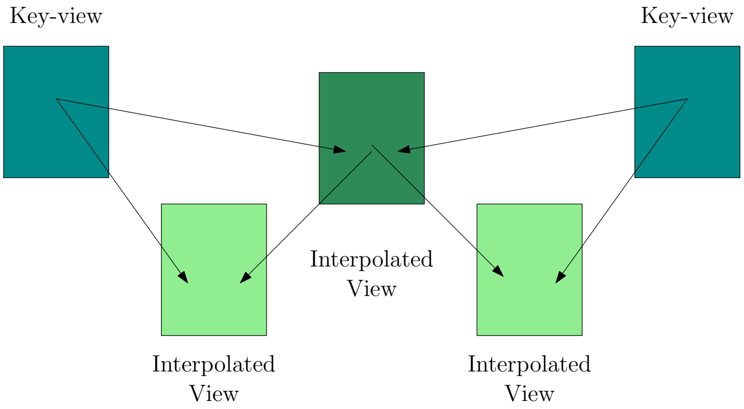
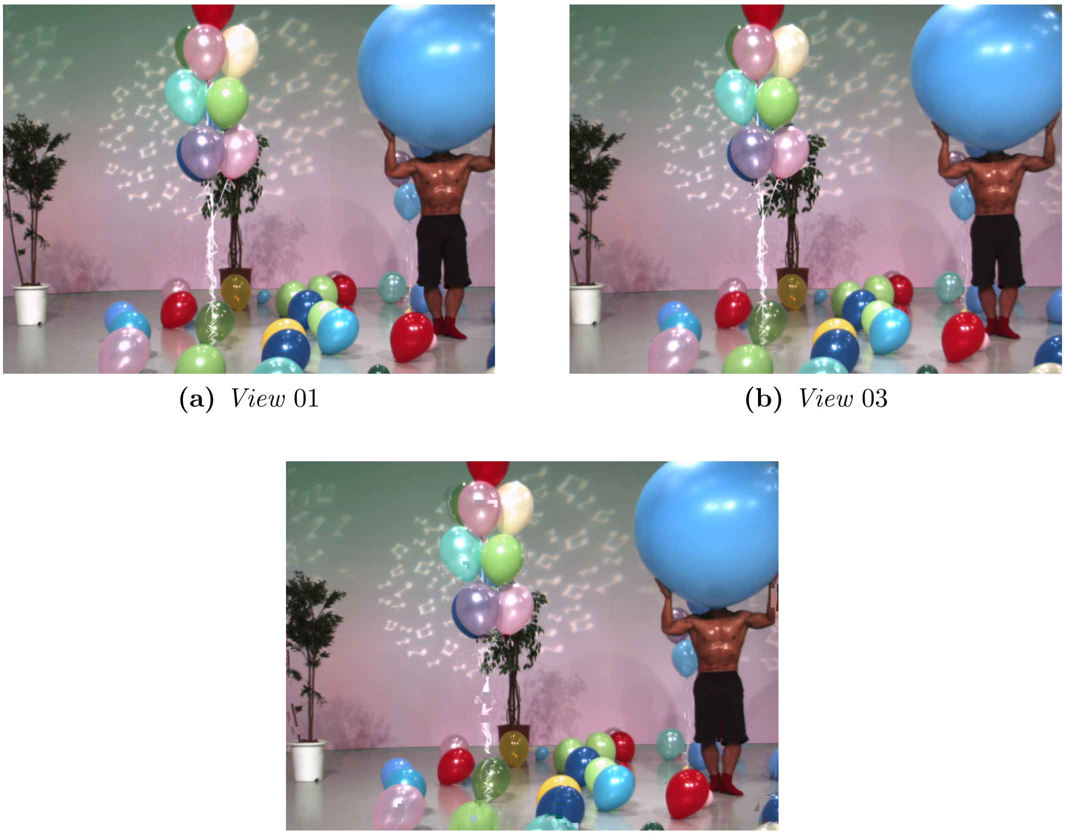


Figure 4.8: *Structure for view interpolation where $M = 4$.*



(a) *View* 01

(b) *View* 03

(c) *Interpolated view*

Figure 4.9: *Example of interpolation of views 01 and 03 of Balloons sequence using the IST-MCFI software.*

In this work, as explained in previous sections, it is only used subsampling factors that are powers of two. Therefore, all interpolation operations can be reduced to the usage of successive interpolations by two. For example, an interpolation of eight frames comes down to be the interpolation by two applied thrice.

4.5 Performance evaluation

4.5.1 Rate-distortion performance

In order to compare two different coding solutions in terms of compression efficiency it is necessary to find a good quality metric for comparison between the original and coded sequences. Alongside with that, computing the differences in bitrate will provide a good idea of the compression performance. The next paragraphs will explore the Peak Signal to Noise Ratio (PSNR) which is the most used metric in multiview compression literature.

Quality measure

The Peak Signal-to-Noise Ratio (PSNR) is a metric based on a ratio between the maximum possible coding error in an image and the actual coding error. The PSNR value for an 8-bit image is then given by

$$PSNR = 10 \cdot \log_{10}\left(\frac{255^2}{MSE}\right), \tag{4.3}$$

where the MSE, an error estimator called mean squared error, computed between the original $M \times N$ image $I(x, y)$ and the reconstructed image $I'(x, y)$ which was subjected to the codec, is given by:

$$MSE = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(I(x,y) - I'(x,y)\right)^2. \tag{4.4}$$

Rate measure

The usual choice for rate measurement for the output bitstream of a single picture is the total amount of bits that can represent a single pixel ($bits/pixel$). For multiview coding purposes, this choice might not be so convenient. The reason is that one can have two sequences with the same output bitrate in $bits/pixel$ but one sequence has a larger angular density and therefore much more redundancy. However, if an angular density factor is included, one can correct this fault. A convenient unit in this case may be $bits/pixel/degree$, where the $1/degree$ represents the angular density in the rate formula.

Rate-distortion curve

With a set of different resulting quality and rate measurements of a coding process, one can plot the so called *rate-distortion curve* or *RD-curve*. This curve illustrates the trade-off between rate and quality and also provides a basis for performance comparison between different experiments.

4.5.2 Bjøntegaard delta comparisson

A metric to compare two distinct coding methods or choices of parameters is necessary to compare different solutions. In this work, a metric known as *Bjøntegaard delta*(BD)[24] will be used.

In this methodology the rate-distortion points are fitted to a cubic curve and then the area between two curves A and B is computed in order to measure the "distance" between them. This distance, if the integral (Δ_{rate}) is taken along the rate-axis, measures how much rate A has saved in comparison with B in the same quality conditions. For example, a Δ_{rate} of -50 indicates that codec A has saved half of the bits or 50% in comparison with codec B. Similarly, if the integral ($\Delta_{quality}$) is taken along the quality-axis, it measures how much betteris the quality of A in comparison with B in the same rate conditions. Figure 4.10 illustrates how this distance is calculated.

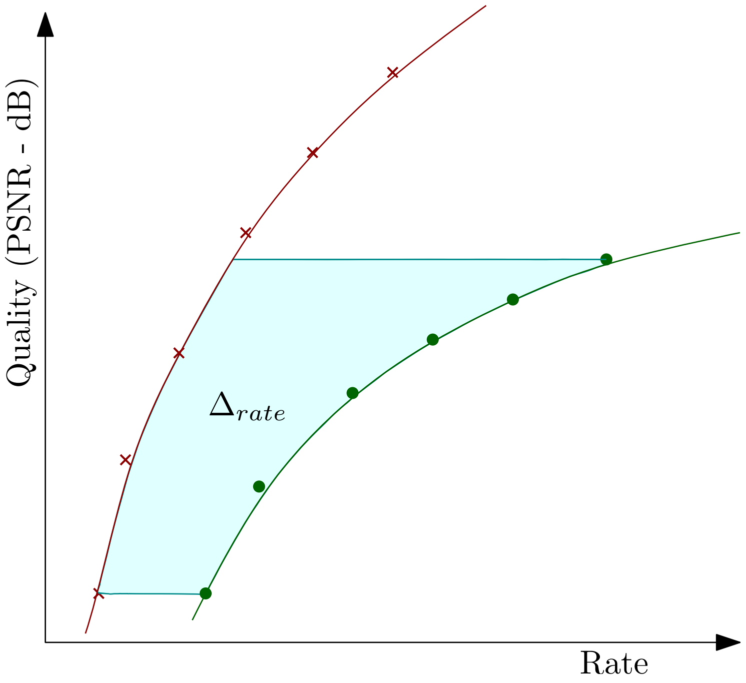


Figure 4.10: *Bjøntegaard delta for rate saving between two approximated cubic curves.*

All results in this chapter use the Bjøntegaard Delta to compare different coding solutions with the simulcast coding as explained in the previous chapter. In this work, all the rate values used for Bjøntegaard delta comparison are measured in *bits/pixel/degree*. As an example, if a coding solution A has the $\Delta_{rate} = -90$ compared with the simulcast result and a coding solution B has $\Delta_{rate} = -70$ also compared with simulcast, then, the coding solution A is the best, saving 20% more rate from simulcast then coding solution B.

4.5.3 Convex hull analysis

While the Bjøntegaard Delta informs how much a solution A is better than solution B in general, sometimes it is necessary to know which of the solutions is the best among the others for a specific rate or a specific quality level.

For this reason, it is necessary to use an algorithm that points out the solution which provides the best quality level when the rate is specified or that points out the solution which provides the lowest bitrate given a specific quality. These points that meet those characteristics can be obtained by a convex hull of the rate-distortion points [25].

A convex hull of RD points is the set of the most external points that can envelope the other points presenting the best compromise between rate and quality given the bitrate. Figure 4.11 illustrates a convex hull extracted from a set of points.

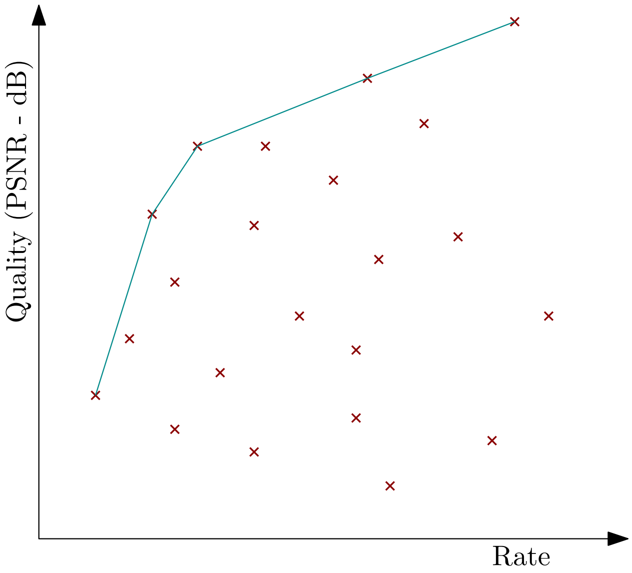


Figure 4.11: *Convex hull (blue) for a set of rate-distortion points.*

In conclusion, the best rate-distortion point to be found for a given quality, for example, will be the point that is the closest to the convex hull for that quality.

Having defined all necessary concepts, one can set-up and execute the experiments proposed in Chapter 3. The analysis and results of these experiments are described in the next chapter.

Chapter 5

Results

This chapter present the results for the experiments proposed in Chapter 3 under the conditions and methodologies established on Chapter 4.

5.1 Simulcast Coding

After submitting all the multiview sequences to the HEVC encoder and coding each view independently, one can calculate the mean PSNR of the decoded views and the total bitrate of the sequence and pair them in a rate-distortion point. In total there are eight bitrate-quality pairs (one pair for each QP value) per tested sequence. The resulting rate-distorion curves for each multiview sequence can be seen in Figure 5.1.

Observing the rate-distortion curves shown in Figure 5.1, one can notice a formation of two distinct groups of curves.

The top-left group represents basically the sequences with low angular density: Pantomime, Dog, Balloons, Champagne Tower and Kendo. Their low angular density causes the reduced bitrate and therefore the observed rate-distortion behaviour comparing with the other curves. Figure 5.2 presents separately these curves.

The bottom-right group of Figure 5.1 represents the sequences with high angular density. This group includes San Miguel- UFRJ, San Miguel - UHasselt, Train and Elephant sequences. As they have more views per degree, the bitrate of these curves tend to be greater than the others.

Apart from these groups, the Audi TT sequence, in spite of being a high angular density multiview sequences, has some characteristics of both groups. This fact occurs due to its intra-view redundancy provided mainly by the flat monotonic background.

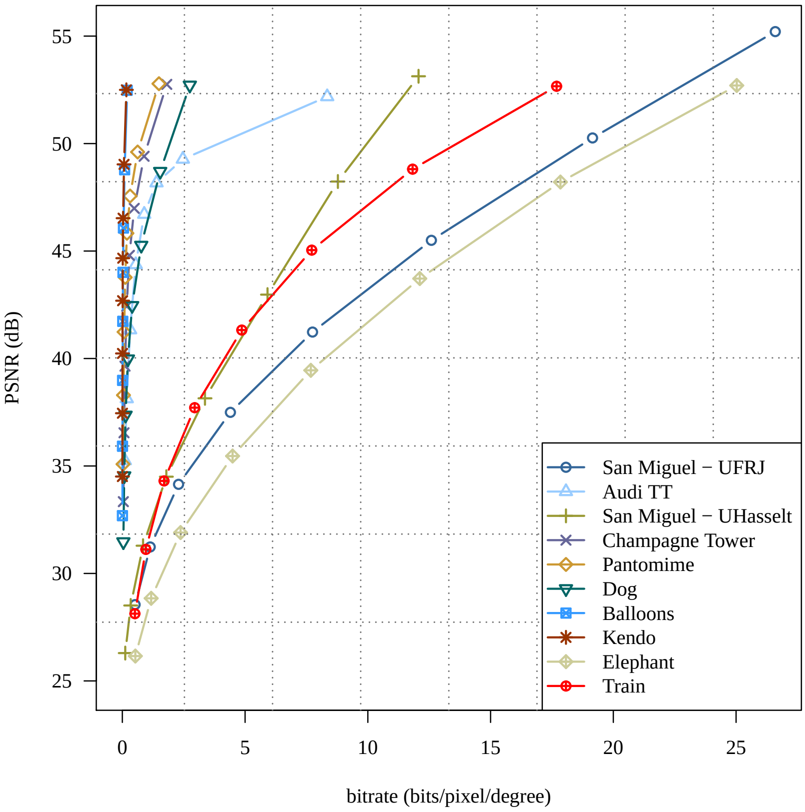


Figure 5.1: *Rate-distortion curve for simulcasting experiment.*

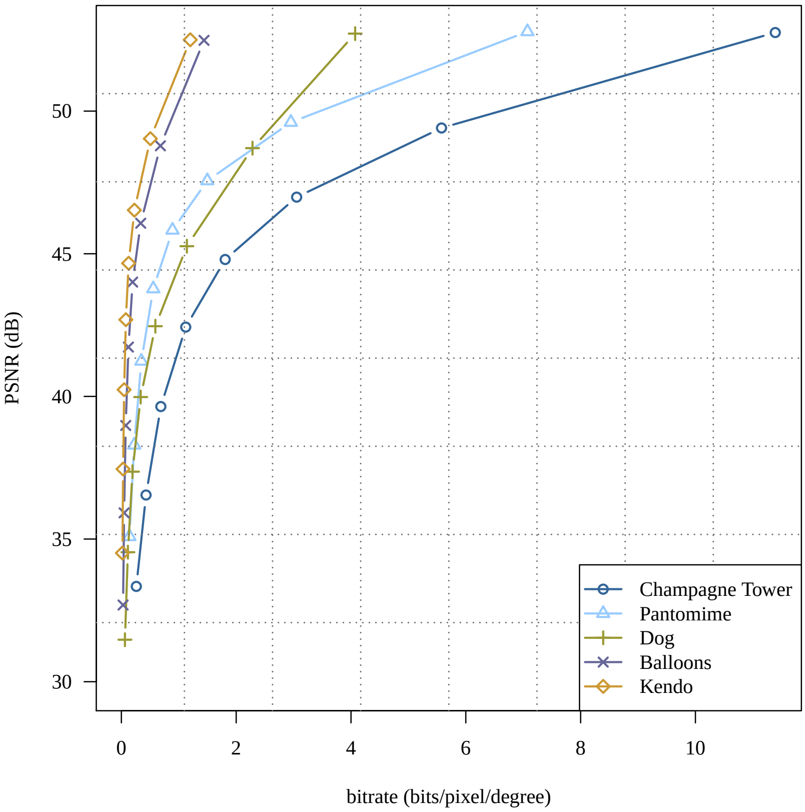


Figure 5.2: *Rate-distortion curve of the low density multiview sets for simulcasting.*

These results are the first attempt in multiview sequences compression. As no inter-view redundancy is explored in this experiment, these rate-distortion results can be used in the next experiment as a benchmark in order to show comparatively how much the inter-view reduction can enhance the multiview compression performance.

5.2 Multiview coding

The following tables present the results of encoding multiview sequences exploring the inter-view redundancy by using the conditions detailed in Subsection 4.4.1.

		GOV Size							
Subsampling factor	**GOV Structure**	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**
1	IPPPP	-83.788	-83.788	-83.788	-83.788	-83.788	-83.788	-83.788	-83.788
	IBBBI	-68.7135	-79.1208	-85.2994	-89.139	-91.7872	-93.5332	-94.4542	-94.9906
	IBBBP	-87.3502	-89.8475	-91.7118	-92.9939	-93.84	-94.4309	-94.8482	-95.1499
2	IPPPP	-81.3718	-81.3718	-81.3718	-81.3718	-81.3718	-81.3718	-81.3718	-
	IBBBI	-67.5994	-77.8493	-83.7811	-87.269	-89.7713	-91.0725	-91.7492	-
	IBBBP	-84.9994	-87.5343	-89.3812	-90.5548	-91.2629	-91.7327	-92.0253	-
4	IPPPP	-78.5733	-78.5733	-78.5733	-78.5733	-78.5733	-78.5733	-	-
	IBBBI	-66.9201	-76.4358	-81.6107	-84.6752	-86.4847	-87.34	-	-
	IBBBP	-82.2509	-84.6299	-86.25	-87.1152	-87.5721	-87.7948	-	-
8	IPPPP	-74.5294	-74.5294	-74.5294	-74.5294	-74.5294	-	-	-
	IBBBI	-65.0355	-73.3451	-77.6909	-79.803	-80.8322	-	-	-
	IBBBP	-77.9524	-80.0282	-81.1843	-81.5597	-81.5631	-	-	-
16	IPPPP	-69.2559	-69.2559	-69.2559	-69.2559	-	-	-	-
	IBBBI	-61.62	-68.4932	-71.6144	-72.3902	-	-	-	-
	IBBBP	-72.1449	-73.5591	-73.9834	-73.5249	-	-	-	-
32	IPPPP	-63.3032	-63.3032	-63.3032	-	-	-	-	-
	IBBBI	-57.3325	-62.2731	-63.4685	-	-	-	-	-
	IBBBP	-65.0118	-65.4475	-64.7222	-	-	-	-	-
64	IPPPP	-55.7619	-55.7619	-	-	-	-	-	-
	IBBBI	-51.4957	-53.7931	-	-	-	-	-	-
	IBBBP	-55.9997	-55.1208	-	-	-	-	-	-
128	IPPPP	-46.262	-	-	-	-	-	-	-
	IBBBI	-43.2912	-	-	-	-	-	-	-
	IBBBP	-44.8952	-	-	-	-	-	-	-

Table 5.1: *BD-Rate results for San Miguel sequence relative to the simulcasting encoding.*

		GOV Size							
Subsampling factor	**GOV Structure**	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**
1	IPPPP	-93.1256	-93.1256	-93.1256	-93.1256	-93.1256	-93.1256	-93.1256	-93.1256
	IBBBI	-73.768	-85.1673	-90.9664	-94.3355	-96.3779	-97.5072	-98.1066	-98.394
	IBBBP	-95.2386	-96.6409	-97.5234	-97.9884	-98.2826	-98.4605	-98.5504	-98.5584
2	IPPPP	-91.9476	-91.9476	-91.9476	-91.9476	-91.9476	-91.9476	-91.9476	-
	IBBBI	-73.0407	-84.2305	-90.3544	-93.6983	-95.7487	-96.8432	-97.3774	-
	IBBBP	-94.4771	-96.0261	-96.8212	-97.2621	-97.5436	-97.6847	-97.6982	-
4	IPPPP	-90.787	-90.787	-90.787	-90.787	-90.787	-90.787	-	-
	IBBBI	-71.7648	-83.4181	-89.4355	-92.6898	-94.6524	-95.6148	-	-
	IBBBP	-93.502	-94.8855	-95.6527	-96.0022	-96.2144	-96.2205	-	-
8	IPPPP	-89.6382	-89.6382	-89.6382	-89.6382	-89.6382	-	-	-
	IBBBI	-71.3761	-82.4791	-88.1963	-91.102	-92.7648	-	-	-
	IBBBP	-91.9081	-93.1589	-93.7617	-93.9193	-93.8609	-	-	-
16	IPPPP	-86.6113	-86.6113	-86.6113	-86.6113	-	-	-	-
	IBBBI	-69.6061	-80.2751	-85.4008	-87.6367	-	-	-	-
	IBBBP	-88.7588	-89.7356	-90.0599	-89.6109	-	-	-	-
32	IPPPP	-81.6649	-81.6649	-81.6649	-	-	-	-	-
	IBBBI	-66.7291	-76.2076	-80.1314	-	-	-	-	-
	IBBBP	-83.205	-83.7311	-83.087	-	-	-	-	-
64	IPPPP	-73.2677	-73.2677	-	-	-	-	-	-
	IBBBI	-61.5821	-68.9604	-	-	-	-	-	-
	IBBBP	-73.9076	-72.9858	-	-	-	-	-	-
128	IPPPP	-60.5453	-	-	-	-	-	-	-
	IBBBI	-52.9751	-	-	-	-	-	-	-
	IBBBP	-59.1188	-	-	-	-	-	-	-

Table 5.2: *BD-Rate results for Audi TT sequence relative to the simulcasting encoding.*

		GOV Size						
Subsampling factor	**GOV Structure**	**2**	**4**	**8**	**16**	**32**	**64**	**128**
1	IPPPP	-39.5447	-39.5447	-39.5447	-39.5447	-39.5447	-39.5447	-39.5447
	IBBBI	-25.1565	-39.2439	-46.814	-50.6651	-52.9774	-54.0343	-55.4802
	IBBBP	-45.2769	-49.1789	-51.7279	-53.2708	-54.3148	-55.0192	-56.1344
2	IPPPP	-30.8632	-30.8632	-30.8632	-30.8632	-30.8632	-30.8632	-
	IBBBI	-20.381	-31.5282	-36.4896	-38.0313	-38.9193	-40.2245	-
	IBBBP	-35.3513	-38.0022	-39.4158	-40.1258	-40.4185	-41.2504	-
4	IPPPP	-21.992	-21.992	-21.992	-21.992	-21.992	-	-
	IBBBI	-15.0751	-22.3429	-23.9575	-23.4597	-24.5956	-	-
	IBBBP	-24.3293	-25.4694	-25.8956	-25.6676	-26.1215	-	-
8	IPPPP	-12.6727	-12.6727	-12.6727	-12.6727	-	-	-
	IBBBI	-8.849	-11.4286	-10.2631	-9.9077	-	-	-
	IBBBP	-12.5924	-12.6887	-12.238	-12.0146	-	-	-
16	IPPPP	-4.3984	-4.3984	-4.3984	-	-	-	-
	IBBBI	-2.3034	-1.4365	-0.3898	-	-	-	-
	IBBBP	-3.1626	-2.5888	-2.2449	-	-	-	-
32	IPPPP	-0.2567	-0.2567	-	-	-	-	-
	IBBBI	2.181	2.6479	-	-	-	-	-
	IBBBP	1.5468	1.5654	-	-	-	-	-
64	IPPPP	-0.1003	-	-	-	-	-	-
	IBBBI	2.1094	-	-	-	-	-	-
	IBBBP	1.3635	-	-	-	-	-	-

Table 5.3: *BD-Rate results for San Miguel - UHasselt sequence relative to the simulcasting encoding.*

		GOV Size					
Subsampling factor	**GOV Structure**	**2**	**4**	**8**	**16**	**32**	**64**
1	IPPPP	-32.7343	-32.7343	-32.7343	-32.7343	-32.7343	-32.7343
	IBBBI	-21.8267	-33.2003	-38.7033	-40.7542	-42.2113	-43.6662
	IBBBP	-36.8355	-39.9749	-41.8626	-42.5182	-43.3903	-44.2635
2	IPPPP	-21.4927	-21.4927	-21.4927	-21.4927	-21.4927	-
	IBBBI	-14.9319	-22.7686	-25.0921	-25.1667	-26.7206	-
	IBBBP	-24.4472	-26.0134	-26.835	-27.074	-27.739	-
4	IPPPP	-13.258	-13.258	-13.258	-13.258	-	-
	IBBBI	-8.8422	-12.8543	-12.3263	-12.355	-	-
	IBBBP	-12.8826	-14.0237	-14.0027	-13.9108	-	-
8	IPPPP	-4.799	-4.799	-4.799	-	-	-
	IBBBI	-4.1525	-3.1636	-2.8249	-	-	-
	IBBBP	-4.2568	-3.9463	-4.0273	-	-	-
16	IPPPP	0.6105	0.6105	-	-	-	-
	IBBBI	2.3349	1.4475	-	-	-	-
	IBBBP	2.0807	1.1996	-	-	-	-
32	IPPPP	0.6334	-	-	-	-	-
	IBBBI	-0.6925	-	-	-	-	-
	IBBBP	-0.7105	-	-	-	-	-

Table 5.4: *BD-Rate results for Champagne Tower sequence relative to the simulcasting encoding.*

		GOV Size					
Subsampling factor	**GOV Structure**	**2**	**4**	**8**	**16**	**32**	**64**
1	IPPPP	-59.5295	-59.5295	-59.5295	-59.5295	-59.5295	-59.5295
	IBBBI	-30.8809	-51.5858	-63.9064	-71.2933	-76.0904	-77.6495
	IBBBP	-65.3586	-71.6223	-75.4125	-77.0494	-78.2981	-78.2617
2	IPPPP	-51.7065	-51.7065	-51.7065	-51.7065	-51.7065	-
	IBBBI	-28.4932	-47.2896	-58.0381	-64.1973	-66.6502	-
	IBBBP	-59.1094	-64.2971	-66.5598	-67.9954	-67.7546	-
4	IPPPP	-46.0712	-46.0712	-46.0712	-46.0712	-	-
	IBBBI	-25.7319	-42.8967	-51.2109	-53.2044	-	-
	IBBBP	-51.2947	-54.8957	-56.4837	-55.0886	-	-
8	IPPPP	-36.3637	-36.3637	-36.3637	-	-	-
	IBBBI	-22.8914	-34.5315	-36.8439	-	-	-
	IBBBP	-39.9051	-40.5873	-38.7169	-	-	-
16	IPPPP	-23.179	-23.179	-	-	-	-
	IBBBI	-14.7643	-19.4622	-	-	-	-
	IBBBP	-22.5865	-20.6354	-	-	-	-
32	IPPPP	-8.5723	-	-	-	-	-
	IBBBI	-5.9263	-	-	-	-	-
	IBBBP	-5.7288	-	-	-	-	-

Table 5.5: *BD-Rate results for Pantomime sequence relative to the simulcasting encoding.*

		GOV Size					
Subsampling factor	**GOV Structure**	**2**	**4**	**8**	**16**	**32**	**64**
1	IPPPP	-56.6951	-56.6951	-56.6951	-56.6951	-56.6951	-56.6951
	IBBBI	-29.0687	-49.1883	-61.241	-69.7013	-75.0469	-77.3706
	IBBBP	-62.8929	-68.8944	-73.2673	-75.9359	-77.6016	-78.42
2	IPPPP	-49.7253	-49.7253	-49.7253	-49.7253	-49.7253	-
	IBBBI	-27.1061	-44.7062	-56.4335	-62.8226	-66.1925	-
	IBBBP	-56.4682	-62.0022	-65.3535	-67.1807	-68.0314	-
4	IPPPP	-43.1861	-43.1861	-43.1861	-43.1861	-	-
	IBBBI	-23.876	-41.4084	-49.8879	-52.8626	-	-
	IBBBP	-49.2939	-53.8254	-55.8193	-55.8927	-	-
8	IPPPP	-35.6711	-35.6711	-35.6711	-	-	-
	IBBBI	-23.0232	-34.7971	-38.465	-	-	-
	IBBBP	-40.2768	-42.1802	-42.1012	-	-	-
16	IPPPP	-24.1715	-24.1715	-	-	-	-
	IBBBI	-15.9251	-22.4176	-	-	-	-
	IBBBP	-25.6257	-26.1973	-	-	-	-
32	IPPPP	-11.606	-	-	-	-	-
	IBBBI	-8.099	-	-	-	-	-
	IBBBP	-12.0953	-	-	-	-	-

Table 5.6: *BD-Rate results for Dog sequence relative to the simulcasting encoding.*

		GOV Size	
Subsampling factor	**GOV Structure**	**1**	**2**
1	IPPPP	-41.825	-41.825
	IBBBI	-24.0399	-36.1401
	IBBBP	-45.0033	-46.1075
2	IPPPP	-28.4362	-
	IBBBI	-16.1737	-
	IBBBP	-29.5718	-

Table 5.7: *BD-Rate results for Balloons sequence relative to the simulcasting encoding.*

		GOV Size	
Subsampling factor	**GOV Structure**	**1**	**2**
1	IPPPP	-41.6034	-41.6034
	IBBBI	-24.9083	-36.76
	IBBBP	-45.3551	-46.468
2	IPPPP	-28.3957	-
	IBBBI	-17.183	-
	IBBBP	-29.7518	-

Table 5.8: *BD-Rate results for Kendo sequence relative to the simulcasting encoding.*

		GOV Size							
Subsampling factor	**GOV Structure**	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**
1	IPPPP	-88.3419	-88.3419	-88.3419	-88.3419	-88.3419	-88.3419	-88.3419	-88.3419
	IBBBI	-49.3371	-73.7297	-85.6675	-91.6012	-94.672	-96.1789	-96.8706	-97.0971
	IBBBP	-92.4192	-94.4043	-95.7483	-96.5593	-97.028	-97.267	-97.3198	-97.266
2	IPPPP	-85.5311	-85.5311	-85.5311	-85.5311	-85.5311	-85.5311	-85.5311	-
	IBBBI	-49.1206	-72.6391	-84.1402	-89.7613	-92.6909	-94.0446	-94.4916	-
	IBBBP	-89.3383	-91.8523	-93.4553	-94.3614	-94.8219	-94.9259	-94.8242	-
4	IPPPP	-79.7839	-79.7839	-79.7839	-79.7839	-79.7839	-79.7839	-	-
	IBBBI	-47.477	-70.0322	-80.8538	-86.0341	-88.6809	-89.5807	-	-
	IBBBP	-84.5798	-87.5468	-89.3146	-90.182	-90.3946	-90.2325	-	-
8	IPPPP	-71.8798	-71.8798	-71.8798	-71.8798	-71.8798	-	-	-
	IBBBI	-44.3547	-65.1376	-74.8079	-79.1338	-80.8586	-	-	-
	IBBBP	-77.2907	-80.4307	-82.0866	-82.3673	-82.0765	-	-	-
16	IPPPP	-61.0786	-61.0786	-61.0786	-61.0786	-	-	-	-
	IBBBI	-38.9095	-56.9286	-64.5527	-66.8681	-	-	-	-
	IBBBP	-66.5117	-69.2756	-69.8108	-68.9977	-	-	-	-
32	IPPPP	-47.7388	-47.7388	-47.7388	-	-	-	-	-
	IBBBI	-31.1209	-44.8585	-49.0799	-	-	-	-	-
	IBBBP	-52.0668	-53.0427	-51.7757	-	-	-	-	-
64	IPPPP	-32.8902	-32.8902	-	-	-	-	-	-
	IBBBI	-21.6593	-29.6697	-	-	-	-	-	-
	IBBBP	-34.0858	-33.0005	-	-	-	-	-	-
128	IPPPP	-17.3925	-	-	-	-	-	-	-
	IBBBI	-11.9767	-	-	-	-	-	-	-
	IBBBP	-16.0816	-	-	-	-	-	-	-

Table 5.9: *BD-Rate results for Elephant sequence relative to the simulcasting encoding.*

		GOV Size							
Subsampling factor	**GOV Structure**	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**
1	IPPPP	-88.3925	-88.3925	-88.3925	-88.3925	-88.3925	-88.3925	-88.3925	-88.3925
	IBBBI	-49.7722	-73.3749	-85.1558	-91.1241	-94.3264	-95.9457	-96.6141	-96.7732
	IBBBP	-92.1959	-94.2532	-95.5004	-96.2707	-96.7224	-96.9192	-96.9169	-96.8671
2	IPPPP	-84.9295	-84.9295	-84.9295	-84.9295	-84.9295	-84.9295	-84.9295	-
	IBBBI	-48.4786	-71.7736	-83.3276	-89.1509	-92.3185	-93.6432	-93.96	-
	IBBBP	-89.0201	-91.4356	-92.9844	-93.861	-94.251	-94.2532	-94.1608	-
4	IPPPP	-79.5946	-79.5946	-79.5946	-79.5946	-79.5946	-79.5946	-	-
	IBBBI	-46.7575	-69.149	-80.1754	-85.6897	-88.2427	-88.8683	-	-
	IBBBP	-84.1649	-87.0643	-88.7475	-89.4459	-89.4468	-89.261	-	-
8	IPPPP	-71.6565	-71.6565	-71.6565	-71.6565	-71.6565	-	-	-
	IBBBI	-43.6611	-64.6362	-74.7941	-78.9134	-80.1224	-	-	-
	IBBBP	-76.9673	-79.986	-81.3234	-81.2327	-80.8684	-	-	-
16	IPPPP	-61.1049	-61.1049	-61.1049	-61.1049	-	-	-	-
	IBBBI	-38.6957	-57.4733	-64.9373	-66.3554	-	-	-	-
	IBBBP	-66.3457	-68.6697	-68.667	-67.7077	-	-	-	-
32	IPPPP	-47.3594	-47.3594	-47.3594	-	-	-	-	-
	IBBBI	-32.0688	-45.3947	-48.3545	-	-	-	-	-
	IBBBP	-50.9922	-51.0059	-49.9585	-	-	-	-	-
64	IPPPP	-30.1814	-30.1814	-	-	-	-	-	-
	IBBBI	-21.613	-27.6095	-	-	-	-	-	-
	IBBBP	-29.9538	-29.3414	-	-	-	-	-	-
128	IPPPP	-11.9282	-	-	-	-	-	-	-
	IBBBI	-9.6315	-	-	-	-	-	-	-
	IBBBP	-11.2048	-	-	-	-	-	-	-

Table 5.10: *BD-Rate results for Train sequence relative to the simulcasting encoding.*

Tables 5.1 to 5.10 show that inter-view prediction indeed can reduce the total bitrate of a multiview sequence without subsampling over the simulcast coding. It can save from 20% to 50% more bits than simulcasting (for the same quality level) in low-density multiview sequences (Kendo and Balloons) and up to 98% in in high-density sequences, like San Miguel – UFRJ and Audi TT.

From theses results one can analyse some performance behaviours according to the main tested parameters: structure, in order to find out which GOV structure is a good encoding choice; angular density through subsampling factor, to find out how the sequence would behave with a different view density and which GOV size can perform a better compression and in which case. Figure 5.3 shows the best result from each subsampling factor for each multiview sequence. The next paragraphs analyse the results from Tables 5.1 to 5.10 and Figure 5.3.

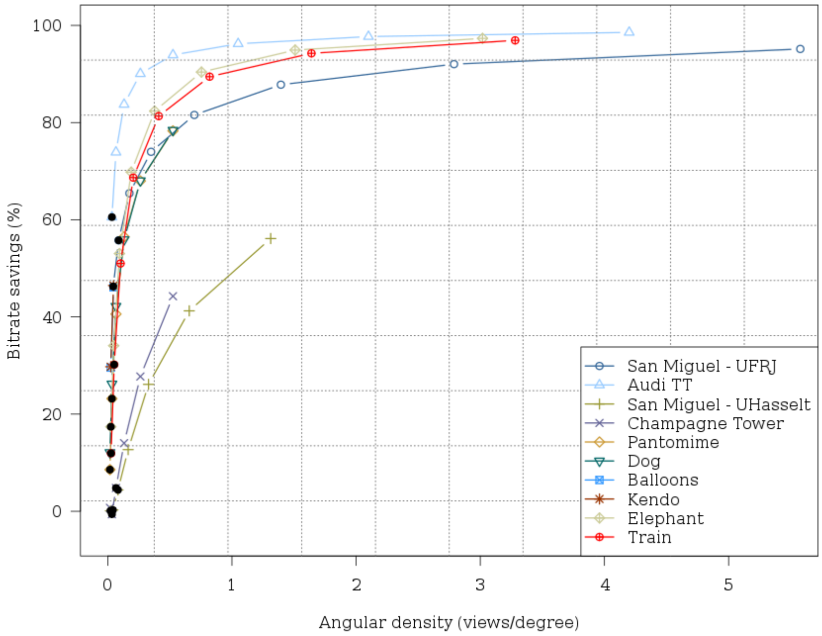


Figure 5.3: *Bitrate savings curves according to the angular density of the sequence. Each point in the curve is the coding best result of a subsampling factor. Black dots point out the results where IPPPP structure performed better than the IBBBP structure.*

GOV structure considerations

One important fact concerning the inter-view prediction structures is how much the information from adjacent pictures can improve the coding performance over not using this information at all, as in the simulcast experiment. For example, the IPPPP structure, as it uses mostly P-views, relies less on the other views redundancy using only the information carried by the previously encoded view. On the other hand, IBBBP and IBBBI structures use information from more than one view, from both sides. However, the reference views, depending on the GOV size can be located many views away from the one to be encoded. Taking these facts into consideration, it is possible to look at the results regarding the GOV structures.

From Tables 5.1 to 5.10 and Figure 5.3, one can reach some conclusions concerning GOV structures:

- The IBBBP structure, according to this experiment, is the best choice, among the tested structures, whenever the angular density is above 0.2 views per degree. It happens because the spatial correlation between two views, when they are placed far from each other can be smaller than the correlation between two adjacent views. In this case, choosing only the adjacent view rather two others with the distance greater than two views can be the best option;

- Below the angular density of 0.2 views per degree, in some cases the IPPPP structure can outperform the IBBBP structure. This result shows that the less dense is the multiview sequence, one should choose a one-directional prediction rather than a bi-directional prediction;

- The IBBBI is always outperformed by the IBBBP structure. The fact that the second uses two intra-view prediction views instead of one intra-view and a P-view directly predicted from the I-view, affects the coding performance in favour of the IBBBP structure in all tested situations.

Subsampling considerations

Another important result is that the subsampling factor of a multiview sequence impacts directly on the bitrate savings. As the subsampling factor increases, the Bjontegaard Delta result points to a loss in bitrate savings for inter-view prediction. Some cases show that inter-view prediction may have a worse performance than simulcasting for sequences with high subsampling factor, for example, for San Miguel – UHasselt sequence with subsampling factors of 32 and 64.

The same difference in performance can be noticed comparing a high-density sequence with a low-density one: For example, *San Miguel - UFRJ* subsampled by 64 (angular density of 0.1744) result has an equilavent performance to *Pantomime* subsampled by 8 (angular density of 0.1314).

GOV Size /Angular density considerations

According to the results, inter-view prediction can reduce the total bitrate of a multiview sequence without subsampling over simulcast coding. It can save from 20% to 50% in low-density multiview sequences (*Kendo* and *Balloons*) and up to 94% to 97% in a high-density multiview sequences, like *San Miguel – UFRJ* and *Audi TT*.

One can notice that, usually, when the GOV size increases, the saving performance also increases. In some cases, however, the performance reaches a maximum level and starts to drop for large GOV sizes. This phenomenon occurs in every tested sequence except *Kendo* and *Balloons*.

If one considers only one subsampling factor of a multiview sequence and analyse the compression performance under the best structure (the IBBBP structure in most cases), the GOV size effects on compression performance can be isolated and analysed properly.

By isolating these cases when the IBBBP structure performance reaches a maximum level, an attempt was made to relate the maximum performance with the angular density of P-views.

In order to find out the angular density which provides the maximum compression performance, a cubic-spline regression was composed of BD-rate values of each subsampling factor and their respective angular densities between each P-view of the sequence. From the regressed curve, the point of maximum BD-rate could be found. This maximum is calculated using the maximum value and the two adjacent values on the table with their respective P-view angular densities, as exemplified in Figure 5.4

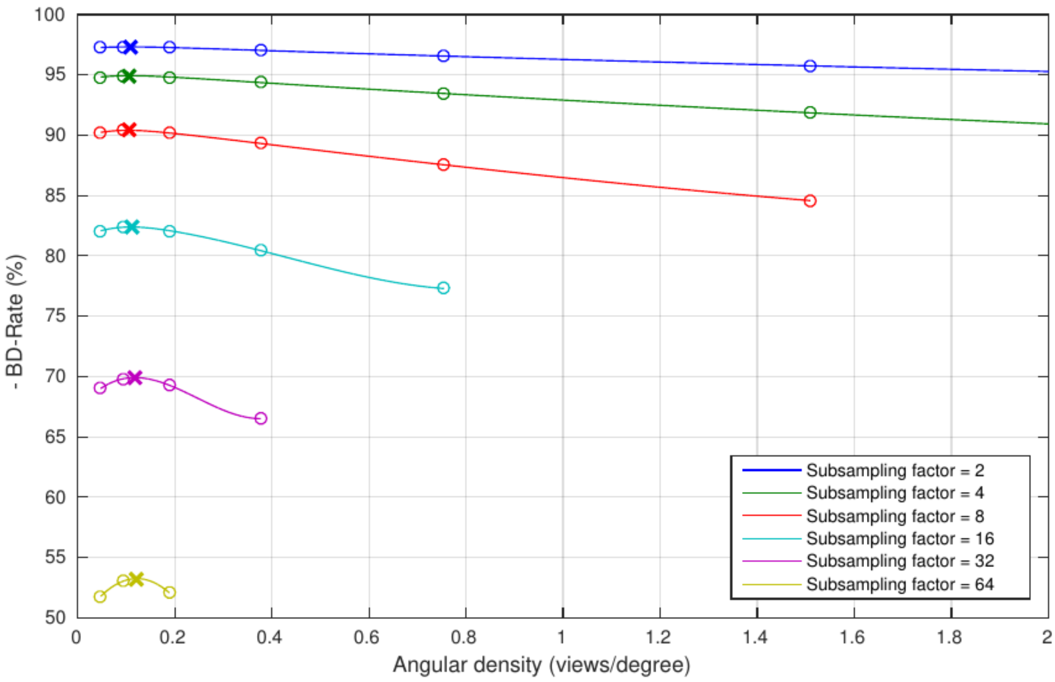


Figure 5.4: *Detail of BD-Rate values (circles) for the Elephant sequence regressed into cubic-spline curves. The maximum calculated values are marked with crosses.*

For every occurrence of an inflexion in BD-rate values, the maximum point is calculated and related to their respective P-views angular densities. Table 5.11 points out the average maximum values for each sequence.

From these results, one can notice that even though the test sequences have angular densities varying from 0.0921 to 11.1539 views per degree, the maximum compression performance is achieved when the angular density between P-views is around 0.027 to 0.111 (average of 0.0677 P-views/degree). Looking back at Figure 5.4, one can see that this range of densities is small when compared with the all range of angular densities tested in this work (up to 11.15 views/degree, according to Table 4.1).

It short, when encoding a multiview sequence using the IBBBP prediction structure, one could chose a GOV size which provides an angular density of P-views around 0.0677 for a better compression performance.

Sequence Name	**Angular density (P-views/degree)**
San Miguel - UFRJ	0.111090 ± 0.012038
Audi TT	0.077082 ± 0.008795
San Miguel - UHasselt	0.113099 ± 0.000000
Champagne Tower	0.053072 ± 0.000000
Pantomime	0.034128 ± 0.008025
Dog	0.027011 ± 0.000000
Balloons	-
Kendo	-
Elephant	0.056151 ± 0.002248
Train	0.077133 ± 0.003260

Table 5.11: *Average maximum BD-Rate values for multiview sequences according to their angular densities between P-views).*

5.3 Interpolation of encoded subsampled multiview sequences

This experiment has as purpose to evaluate the effect of the inter-view redundancy removal. This removal is performed by the subsampling operation, as detailed in Section 5.2, creating a reduced version of the sequence with lower angular density.

However, one cannot directly compare sequences with different subsampling factors as they have a different number of views but only comparing sequences under different encoding parameters.

In order to perform this comparison, one can apply an interpolator to the decoded sequence and match the number of views. The interpolation at the decoder can partially recover the loss in multiview smoothness, allowing subsampled sequences to be smaller in size of data and also look natural.

Some arguments in favour of this operation are:

- Angular density will not be harmed due to the interpolation at the decoder;

Table 5.12: *Best coding parameters for inter-view prediction.*

Multiview Sequence	Subsampling Factor	GOV Structure	GOV Size
San Miguel - UFRJ	2	IBBBP	256
	4	IBBBP	128
	8	IBBBP	64
	16	IBBBP	32
	32	IBBBP	8
	64	IBBBP	4
	128	IBBBP	2
	256	IPPPP	1
Audi TT	2	IBBBP	256
	4	IBBBP	128
	8	IBBBP	64
	16	IBBBP	16
	32	IBBBP	8
	64	IBBBP	4
	128	IBBBP	2
	256	IPPPP	1
San Miguel - UHasselt	2	IBBBP	64
	4	IBBBP	32
	8	IBBBP	4
	16	IPPPP	1
	32	IPPPP	1
	64	IPPPP	1
Champagne Tower	2	IBBBP	32
	4	IBBBP	4
	8	IPPPP	1
	16	IPPPP	1
	32	IBBBP	2
Pantomime	2	IBBBP	16
	4	IBBBP	8
	8	IBBBP	4
	16	IPPPP	1
	32	IPPPP	1
Pantomime	2	IBBBP	16
	4	IBBBP	8
	8	IBBBP	4
	16	IPPPP	1
	32	IPPPP	1
Dog	2	IBBBP	32
	4	IBBBP	16
	8	IBBBP	4
	16	IBBBP	4
	32	IBBBP	2
Balloons	2	IBBBP	2
Kendo	2	IBBBP	2
Elephant	2	IBBBP	64
	4	IBBBP	32
	8	IBBBP	16
	16	IBBBP	8
	32	IBBBP	4
	64	IBBBP	2
	128	IPPPP	1
Train	2	IBBBP	64
	4	IBBBP	32
	8	IBBBP	8
	16	IBBBP	4
	32	IBBBP	4
	64	IPPPP	1
	128	IPPPP	1

- To encode a subsampled version of the sequence can reduce drastically the coding time and memory consumption, as they are directly proportional to the number of images;
- For low subsampling factors the interpolation can be enough to mask the subsampling operation.

However, there are also some drawbacks:

- Depending on the application, an interpolation at the decoder can critically harm the user experience, if the application runs in real-time.
- In some cases, interpolation cannot handle object occlusions very well and can harm the total multiview experience.

Another important point of this experiment is that through it one can check-in whether HEVC is effective in exploring the redundancy with in a multiview sequence. This is so because if subsampling followed by HEVC coding followed by interpolation at the decoder has better coding efficiency than to apply HEVC coding directly to the sequence without subsampling, then a simple subsampling plus interpolation operation is better to exploit the redundancy among the dropped frames than HEVC. Then, another objective of this experiment is to verify for which angular resolutions is HEVC able to effectively exploit the interview redundancy.

In this experiment, the view synthesis algorithm described in 4.4.3 will be used.

From the experiment described in the 3, rate-distortion curves for each one of the subsampling factors will be compared using the Bjøntegaard Delta-Rate. However, for some subsampling factors, the interpolated sequences have presented a very low PSNR making impossible the calculation of the area between two curves and consequently preventing the Bjøntegaard Delta to be calculated. For example, the *Champagne Tower* sequence has presented RD-curves as seen in Figure 5.5.

This fact occurs because when the subsampling operation removes crucial information. It can also occur or due to difficulty presented to interpolate some type of content such as water or other reflexive objects. In this case, it is not possible to compute the Bjøntegaard Delta-Rate because there is not an effective area between the curves along the bitrate axis. In addition, the quality of the interpolated sequence is compromised by the badly interpolated views at the decoded sequence. It occurs mainly with the *Champagne Tower* and *San Miguel - UHasselt* sequences and also with sequences submitted to a high subsampling factor.

The cases where the Bjøntegaard Delta cannot be calculated will not be considered in this analysis once the interpolated sequence has corrupted views. The following images (Figures 5.6 to 5.13) exhibit only the valid curves.

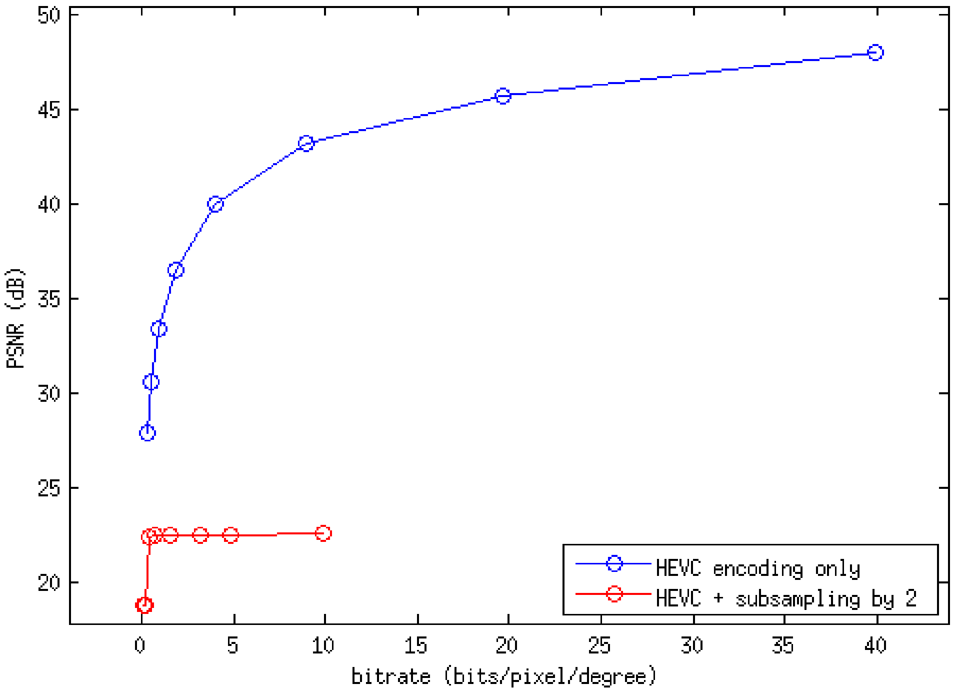


Figure 5.5: *Rate-distortion curves for a subsampled by two version of Champagne Tower sequence and interpolated at the HEVC decoder compared with the original sequence.*

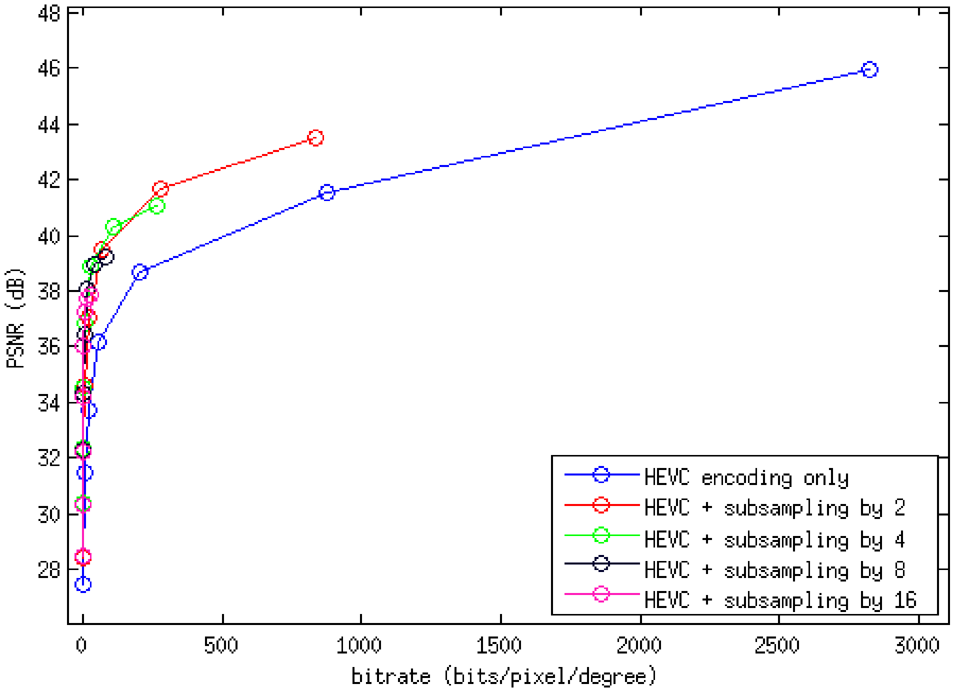


Figure 5.6: *Rate-distortion curves for subsampled versions of San Miguel - UFRJ sequences interpolated at the HEVC decoder.*

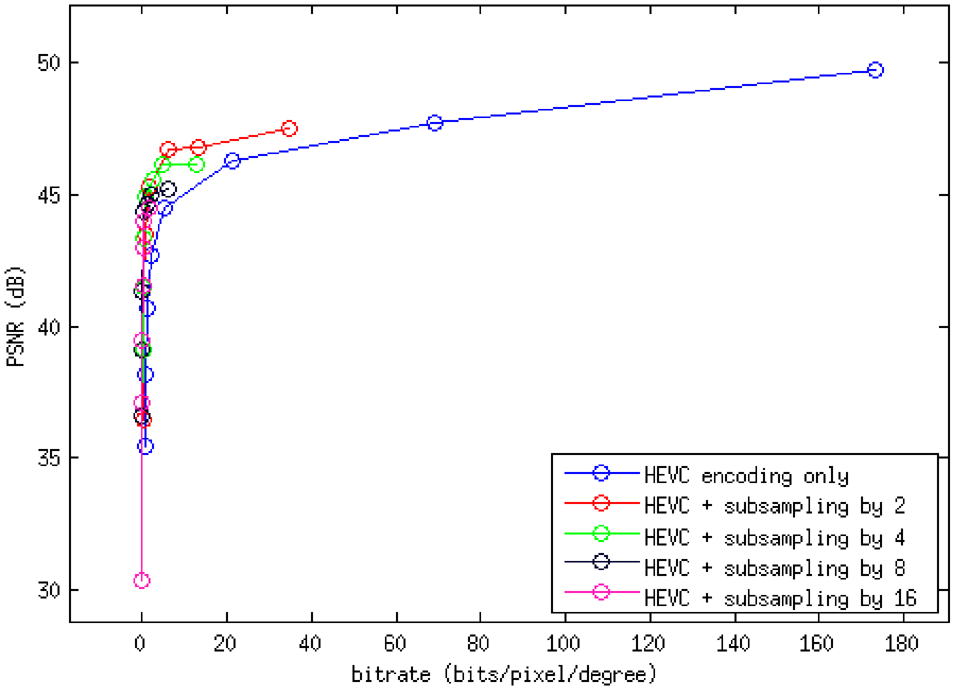


Figure 5.7: *Rate-distortion curves for subsampled versions of Audi TT sequences interpolated at the HEVC decoder.*

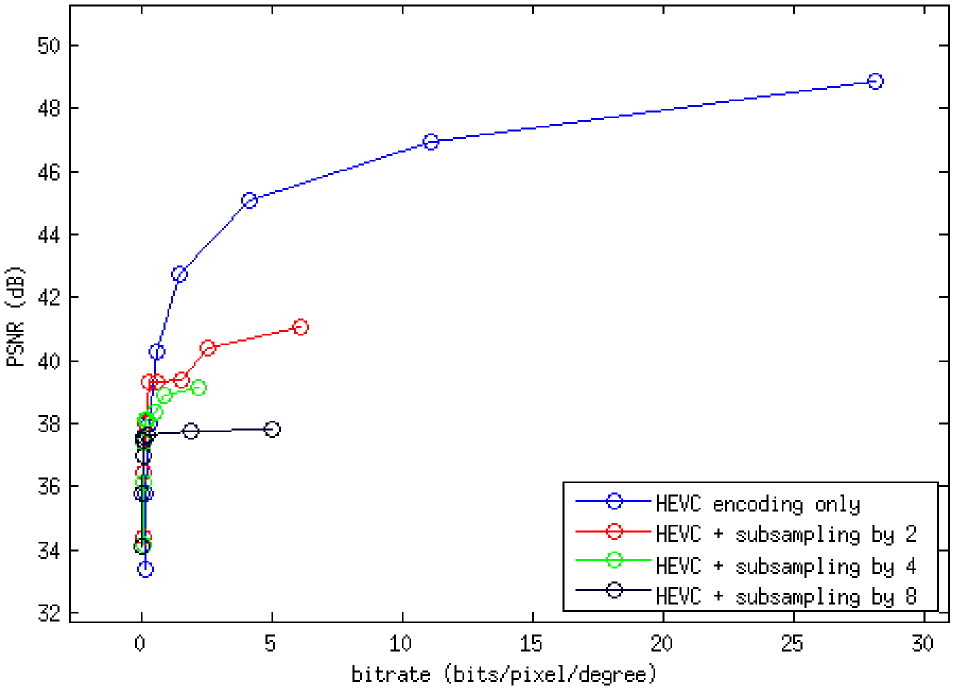


Figure 5.8: *Rate-distortion curves for subsampled versions of Pantomime sequences interpolated at the HEVC decoder.*

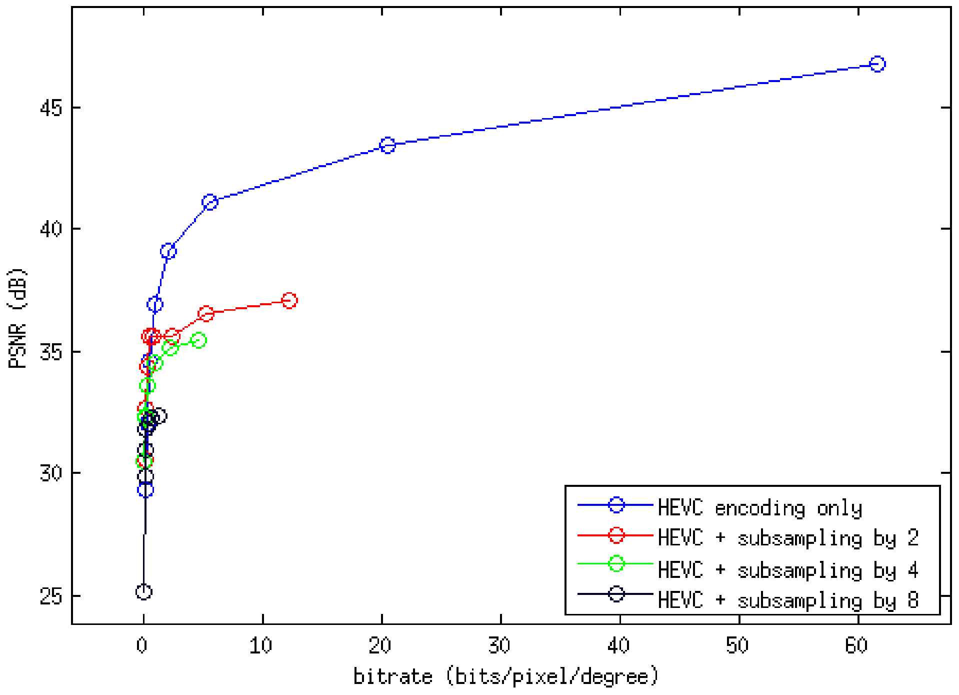


Figure 5.9: *Rate-distortion curves for subsampled versions of Dog sequences interpolated at the HEVC decoder.*

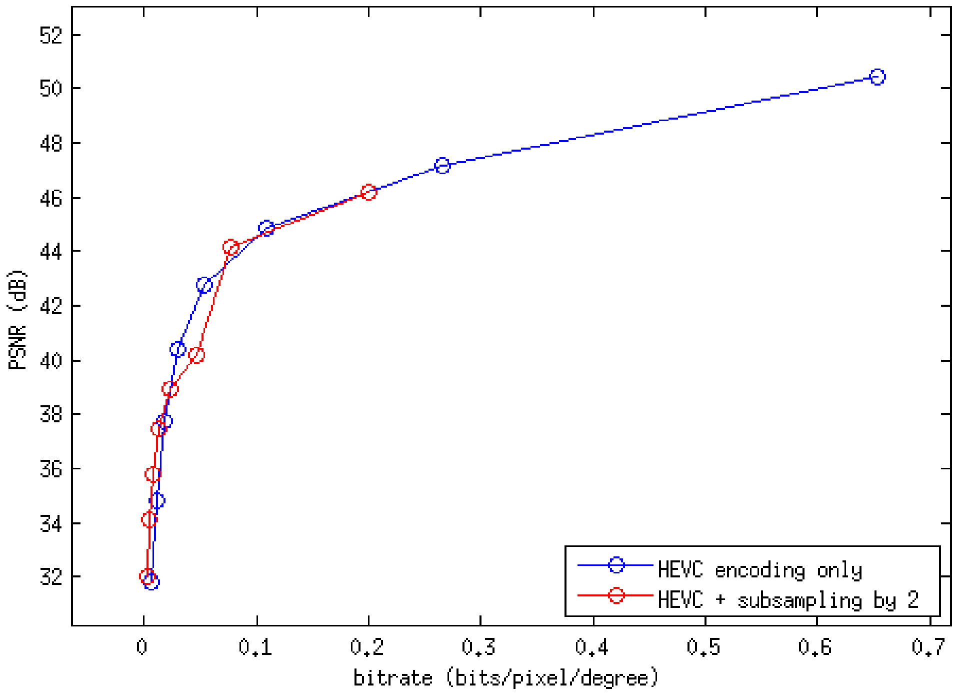


Figure 5.10: *Rate-distortion curves for subsampled versions of Balloons sequences interpolated at the HEVC decoder.*

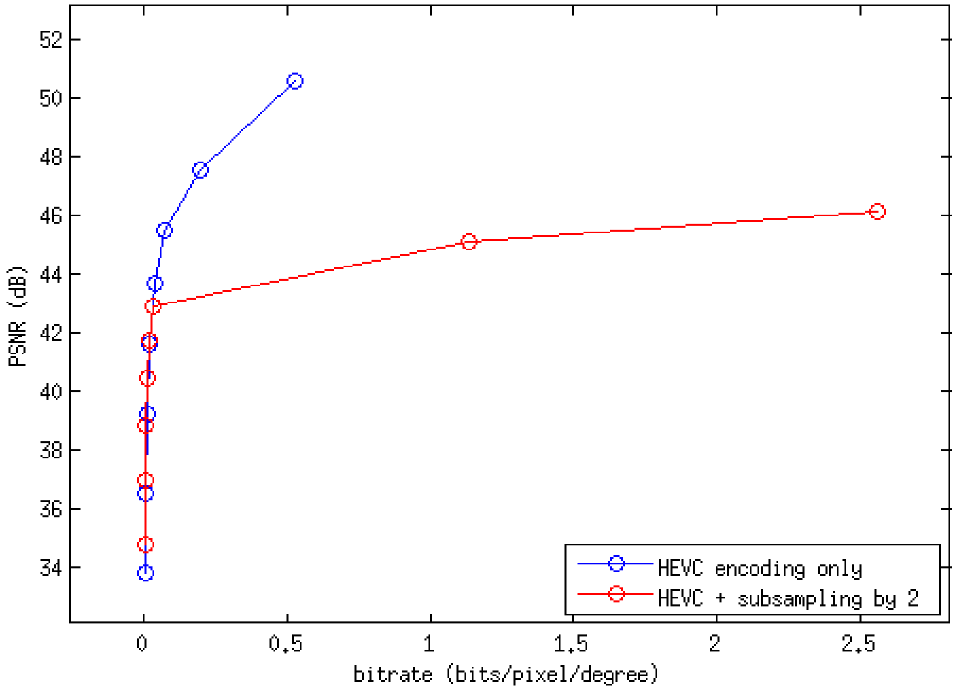


Figure 5.11: *Rate-distortion curves for subsampled versions of Kendo sequences interpolated at the HEVC decoder.*

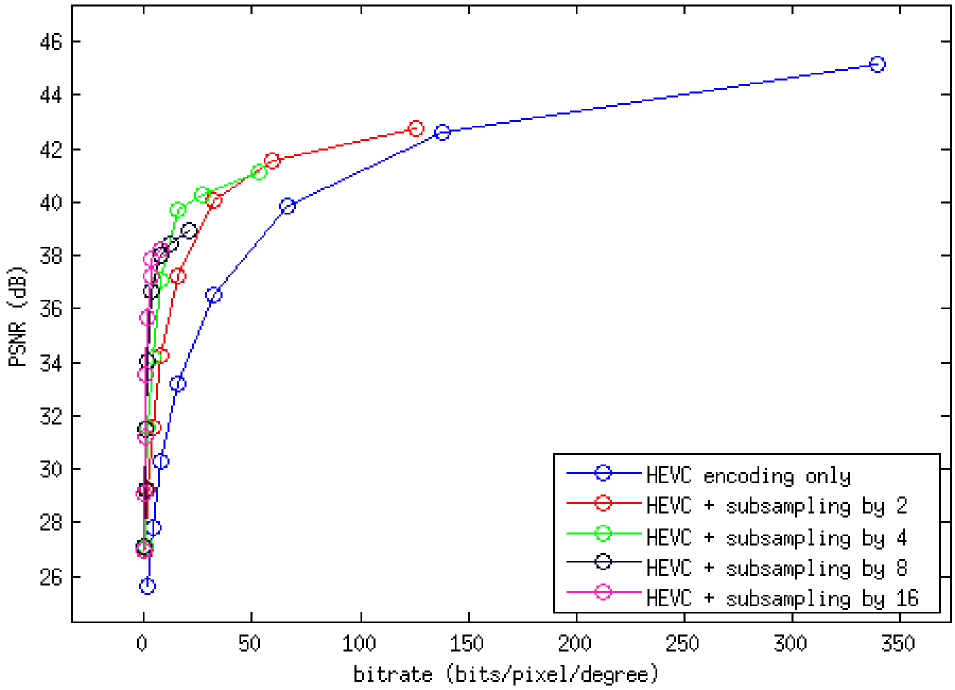


Figure 5.12: *Rate-distortion curves for subsampled versions of Elephant sequences interpolated at the HEVC decoder.*

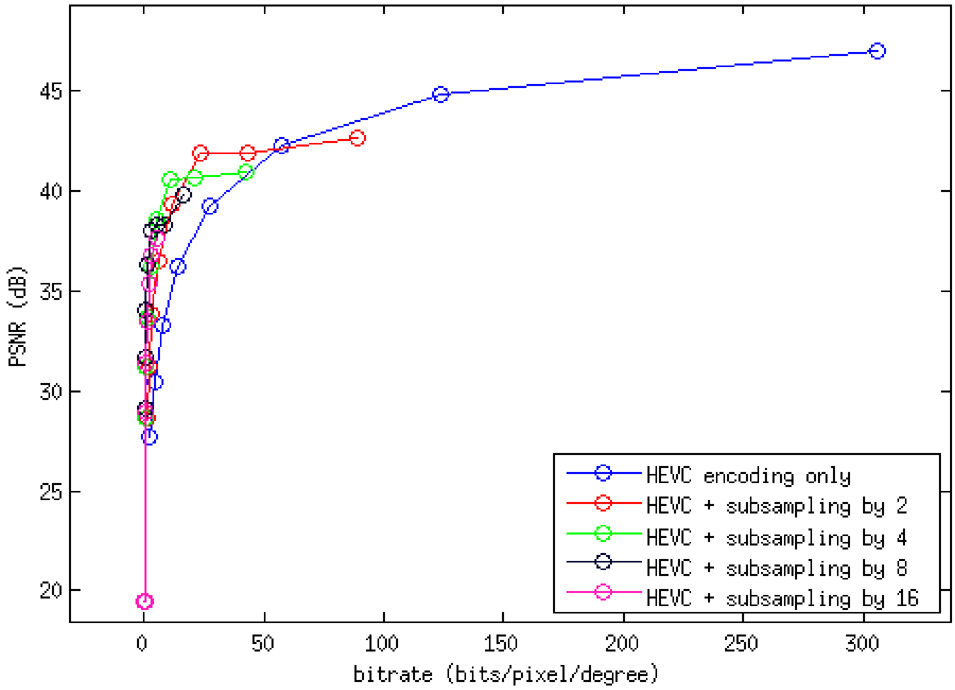


Figure 5.13: *Rate-distortion curves for subsampled versions of Train sequences interpolated at the HEVC decoder.*

Bjøntegaard Delta analysis

On the Table 5.13, one can compare the results from the inter-view prediction performance and the results from subsampled sequences interpolated at the decoder. This comparison is given in percentage of bits saved in comparison with the sequences encoded only with inter-view prediction.

As one can see from Table 5.13, by discarding views to the inter-view redundancy, one can save up to 90% more bits in certain cases. It is also noticiable that there is not a Bjøntegaard Delta for each proposed subsampling factor. In fact, the highest subsampling factor where the Bjø]ntegaard Delta is still valid was 16 from sequences with high angular density as *San Miguel - UFRJ*, *Audi TT*, *Elephant* and *Train*.

On the other hand, for each case where the Bjøntegaard Delta was valid, it is noticiable that the discarding views operation combined with the inter-view prediction can overperform the inter-view prediction of the original sequence.

The following paragraphs will analyse these cases by studying the sequences by groups in order to clarify the effects of subsampling on each sequence.

High angular density sequences

This group is composed by *San Miguel - UFRJ*, *Audi TT*, *Elephant* and *Train* sequences. Within this group, discarding views at the encoder can save at least 97% of the bits, with respect to the simulcast approach or 70% of the bits more

Table 5.13: *Comparison of coding performance between inter-view prediction and the interpolation of subsampled versions of the sequences. The results displayed below correspond only to the cases where the Bjøntegaard Delta was valid.*

Multiview Sequence	**Subsampling Factor**	**GOV Structure**	**GOV Size**	**Subsampling + Inter-view prediction**
San Miguel - UFRJ	2	IBBBP	256	-70.97469
	4	IBBBP	128	-84.92816
	8	IBBBP	64	-92.21771
	16	IBBBP	32	-96.03436
Audi TT	2	IBBBP	256	-70.8014
	4	IBBBP	128	-85.94478
	8	IBBBP	64	-92.73377
	16	IBBBP	16	-96.59286
San Miguel - UHasselt	2	IBBBP	64	161.7287
Pantomime	2	IBBBP	16	-35.48523
	4	IBBBP	8	-69.28838
	8	IBBBP	4	-84.35709
Dog	2	IBBBP	32	-29.29627
	4	IBBBP	16	-65.61928
	8	IBBBP	4	-78.89232
Balloons	2	IBBBP	2	5.644259
Kendo	2	IBBBP	2	-24.8888
Elephant	2	IBBBP	64	-24.68993
	4	IBBBP	32	-60.33033
	8	IBBBP	16	-79.97462
	16	IBBBP	8	-89.90357
Train	2	IBBBP	64	-16.06184
	4	IBBBP	32	-55.50946
	8	IBBBP	8	-78.25349
	16	IBBBP	4	-88.8751

than the plain HEVC coding. The solution of interpolating subsampled sequences always performs better than the inter-view prediction itself for each subsampling factor tested. One can conclude, then, that for high-angular density sequences, it is always worthy to reduce the inter-view redundancy first, by dropping views out, before encoding the sequence and performing a inter-view prediction.

Medium angular density sequences

This group comprises *San Miguel - UHasselt*, *Champagne Tower*, *Pantomime* and *Dog* sequences. The proposed solution, that is to interpolate subsampled sequences, have saved only 8.45% compared with the simulcast coding, or has spent $161,73\%$ more bits than the HEVC only solution for the *San Miguel - UHasselt* sequence. This low compression performance occurs due to the complexity of the contents on the scene. These contents, as water pouring from a fountain and transparent objects as crystal glasses, make it difficult to the interpolation process producing malformed views at the decoder.

The same results occur for the *Champagne Tower* sequence, in which reflexive objects as crystal glasses on the scene are responsible for low-PSNR interpolated views and invalid Bjøntegard delta values.

Nevertheless, the medium-angular density sequences without complex objects in scene, as *Pantomime* and *Dog*, can present some compression results close to the high-density sequences.

For this group of sequences, one can conclude that the subsampling plus interpolation results are highly dependent on the scene content. Depending on this, the compression performance of the sequence subsampling can fit either on the high angular density performance level or in the low angular density performance level.

Low angular density sequences

This group comprehends the two sequences with the lower angular density: *Balloons* and *Kendo*. As expected, discarding viewa in this case does not perform as good as in the other groups in a rate-distortion sense, once the inter-view redundancy is lower than the redundancy in the other groups, and then, the subsampling operation tends to eliminate important spatial information. This inter-view information loss reflects on the decoder resulting in a poorer interpolation. This is what happens at the *Balloons* sequence, where the interpolation cannot predict certain parts of the images, mainly the parts containing the balloons, because their reflection are highly dependent on the position of the camera.

However, for the *Kendo* sequence, discarding half of the views helps the inter-view prediction to save 13% more than usual. This result is not as good as the results obtained in other sequences, but it shows that even in low-density sequences, there is still a fair amount of inter-view redundancy to be eliminated.

Analysis of the rate-distortion curves using the convex hull

However, as explained in Subection 4.5.2, the Bjøntegaard Delta metric just compares curves using common PSNR points from both curves. Therefore, if two curves A and B have in common points between $30dB$ and $40dB$, the metric can only say "A performance is better than B between $30dB$ and $40dB$". RD-curves from Figures 5.6 to 5.13, show that just the high density sequences present RD-curves from low to high PSNR levels. The Pantomime sequence for example, can only be compared with the original sequence using four RD points.

This fact shows that the Bjøntegaard Delta does not always points to the best coding solution for all quality levels, or similarly, for all bitrate values. In order to find out which coding solution is the best given a bitrate is necessary to find the point which has the best rate-distortion compromise in an RD-plot. The set of points that meet those characteristics are part of a convex hull of the RD-points, as explained in Subsection 4.5.3. Figure 5.14 exemplifies the convex hull points for the *San Miguel - UFRJ* rate-distortion results.

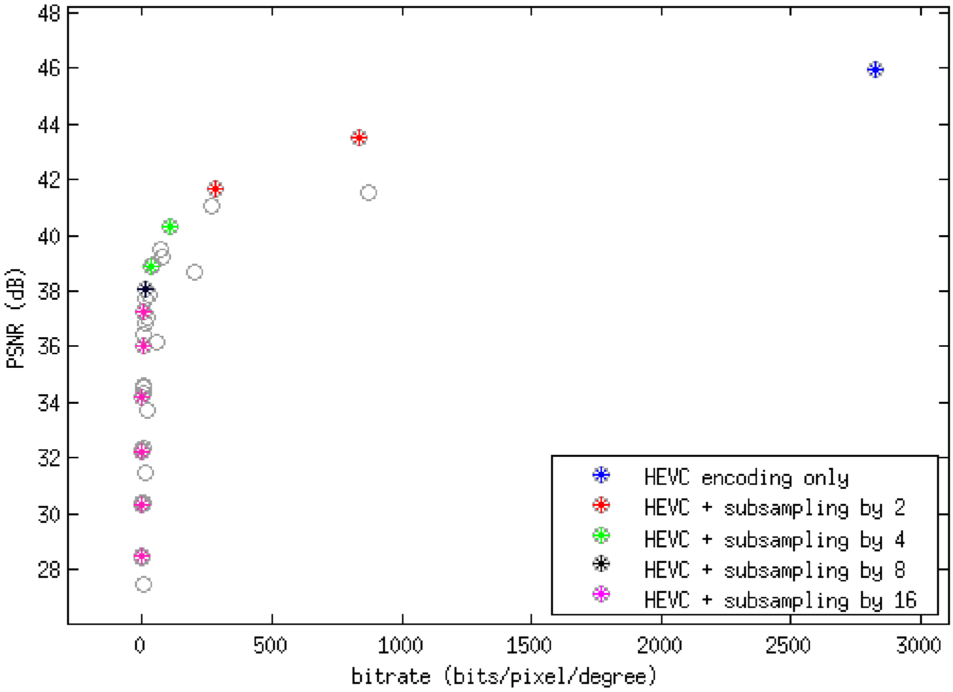


Figure 5.14: *Convex hull points for San Miguel - UFRJ rate-distorion results.*

With the help of the convex hull, it is possible to discover which coding solution is the best given a target bitrate or given a desired final quality.

For example, looking again at Figure 5.14, one can say that if the desired PSNR of a decoded sequence is $40dB$, the best coding solution is to subsample the original

San Miguel - UFRJ by four, encode it with HEVC and then interpolate by four the decoded sequence. The same rule can be applied for choosing a target bitrate. If the user wants to spend 500 *bits/pixel/degree* at most, he can use the HEVC with subsampling the input sequence by two. This solution will provide him the best rate-distortion compromise among the coding solutions.

On Table 5.14, are listed all quality and bitrate ranges where each solution is the best for each sequence. Using this table is also possible to find the best coding solution and the best parameters given a target bitrate by consulting the *Target Bitrate* column. Also, one can find the best parameters and coding solutions for a given the target quality by consulting the column *Target Quality*.

Table 5.14: *Quality range and bitrate range in which each coding solution performs is the best choice. Elephant sequence subsampled by eight does not present a range where it is the best solution.*

Multiview Sequence	**Subsampling Factor**	**GOV Structure**	**GOV Size**	**Target Bitrate (bits/pixel/degree)**	**Target Quality (dB)**
San Miguel - UFRJ	1	IBBBP	256	$R > 1828.2$	$Q > 44.7242$
	2	IBBBP	256	$194.9 < R \leq 1828.2$	$40.9850 < Q \leq 44.7242$
	4	IBBBP	128	$22.1 < R \leq 194.9$	$38.4800 < Q \leq 40.9850$
	8	IBBBP	64	$9.2 < R \leq 22.1$	$37.6700 < Q \leq 38.4800$
	16	IBBBP	32	$R \leq 9.2$	$Q \leq 37.6700$
Audi TT	1	IBBBP	256	$R > 103.8606$	$Q > 48.5886$
	2	IBBBP	256	$4.2723 < R \leq 103.8606$	$46.1200 < Q \leq 48.5886$
	4	IBBBP	128	$0.6098 < R \leq 4.2723$	$44.6150 < Q \leq 46.1200$
	8	IBBBP	64	$0.0266 < R \leq 0.6098$	$33.4950 < Q \leq 44.6150$
	16	IBBBP	16	$R \leq 0.0266$	$Q \leq 33.4950$
Champagne Tower	1	IBBBP	64	$R > 0.1697$	$Q > 23.3528$
	2	IBBBP	32	$R \leq 0.1697$	$Q \leq 23.3528$
Pantomime	1	IBBBP	32	$R > 0.4130$	$Q > 39.8052$
	2	IBBBP	16	$0.1812 < R \leq 0.4130$	$38.7250 < Q \leq 39.8052$
	4	IBBBP	8	$0.0771 < R \leq 0.1812$	$37.7950 < Q \leq 38.7250$
	8	IBBBP	4	$R \leq 0.0771$	$Q \leq 37.7950$
Dog	1	IBBBP	64	$R > 0.6523$	$Q > 36.2300$
	2	IBBBP	32	$0.1330 < R \leq 0.6523$	$33.3150 < Q \leq 36.2300$
	4	IBBBP	16	$0.0214 < R \leq 0.1330$	$27.7800 < Q \leq 33.3150$
	8	IBBBP	4	$R \leq 0.0214$	$Q \leq 27.78002$
Balloons	1	IBBBP	4	$R > 0.0221$	$Q > 38.9558$
	2	IBBBP	2	$R \leq 0.0221$	$Q \leq 38.9558$
Kendo	1	IBBBP	4	$R > 0.0254$	$Q > 42.7223$
	2	IBBBP	2	$R \leq 0.0254$	$Q \leq 42.72238$
Elephant	1	IBBBP	128	$R \leq 232.3479$	$Q \leq 43.9453$
	2	IBBBP	64	$42.8442 < R \leq 232.3479$	$40.9000 < Q \leq 43.9453$
	4	IBBBP	32	$9.3085 < R \leq 42.8442$	$38.7800 < Q \leq 40.9000$
	8	IBBBP	16	n.a.	n.a.
	16	IBBBP	8	$R \leq 9.3085$	$Q \leq 38.7800$
Train	1	IBBBP	64	$R > 73.3412$	$Q > 43.3533$
	2	IBBBP	64	$17.1876 < R \leq 73.3412$	$41.2400 < Q \leq 43.3533$
	4	IBBBP	32	$6.8977 < R \leq 17.1876$	$39.2950 < Q \leq 41.2400$
	8	IBBBP	8	$0.1674 < R \leq 6.8977$	$24.2750 < Q \leq 39.2950$
	16	IBBBP	4	$R \leq 0.1674$	$Q \leq 24.2750$

From Table 5.14, it is also possible to conclude some facts relating the angular density of the sequences and their performance when using subsampling at the encoder and interpolation at the decoder..

For example, the range of bitrates in which a coding solution is the best choice depends on the angular density of the original sequence. For example, it is less likely to choose a subsampling factor different from one (no subsampling factor) for low density sequences then for high density sequences. Figure 5.15 illustrates the difference in bitrate ranges between a low density and a high density mutiview sequence.

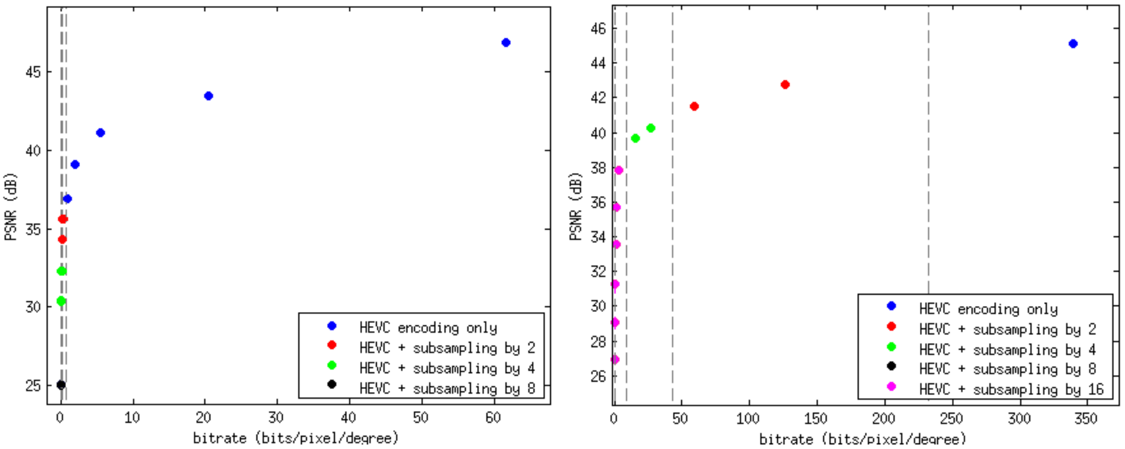


Figure 5.15: *Subsampling ranges on the convex hull for Dog (left) and Elephant (right) sequences.*

Figure 5.15 shows that the bitrate ranges where one can use the subsampling operation of any factor greater than one is very narrow for low density sequences. High density sequences, on the other hand, provides a wider range of bitrates where the subsampling operation appears as the best solution. In other words, for high angular density sequences, given the bitrate, the best coding solution (among the coding solutions covered in this work) will most probably involve the subsampling operation. For low angular density sequences though, results say that the best coding solution does not involve discarding views.

The same can be said about the quality values. It is possible to use highest subsampling factors, saving bitrate, with high angular density sequences as the best approach while for low angular density sequences, in most cases, it is better not to put views away.

As an example, it will be chosen the mean quality of $40dB$ for the decoded sequence. At this quality level, Table 5.14 indicates that the best coding choice for *San Miguel - UFRJ*, *Elephant* and *Train* is the subsampling factor of four, and for *Audi TT* the best solution is a subsampling factor of eight. On the other hand, for the other sequences, with the exception of *Kendo*, the better solution is to encode the full sequence, without the subsampling operation, in order to achieve such quality level.

This result points to the fact that the HEVC coding tools are not well suited for exploiting the high level of inter-view redundancy present in high density multiview sequences. In order to achieve good coding efficiency one has to combine HEVC with subsampling at the encoder and interpolation at the decoder. It is an indication that it is worthy to pursue alternative coding methods for high density multiview sequences, and, as a consequence, light fields.

Chapter 6

Conclusions

This work has investigated the properties of static linearly-arranged light fields when compressed by different coding solutions based on the HEVC standard, by taking advantage of the the natural redundancy between camera views. Chapter 2 described the light-field model and presented the multiview sequences. Chapter 3 described the proposed experiments for mutiview coding while Chapter 4 dealt with all tools and its relevant parameters in order to perform the experiments.

All multiview sequences were chosen to be different one from another for the assessment of the compression performances to be as comprehensive as possible.

Chapter 5 has presented tests with different inter-view prediction structures and and investigated their parameters, such as GOV size and GOV structure. It also has shown that if the sequence does not contain difficult content, one can encode just a fraction of the views, sparing the encoder of dealing with a large amount of data. Then, the original number of views can be restored by applying an interpolation at the decoder. This solution has led this work to the following conclusions:

- High-angular density sequences (with the angular density of at least 6 views per degree) when subsampled before the encoder can reduce about 85% more bits than when encoded by plain-HEVC alone without redundancy removal;
- As the angular density decreases, the subsampling operation can become more susceptible to the content of the portrayed scene. Depending on this content, the subsampling can improve the inter-view prediction or not, as it also depends on the capacity of the interpolator at the decoder to portray the content at the interpolated view;
- Liquids, transparent, translucent or very reflexive objects can deteriorate the interpolated view as reflection spots changes according to the position of the camera while the rest of the scene is static;

- Low angular density sequences when submitted to a subsampling operation present reduced performance, that is highly dependent on their content. One reason is the reduced inter-view redundancy caused by the reduced correlation between contents. This reduced correlation occurence is due to the fact that the distance between the cameras is larger. Then, when views are discarded, important inter-view data is lost, diminishing the prediction performance and also reducing the quality of the interpolated views;

- While low angular density sequences present a very narrow bitrate range where the best coding solution presents subsampling operations, the high density sequences present wide ranges, indicating that subsampling the sequence can be a good choice before the usual HEVC coding for high angular density sequences;

- In other to obtain a decoded sequence at a target quality, the best choice is to subsample the sequence before encoding it if the sequence presents a high density of views, reducing the total amount of data. On the other hand, not subsampling low density sequences is a better solution in a rate-distortion sense.

One can conclude from the presented experiments of that the use of plain-HEVC is usually a good solution for low angular density sequences, with less inter-view redundancy between views. However, for sequences with a higher density of views per degree, just encoding with HEVC is not enough to fully exploit the inter-view redundancy. This fact has been proven by the results that show an improvement in rate-distortion performance by discarding views prior to HEVC encoding, with posterior interpolation at the decoder. It means that the HEVC does not exploit very well the redundancy present in these sequences, pointing out that it is worthy to investigate new alternatives for inter-view redundancy reduction for multiview sequences. With good methods for reducing this inter-view redundancy, it will be possible to extend the research to more complex material, and explore new perceptions in more sophisticated approaches of the plenoptic function.

6.1 Future Work

There are several interesting subjects that could complement or extend this work. Some of these topics are:

- **View synthesis using depth maps:** improve the interpolation operation used in this work by testing specific sequences which contain depth maps, evaluating its overall coding performance;

- **Two-dimensional multiview sequences:** Extend the research to two-dimensional multiview sequences and study the two-dimensional inter-view prediction in order to discover the set of parameters and structures that provide efficiency in compression;

- **Adaptive GOV structure:** As a scene does not contain necessarily the same contents in all recorded views, an adaptive GOV structure could be used. It could also be possible to add an adaptive subsampler combined with a dedicated interpolator in order to keep just the essential information of the multiview sequence.

Appendix A

Multiview test sequences

The following pages present and gives a brief description of each one of the ten chosen multiview sets. Table A.1 gives a brief description of the main characteristics of these sequences.

Table A.1: *List of multiview sequences used in this work and their main characteristics.*

Acquisition	**Origin**	**Number of Views**	**View format**	**Sequence Name**	**Details**
Rendering	UFRJ	850	Static	San Miguel - UFRJ	Page 69
		640		Audi TT	Page 70
	UHasselt	200		San Miguel - UHasselt	Page 71
1-D Linear Camera Array	Nagoya University	79	Video Sequence	Champagne Tower	Page 72
				Pantomime	Page 73
				Dog	Page 74
		7		Balloons	Page 75
				Kendo	Page 76
	MERL	460	Static	Elephant	Page 77
		500		Train	Page 78

San Miguel - UFRJ

San Miguel - UFRJ is a realistic rendered scene based on a *hacienda* in San Miguel de Allende (see Figure A.1). Composed by plant pots, some trees, tables and chairs; there is also a fountain in the middle of the yard, pillars with arcs and balconies.

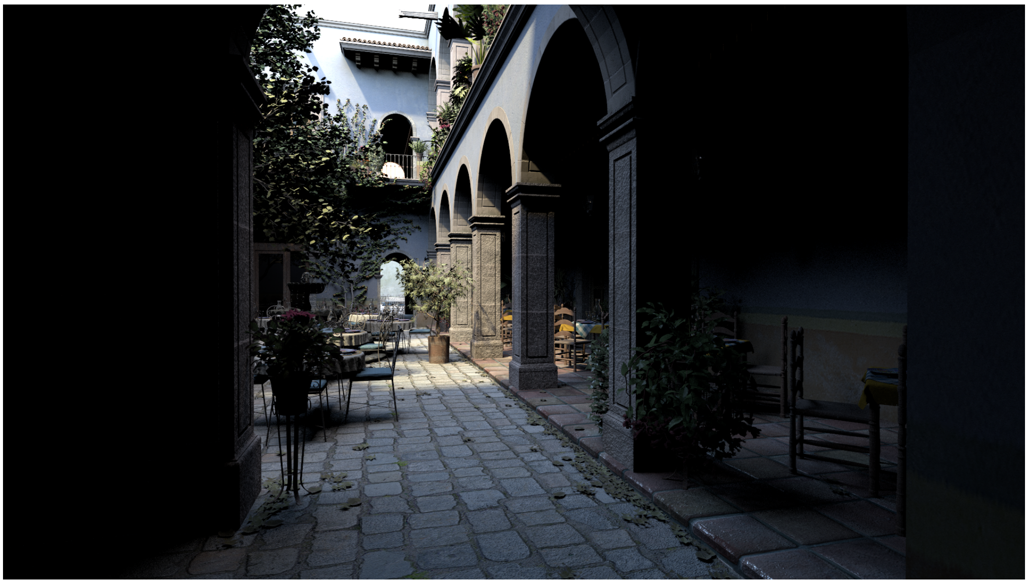


Figure A.1: *View number 425 from San Miguel - UFRJ sequence.*

Luminance resolution:	1920 × 1080;
Number of views:	850 horizontally aligned views;
Number of frames per view:	One frame;
Acquisition method:	Rendered by Physically Based Rendering Software (PBRT) [10];
Depth Maps:	No depth maps available;
Origin:	Signals, Multimedia and Telecommunications Laboratory (SMT) COPPE/UFRJ; Brazil [26]
Copyright:	Only available for academic usage.

Audi TT

Audi TT is a computationally created scene including an Audi TT car in a monochromatic background (see Figure A.2). The foreground of the scene presents a detailed texture and noticeable diffuse and specular lighting while the background only shows a smooth grey colour.

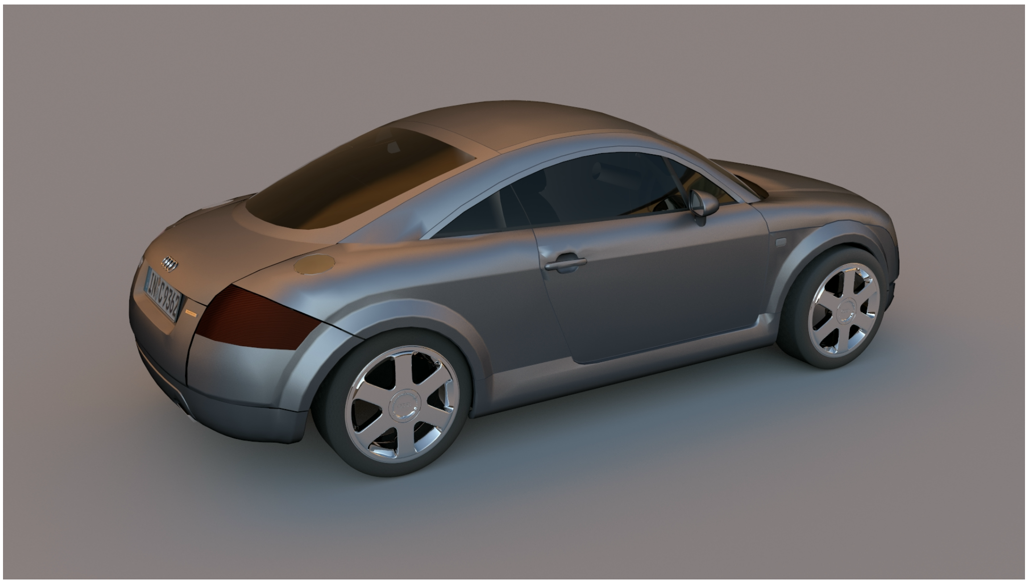


Figure A.2: *View number 320 from Audi TT sequence.*

Luminance resolution:	1920 × 1080;
Number of views:	640 horizontally aligned views;
Number of frames per view:	One frame;
Acquisition method:	Rendered using Physically Based Rendering Software (PBRT); [10]
Depth Maps:	No depth maps available;
Origin:	Signals, Multimedia and Telecommunications Laboratory (SMT) COPPE/UFRJ, Brazil; [26]
Copyright:	Only available for academic usage.

San Miguel - UHasselt

The San Miguel - UHasselt sequence is a rendered hacienda in San Miguel de Allende from a different point of view from the San Miguel - UFRJ sequence (see Figure A.3).

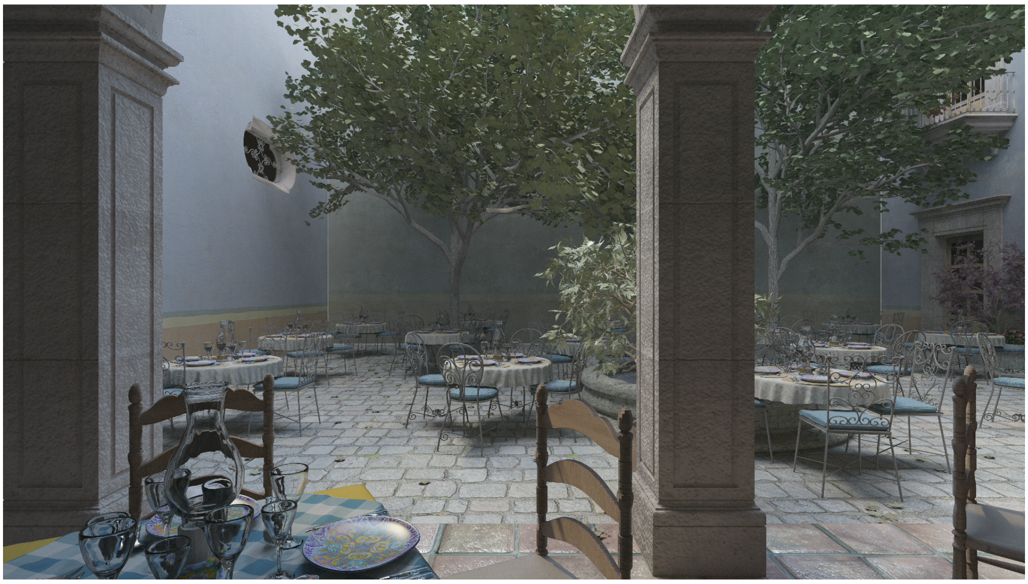


Figure A.3: *View number 200 from San Miguel - UHasselt sequence.*

Luminance resolution:	1920 × 1080;
Number of views:	200 horizontally aligned views;
Number of frames per view:	One frame;
Acquisition method:	Rendered using Physically Based Rendering Software (PBRT) [10]
Depth Maps:	No depth maps available;
Origin:	Universiteit Hasselt, Belgium;
Copyright:	Only available for academic usage production.

Champagne Tower

Champagne Tower shows a woman filling with champagne a six-leveled pyramidal tower of glasses over a table (see Figure A.4). The room also contains some dodecahedron loudspeakers and a black wall as background.

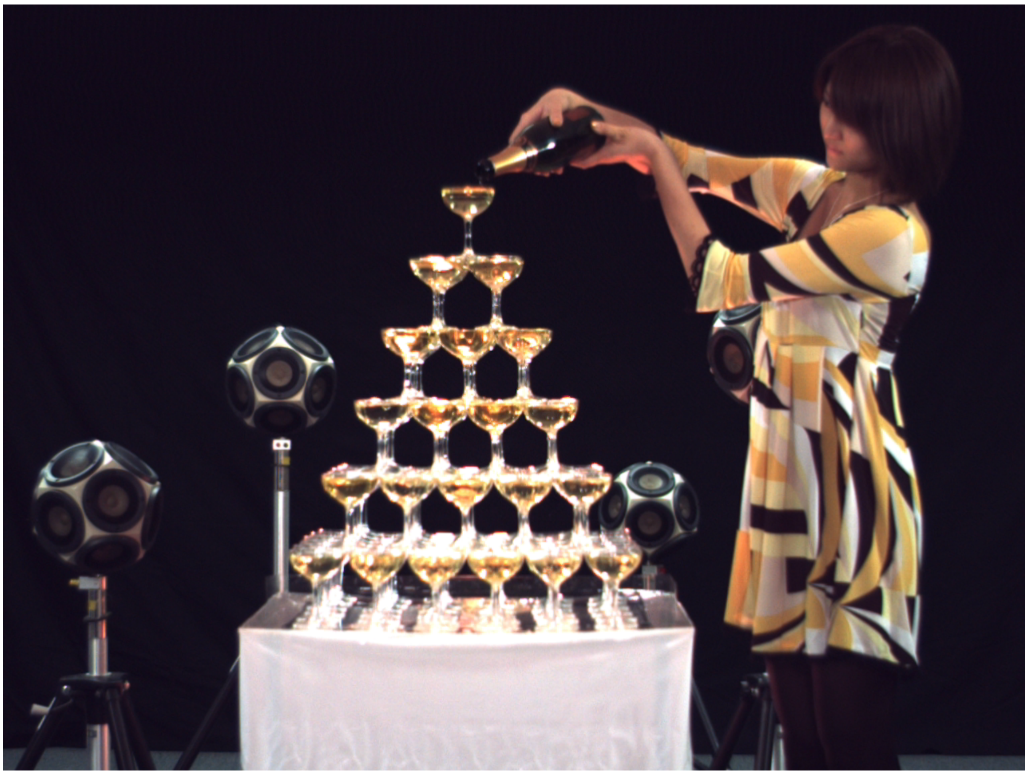


Figure A.4: *First frame from the view number 40 of Champagne Tower sequence.*

Luminance resolution:	1280 × 960;
Number of views:	79 horizontally aligned views;
Number of frames per view:	300 frames;
Acquisition method:	Horizontally aligned cameras with stereo distance (6.35 cm);
Depth Maps:	Three depth maps corresponding to the 37th, 39th and 41st view;
Origin:	Fujii Laboratory at Nagoya University;
Copyright:	Only available for academic usage.

Pantomime

Pantomime is a multiview set showing two clowns performing with a briefcase in front of a black wall (see Figure A.5). Their clothes are very detailed and colourful with many vertical stripes.

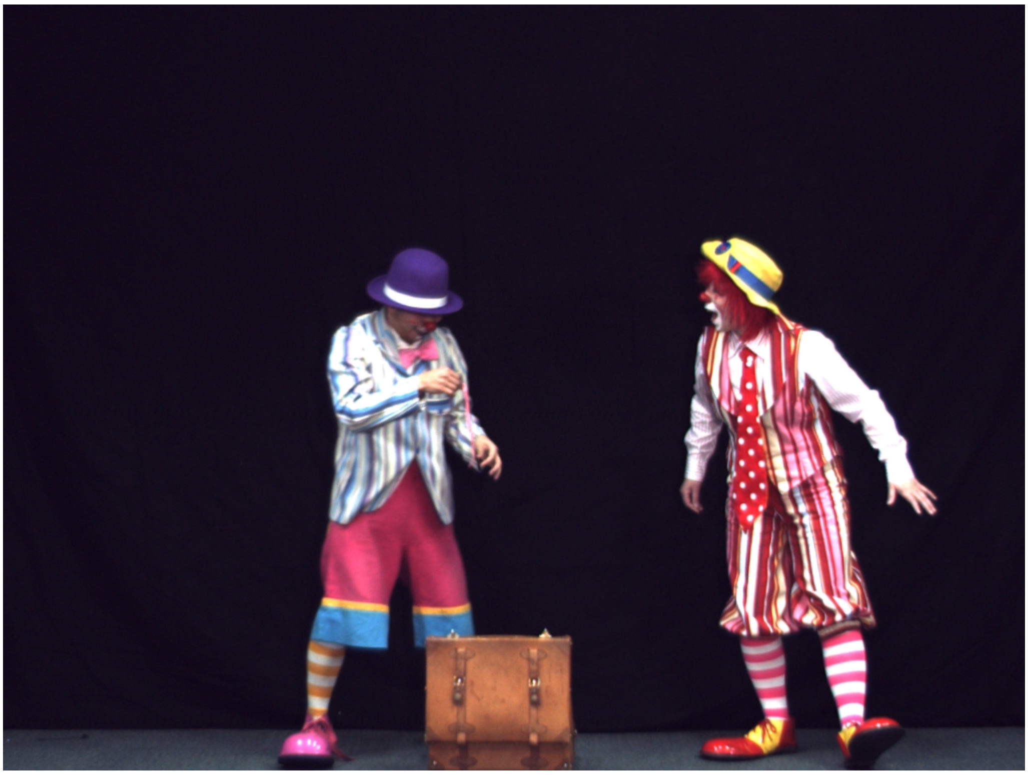


Figure A.5: *First frame from the view number 40 of Pantomime sequence.*

Luminance resolution:	1280 × 960;
Number of views:	79 horizontally aligned views;
Number of frames per view:	300 frames;
Acquisition method:	Horizontally aligned cameras with stereo distance (6.35 cm);
Depth Maps:	Three depth maps corresponding to the 37^{th}, 39^{th} and 41^{st} view;
Origin:	Fujii Laboratory at Nagoya University;
Copyright:	Only available for academic usage.

Dog

Dog is a scene composed by a person and a dog performing some actions in front of a wavy curtain (see Figure A.6). The curtain has many small colourful dots.

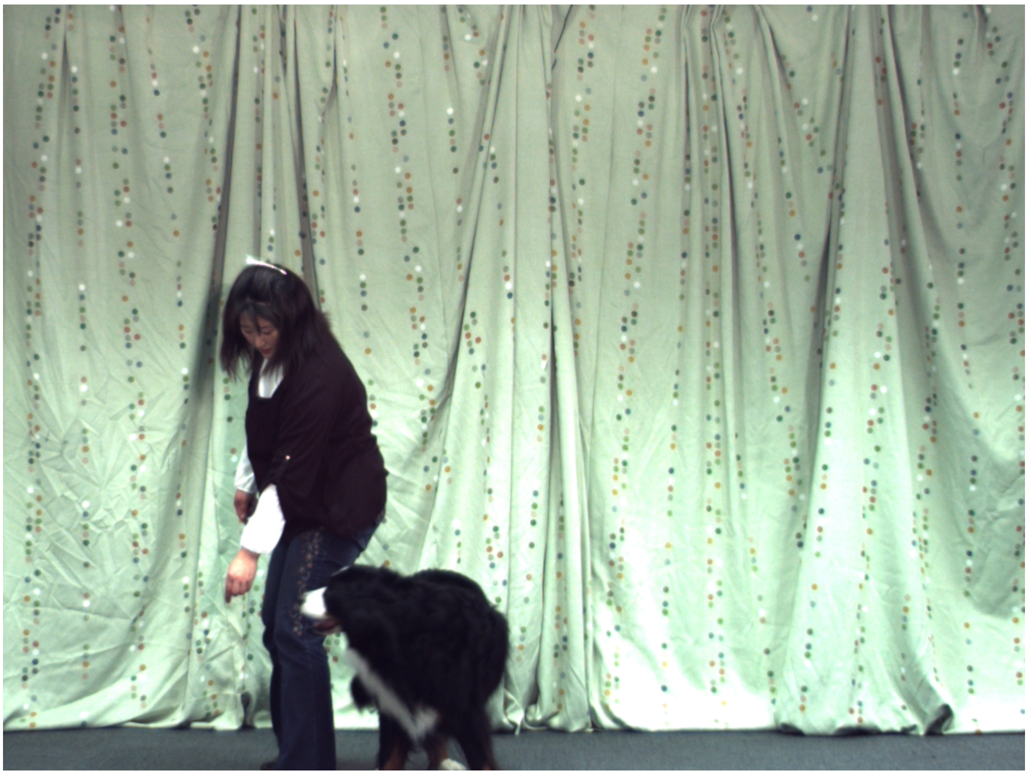


Figure A.6: *First frame from the view number 40 of Dog sequence.*

Luminance resolution:	1280 × 960;
Number of views:	79 horizontally aligned views;
Number of frames per view:	300 frames;
Acquisition method:	Horizontally aligned cameras with stereo distance (6.35 cm);
Depth Maps:	Not available;
Origin:	Fujii Laboratory at Nagoya University;
Copyright:	Only available for academic usage.

Balloons

Balloons is a multiview scene composed by balloons all over a room, a man holding a big ball and some plants (see Figure A.7). The background is composed by a wall lit by a white light drawing musical notes.

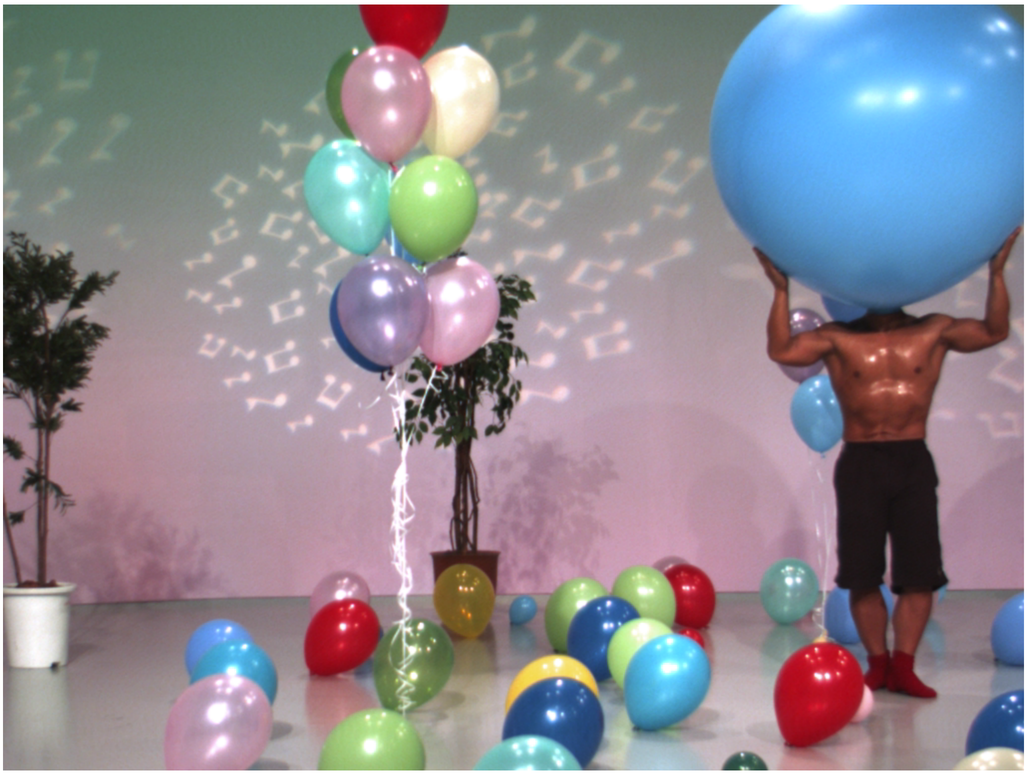


Figure A.7: *First frame from the view number 4 of Balloons sequence.*

Luminance resolution:	1024 × 768
Number of views:	7 horizontally aligned views;
Number of frames per view:	400 frames;
Acquisition method:	Horizontally aligned cameras with 5cm spacing;
Depth Maps:	Not available;
Origin:	Fujii Laboratory at Nagoya University;
Copyright:	Only available for academic usage.

Kendo

Kendo is a multiview scene showing two swordsmen practising kendo (see Figure A.8). They fight in front of an audience, sat on stairs. The scene has also white smoke on the floor and a lit red wall. A plant pot completes the scene.

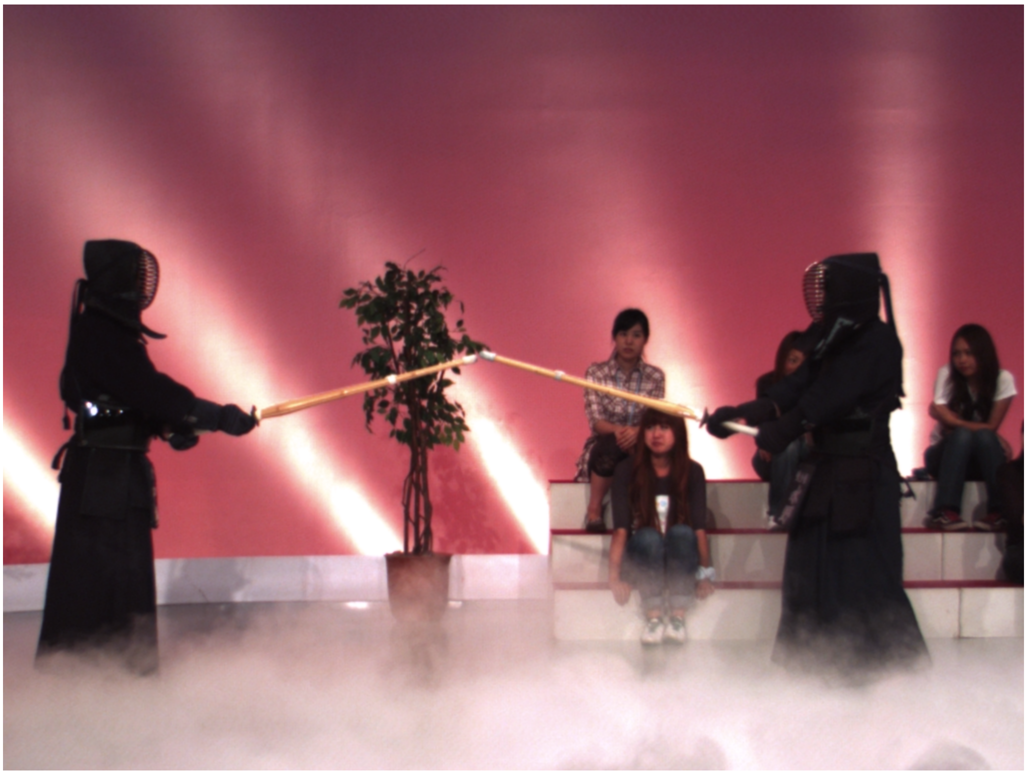


Figure A.8: *First frame from the view number 4 of Kendo sequence.*

Luminance resolution:	1024 × 768;
Number of views:	7 horizontally aligned views ;
Number of frames per view:	400 frames;
Acquisition method:	Horizontally aligned cameras with 5cm spacing;
Depth Maps:	Not available;
Origin:	Fujii Laboratory at Nagoya University;
Copyright:	Only available for academic usage.

Elephant

Elephant is a multiview scene that contains an elephant toy over a table with some plants in the background (see Figure A.9).

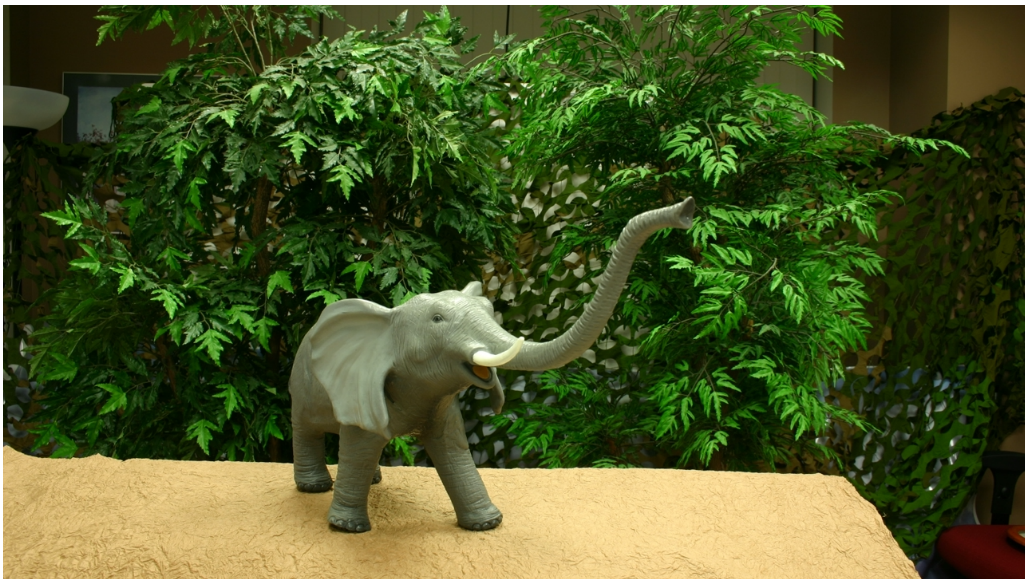


Figure A.9: *View number 230 from Elephant sequence.*

Luminance resolution:	1280 × 853;
Number of views:	460 horizontally aligned views;
Number of frames per view:	One frame;
Acquisition method:	Not available;
Depth Maps:	Not available;
Origin:	Mitsubishi Electric Research Laboratories [27];
Copyright:	Only for non-commercial purposes.

Train

Train is a 3D multiview set depicting a scale model with a train toy over a city model and as background a wallpaper of mountains (see Figure A.10). The model also contains buildings, bushes and poles.

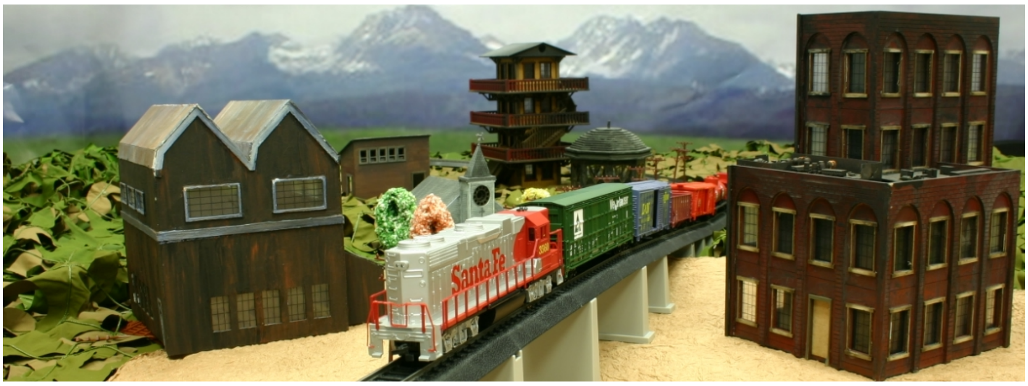


Figure A.10: *View number 250 from Train sequence.*

Luminance resolution:	1255 × 473;
Number of views:	500 horizontally aligned views;
Number of frames per view:	One frame;
Acquisition method:	Not available;
Depth Maps:	Not available;
Origin:	Mitsubishi Electric Research Laboratories [27];
Copyright:	Only for non-commercial purposes.

Bibliography

[1] NG, R., LEVOY, M., MATHIEU BRÉDIF, E. A. *Light Field Photography with a Hand-Held Plenoptic Camera.* Relatório técnico, Stanford University Computer Science Tech Report CSTR 2005-02, 2005.

[2] FUJII LABORATORY. "Nagoya University Multi-view Sequences Download List". Disponível em: <http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/>.

[3] KONRAD, J., HALLE, M. "3-D displays and signal processing", *IEEE Signal Processing Magazine*, v. 6, n. 24, pp. 97–111, 2007.

[4] TANIMOTO, M., RONFARD, R., TAUBIN, G. "Free-Viewpoint Television". In: *Image and Geometry Processing for 3-D Cinematography*, pp. 53–76, Berlin, Heidelberg, Springer Berlin Heidelberg, 2010. ISBN: 978-3-642-12392-4. doi: 10.1007/978-3-642-12392-4_3. Disponível em: <http://dx.doi.org/10.1007/978-3-642-12392-4_3>.

[5] ADELSON, E. H., BERGEN, J. R. "The plenoptic function and the elements of early vision". In: *Computational Models of Visual Processing*, pp. 3–20. MIT Press, 1991.

[6] MAXWELL, J. C. "On the Theory of Three Primary Colours", *W.D. Nevin(ed.), Sci. Papers 1, Cambridge Univ. Press, London*, 1861.

[7] LIOU, H.-L., BRENNAN, N. A. "Anatomically accurate, finite model eye for optical modeling", *JOSA A*, v. 14, n. 8, pp. 1684–1695, 1997.

[8] GERSHUN, A. "The light field", *Journal of Mathematics and Physics*, v. 18, n. 1, pp. 51–151, 1939.

[9] PHARR, M., HUMPHREYS, G. "Physically Based Rendering". www.pbrt.org, 2012.

[10] PHARR, M., HUMPHREYS, G. *Physically Based Rendering: From Theory To Implementation.* Second edition ed. Burlington, Massachusetts, Morgan Kaufmann/Elsevier, 2010.

[11] NG, R. *Digital Light Field Photography.* Tese de Doutorado, Stanford, CA, USA, 2006. AAI3219345.

[12] CONTI, C., LINO, J., NUNES, P., et al. "Spatial prediction based on self-similarity compensation for 3D holoscopic image and video coding". In: *2011 18th IEEE International Conference on Image Processing*, pp. 961–964, Sept 2011. doi: 10.1109/ICIP.2011.6116721.

[13] CONTI, C., NUNES, P., SOARES, L. D. "New HEVC prediction modes for 3D holoscopic video coding". In: *2012 19th IEEE International Conference on Image Processing*, pp. 1325–1328, Sept 2012. doi: 10.1109/ICIP.2012.6467112.

[14] ITU-T, ISO/IEC JTC 1/SC 29 (MPEG). *High efficiency video coding.* Recommendation ITU-T H.265 and ISO/IEC 23008-2, 2013.

[15] MONTEIRO, R., LUCAS, L., CONTI, C., et al. "Light field HEVC-based image coding using locally linear embedding and self-similarity compensated prediction". In: *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 1–4, July 2016. doi: 10.1109/ICMEW.2016.7574670.

[16] FEHN, C. "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV". In: *Electronic Imaging 2004*, pp. 93–104. International Society for Optics and Photonics, 2004.

[17] ASCENSO, J., BRITES, C., PEREIRA, F. "Content Adaptive Wyner-ZIV Video Coding Driven by Motion Activity". In: *2006 International Conference on Image Processing*, pp. 605–608, Oct 2006. doi: 10.1109/ICIP.2006.312408.

[18] MARTINIAN, E., BEHRENS, A., XIN, J., et al. "View synthesis for multiview video compression". In: *Picture Coding Symposium*, v. 37, pp. 38–39, 2006.

[19] TAMBOLI, R. R., APPINA, B., CHANNAPPAYYA, S., et al. "Super-multiview Content with High Angular Resolution", *Image Commun.*, v. 47, n. C, pp. 42–55, set. 2016. ISSN: 0923-5965. doi: 10.1016/j.image.2016.05.010. Disponível em: <`https://doi.org/10.1016/j.image.2016.05.010`>.

[20] BLACK, M. J., ANANDAN, P. "A framework for the robust estimation of optical flow". In: *1993 (4th) International Conference on Computer Vision*, pp. 231–236, May 1993. doi: 10.1109/ICCV.1993.378214.

[21] SHARABAYKO, M. P., PONOMAREV, O. G., CHERNYAK, R. I. "Intra compression efficiency in VP9 and HEVC", *Applied Mathematical Sciences*, v. 7, n. 137, pp. 6803–6824, 2013.

[22] "ITU-T Recommendation H.265: High efficiency video coding". abr. 2015. Disponível em: <http://www.itu.int/rec/T-REC-H.265-201504-I/en>.

[23] FRAUNHOFER HHI. "SVN repository for HM-16.0 software". 2014. Disponível em: <http://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.0>.

[24] BJØNTEGAARD, G. "Calculation of average PSNR differences between RD-curves", *Proceedings of the ITU-T Video Coding Experts Group (VCEG) Thirteenth Meeting*, 2001.

[25] ORTEGA, A., RAMCHANDRAN, K. "Rate-distortion methods for image and video compression", *IEEE Signal processing magazine*, v. 15, n. 6, pp. 23–50, 1998.

[26] SMT/COPPE/UFRJ. "UFRJ Light Field Repository". 2015. Disponível em: <http://www02.smt.ufrj.br/~luiz.tavares/lightfield/>.

[27] UCSD/MERL. "UCSD/MERL Light Field Repository". Disponível em: <vision.ucsd.edu/datasets/lfarchive/>.

[28] FRAUNHOFER HHI. "SVN repository for HTM-12.0 software". 2014. Disponível em: <https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-12.0>.

[29] KUHFUSS, MARCEL. "Doença Forever: An introduction". 2015.

[30] SULLIVAN, G. J., OHM, J.-R., HAN, W., et al. "Overview of the High Efficiency Video Coding (HEVC) Standard." *IEEE Trans. Circuits Syst. Video Techn.*, v. 22, n. 12, pp. 1649–1668, 2012. Disponível em: <http://dblp.uni-trier.de/db/journals/tcsv/tcsv22.html#SullivanOHW12>.

[31] "ITU-T Recommendation H.264 : Advanced video coding for generic audiovisual services". nov. 2007. Disponível em: <http://www.itu.int/rec/T-REC-H.264-200711-I/en>.

[32] RIBEIRO, F. M. L., OLIVEIRA, J. F. L., CIANCIO, A. G., et al. "Quality of Experience in a Multi-view Interactive 3D Environment". 7 2016.

[33] ARMINGTON, J. C., BIERSDORF, W. R. “Flicker and Color Adaptation in the Human Electroretinogram”, *J. Opt. Soc. Am.*, v. 46, n. 6, pp. 393–400, Jun 1956.

[34] LUKACS, M. “Predictive coding of multi-viewpoint image sets”. In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP ’86.*, v. 11, pp. 521–524, Apr 1986. doi: 10.1109/ICASSP.1986.1169032.

[35] LI, Y., SJÖSTRÖM, M., OLSSON, R., et al. “Efficient intra prediction scheme for light field image compression”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 539–543, May 2014. doi: 10.1109/ICASSP.2014.6853654.