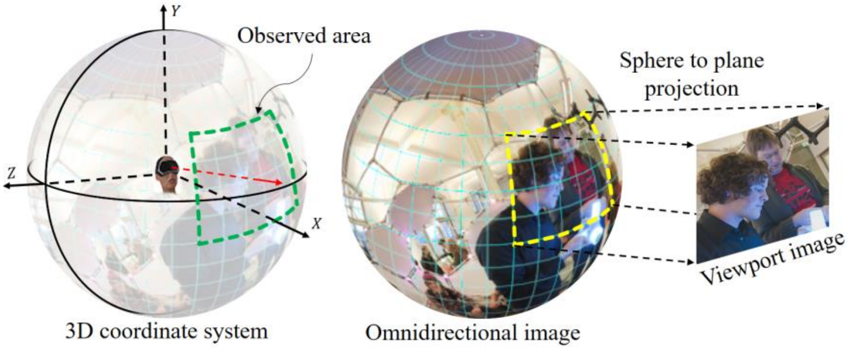
UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO



Assessment and Optimization of Omnidirectional Images Viewport Rendering

Falah Jabar Rahim

Supervisor: Doctor Maria Paula dos Santos Queluz Rodrigues

Co-supervisor: Doctor João Miguel Duarte Ascenso

Thesis approved in public session to obtain the PhD Degree in Electrical and Computer Engineering

Jury final classification: Pass with Distinction

2022

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Assessment and Optimization of Omnidirectional Images Viewport Rendering

Falah Jabar Rahim

Supervisor: Doctor Maria Paula dos Santos Queluz Rodrigues
Co-supervisor: Doctor João Miguel Duarte Ascenso

Thesis approved in public session to obtain the PhD Degree in
Electrical and Computer Engineering

Jury final classification: Pass with Distinction

Jury

Chairperson: Doctor Mário Alexandre Teles de Figueiredo, Instituto Superior Técnico, Universidade de Lisboa

Members of the Committee:

Doctor Luís Eduardo de Pinho Ducla Soares, Escola de Tecnologias e Arquitetura, ISCTE-Instituto Universitário de Lisboa

Doctor Alexandre José Malheiro Bernardino, Instituto Superior Técnico, Universidade de Lisboa

Doctor Pedro António Amado Assunção, Escola Superior de Tecnologia e Gestão, Politécnico de Leiria

Doctor Maria Paula dos Santos Queluz Rodrigues, Instituto Superior Técnico, Universidade de Lisboa

Funding Institutions

This research work was funded by Fundação para a Ciência e Tecnologia (FCT), Portugal, and by Agência Nacional de Inovação, Portugal.

2022

To the memory of my beloved little brother,

Ibrahim (1992-2009),

my father,

Jabar (1942-2020),

and my uncle,

Hasan (1947-2021).

Abstract

Nowadays, omnidirectional (or 360°) visual content is driving the creation of new and immersive services and virtual reality (VR) applications in the fields of medicine, architecture, arts, entertainment, education, sports, and tourism, among others. Omnidirectional visual content is typically captured with a circular array of cameras and represents the whole visual field surrounding the capture point, which allows to provide an immersive experience to the users. To visualize omnidirectional visual content on planar displays, a fraction of the omnidirectional image is projected on a plane, resulting in a 2D image known as *viewport*; this process is usually called viewport rendering. However, since a sphere is not a developable surface, any sphere to plane projection introduces geometrical distortions, such as the stretching of objects and/or the bending of straight lines, which may compromise, in a significant way, the user's quality of experience (QoE). In this context, quality assessment of the viewport images that are produced by the rendering of omnidirectional content is much needed.

The main objective of this Thesis is to subjectively and objectively assess the perceptual impact of the geometric distortions introduced in viewport rendering, mostly due to the sphere to plane projection. Furthermore, it is intended to optimize the sphere to plane projection in a perceptual way, resulting in a perceptually pleasing viewport image after rendering.

Several subjective assessment experiments were conducted with different sphere to plane projections, notably the general perspective projection (GPP) and the Pannini projection (PP). These experiments allowed to evaluate the geometric distortions impact and were followed by the design of new content-aware objective quality metrics, able to assess the perceived geometric distortions in a reliable way. The experimental results show that the proposed metrics are able to assess and predict the viewport quality with a high correlation with the subjective quality scores, i.e., close to human perception. Additionally, the proposed objective metrics were used to optimize the GPP and PP, resulting in content-aware GPP and content-aware PP projections. This procedure allows to minimize the geometric distortions, by globally adapting the projection parameters to the image content, resulting in viewport images with enhanced perceived quality. The content-aware Pannini projection was further optimized by also applying a local adaptation to the content, besides the global one. This allows an extra reduction of the geometric distortions, especially on regions where the human perception is more sensitive, such as objects, resulting in viewports with significant better visual quality than the benchmark, and state-of-the-art, projections, particularly when high field-of-views (~150°) are used.

Keywords: *Omnidirectional Images, Virtual Reality, Sphere to Plane Projection, Viewport Rendering, Geometric Distortions, Subjective Quality Assessment, Objective Quality Assessment, Content-Aware Projection.*

Resumo

Recentemente, o vídeo omnidirecional (ou 360◦) tem conduzido à criação de novos serviços e aplicações de realidade virtual (RV), nas áreas de medicina, arquitetura, artes, entretenimento, educação, desporto e turismo, entre outras. O conteúdo visual omnidirecional é tipicamente capturado com uma matriz de câmaras e representa, numa imagem esférica, todo o campo visual que circunda a zona de captura, permitindo oferecer uma experiência imersiva ao utilizador. Para visualizar este tipo de conteúdos em dispositivos planos, projecta-se uma fracção da imagem esférica num plano, do que resulta uma imagem 2D designada por *viewport*; este processo é genericamente designado por *viewport rendering*. No entanto, a projecção de uma superfície esférica num plano introduz sempre distorções geométricas, como o alongamento de objetos e/ou a flexão de linhas retas, que podem comprometer, de forma significativa, a qualidade da experiência (QoE) do utilizador. Neste contexto, é essencial avaliar a qualidade do *viewport* produzido pelo *rendering* do conteúdo omnidirecional.

O objetivo principal desta dissertação é avaliar, de forma subjetiva e objectiva, o impacto perceptual das distorções geométricas introduzidas no *viewport rendering*, e resultantes da projeção da imagem esférica num plano. Para além disso, pretende-se otimizar esta projecção de forma perceptual, de forma a produzir *viewports* com boa qualidade visual, minimizando o impacto negativo das distorções geométricas.

De forma a atingir os objectivos acima delineados, realizaram-se várias campanhas de avaliação subjectiva, utilizando *viewports* obtidos com diferentes projeções esfera-plano, nomeadamente a projeção perspectiva geral (GPP) e a projeção Pannini (PP). Estas campanhas permitiram avaliar o impacto das distorções geométricas, tendo sido seguidas pela proposta e desenvolvimento de novas métricas objectivas de avaliação de qualidade, capazes de avaliar as distorções geométricas introduzidas. Os resultados experimentais confirmaram que as métricas propostas são capazes de avaliar e prever a qualidade do *viewport* de forma bem correlacionada com os resultados da avaliação subjetiva, ou seja, próximas da percepção humana. Adicionalmente, as métricas objetivas propostas foram utilizadas para otimizar as projecções GPP e PP, de forma adaptada ao contéudo do *viewpor*t. Este procedimento permitiu minimizar as distorções geométricas, adaptando globalmente os parâmetros da projeção ao conteúdo da imagem, resultando em *viewports* com qualidade visual melhorada. A projeção Pannini foi ainda objecto de um processo de optimização adicional, de forma a ter também adaptação local, para além da global. A projecção resultante permitiu uma redução extra das distorções geométricas, especialmente em regiões onde a percepção humana é mais sensível, como sobre objetos, conduzindo a *viewports* com uma qualidade visual significativamente melhor do que a resultante de projecções consideradas estado-da-arte, e em particular quando são considerados campos de visão largos (~150º).

Palavras-chave: *Imagens Omnidirecionais, Realidade Virtual, Projeção Esfera-Plano, Viewport Rendering, Distorções Geométricas, Avaliação de Subjectiva de Qualidade, Avaliação Objectiva de Qualidade, Projeção Adaptada ao Conteúdo.*

Acknowledgments

I want to express my sincere gratitude to my supervisors, Prof. Maria Paula Queluz and Prof. João Ascenso, for their guidance and support, and without whom this work would not be possible. Thank you for your patience and for pushing me to be better.

To my friends and colleagues at Instituto de Telecomunicações, especially Ivo Sousa, Francisco Andrade, André Guarda, Alireza Javaheri, Milad Niknejad, Alireza Sepas-Moghadam, Tanmay Verlekar, Marco Pezzutto, Miskeen Khan, and Miguel Ferreira, for all the unforgettable memories we created together during these years, fruitful discussions, advice and support.

To my friends outside Instituto de Telecomunicações, especially Karwan Kurdi, Rawaz Kurda, Tiago Castelo, Miguel Coxo, Stephen Murphy, Elnaz Shadras, Ewa Szeliga, Lisa Kalthoff, Elliot Butler, Kaja Schmid, Seren Ustundag.

To my family, for their unconditional support.

This work was funded by FCT through national funds, and when applicable, co-funded by EU funds under the projects ENVISION/UID/EEA/50008/2013, PROVEN UIDB/50008/2020, and by ANI project (P2020) "MESMOQoE". I would also like to thank Instituto de Telecomunicações and Instituto Superior Técnico for providing me great working conditions and resources.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AR	Augmented Reality
AVC	Advanced Video Coding
ANOVA	Analysis of Variance
Acc	Accuracy
BT	Bradley Terry
CMP	Cube Map Projection
CPS	Classification by Projection Surface
CPC	Classification by Preserved Characteristic
CMOS	Comparative Mean Opinion Score
$\mathrm{CMOS}_{\mathrm{Pre}}$	Predicted Comparative Mean Opinion Score
CI	Confidence Interval
CV	Cross Validation
CA-GPP	Content-aware General Perspective Projection
CCA	Connected Component Analysis
CE	Classification Error
CD	Correct Decision
CDF	Cumulative Distribution Function
DoF	Degree-of-Freedom
DMOS	Differential Mean Opinion Score
ERP	Equirectangular Projection
EU	European Union
ERI	Equirectangular Image
FCT	Fundação para a Ciência e a Tecnologia
FoV	Field of View
F1	F1 Score
FT	False Tie
FD	False Differentiation
FR	False Ranking
GPP	General Perspective Projection
GIMP	GNU Image Manipulation Program
GUI	Graphical User Interface
GA-PP	Globally Adapted Pannini Projection
GT	Ground Truth
GLA-PP	Globally and Locally Adapted Pannini Projection
HMD	Head Mounted Display
HFoV	Horizontal Field of View
HEC	Hybrid Equi-angular Cube Map Projection
HEVC	High Efficiency Video Coding
ISO	International Organization for Standardization

IEC	International Electrotechnical Commission
ITU	International Telecommunication Union
LC	Line Curvature
LI	Line Inclination
MIPS	Most Isometric ParametrizationS
MPEG	Moving Picture Experts Group
MOP	Multiple Optimized Pannini
NLC	Normalized Line Curvature
NLI	Normalized Line Inclination
OP	Optimized Pannini
OMAF	Omnidirectional MediA Format
PP	Pannini Projection
PC	Pairwise Comparison
Prec	Precision
PLCC	Pearson Linear Correlation Coefficient
QoE	Quality of Experience
Rec	Recall
RMSE	Root Mean Square Error
SCACJ	Stimulus Comparison Adjectival Categorical Judgment
SVM	Support Vector Machine
SVR	Support Vector Regression
SROCC	Spearman Rank-order Correlation Coefficient
TPR	True Positive Rate
VR	Virtual Reality
VFoV	Vertical Field of View
VD	Viewing Direction
VVC	Versatile Video Coding
VS	Viewing Sphere
VC	Vertical Compression
2D	Two Dimensional
3D	Three Dimensional

Chapter 1

Introduction

1.1 Context and Motivation

In recent years, the popularity of omnidirectional visual content and applications is increasing rapidly, notably in virtual reality (VR) and augmented reality (AR). This has been driven by several technological advances, such as affordable 360° cameras, broadband connections, and head-mounted displays (HMD). Omnidirectional visual content can already be found in a large set of applications that users can enjoy, including immersive gaming, remote education, virtual shopping, virtual sports, virtual tours, and even broadcasting of live content. The user can now virtually attend live sports and concerts or watch specially designed movies and documentaries (cinematographic VR). Several VR applications have also been developed for training and education purposes in different scenarios, e.g., military actions, mechanical repairs and construction, surgery and medical care, art and architectural design. Even large technological industries, such as Google and Facebook, provide several applications and services, notably Google Street view, YouTube VR and Facebook 360° photos. Figure 1.1 shows some examples of applications which make use of omnidirectional images or videos. Moreover, the possibilities for future immersive applications and services are rather endless and is expected that in the next few years, more and more applications will emerge. The path towards immersive experiences and applications is also being supported by several standardization activities, e.g. the ISO/IEC MPEG-I, a family of standards for coded representation of immersive media [1]. However, to have a successful application, the user's quality of experience (QoE) - the degree of delight or annoyance of the user of an application or service [2] - should be high; accordingly, techniques to assess and improve the user's QoE have been an important research topic [3]–[5].

VR gaming

Remote education

Virtual shopping

VR virtual tours

Google 360° street view

Facebook 360° photos

360° web virtual tours

YouTube 360° videos

Figure 1.1. Some examples of omnidirectional visual content-based applications.

Typically, omnidirectional visual content contains the information of the scene around the camera, covering the whole 360° (horizontal) × 180° (vertical) viewing range. When this type of content is played, the user can observe any parts of the visual scene by changing the viewing direction ("look around"), which creates the feeling of being physically present. The user can explore the content according to some Degree-of-Freedom (DoF), i.e., the freedom of movement in the three-dimensional (3D) space [6]. There are six DoF, corresponding to three translations - forward/backward (or surge), up/down (or heave), left/right (or sway) - each along a coordinate axis, and three rotations - yaw, pitch, and roll - each around a coordinate axis, as depicted in Figure 1.2. Nowadays, only the three rotational DoFs are supported in most visualization systems of omnidirectional content, which already provides a visual experience more immersive than what is offered by traditional 2D visual content.

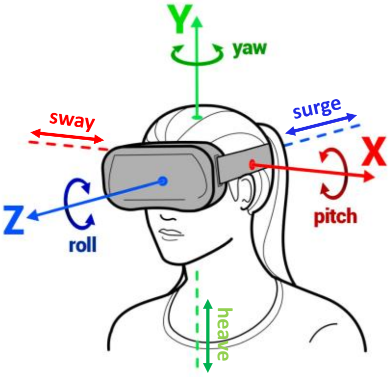


Figure 1.2. The six degrees of freedom (DoF) in the 3D space.

There are several ways to display omnidirectional visual content, that includes head-mounted displays, HMD (e.g., HTC Vive, Sony PlayStation VR, and Oculus Rift), smartphones and tablets, or standard computer monitors. The users may navigate on the visual content by moving his head and/or body on HMDs, with a mouse or keyboard on a computer monitor, or by moving the position of the smartphone or tablet in the physical space. Typically, the HMD provides a better immersive experience, although it is somewhat uncomfortable, expensive, and not accessible to all users. Therefore, watching omnidirectional visual content on smartphones or computer monitors is rather common. Several applications and services provided by Google and Facebook (e.g., the already mentioned Google Street view, Facebook 360 photos, and YouTube VR) aim the use of smartphones and personal computers. Thus, to evaluate and optimize the rendering of omnidirectional images and video on conventional 2D displays is nowadays rather important and is the target application scenario in this Thesis.

Omnidirectional visual content distribution pipeline involves several processing steps that include acquisition, stitching, mapping/inverse mapping, encoding/decoding, rendering, and visualization. Each one of these steps may introduce some visual distortions, such as noise, blurring, visible seams and, most importantly, geometrical distortions. This Thesis focus on the rendering step, which is responsible for the introduction of geometric distortions which may significantly impair the quality of experience offered to the users. Rendering must be always performed since the full omnidirectional image is not directly shown to the users. This process consists in projecting a fraction of the omnidirectional image (or spherical image) on a plane, as illustrated in Figure 1.3b); this generates a 2D image known as the viewport, which is viewed by the users.

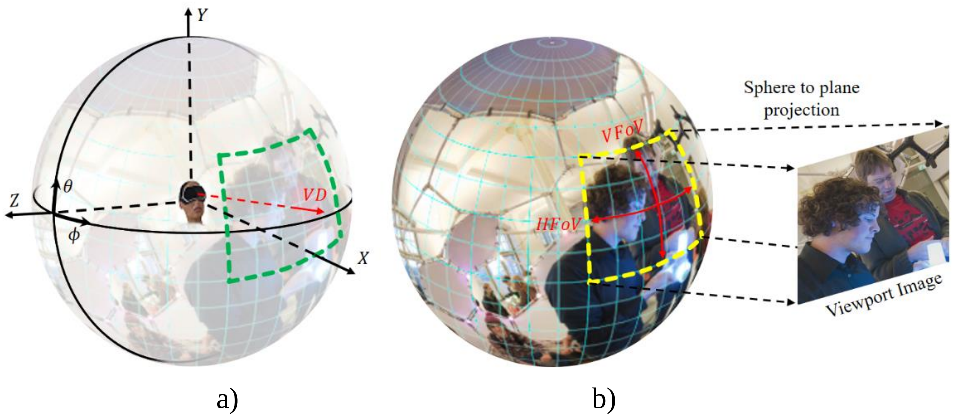


Figure 1.3. Viewport rendering: a) Spherical image and its coordinate system; b) Spherical image and rendered viewport, obtained by projecting a portion of the spherical image onto a plane.

The viewport content is defined by the viewing direction (VD) and by the horizontal and vertical fields of view (HFoV, VFoV), *cf*. Figure 1.3; when the user changes the viewing direction, the corresponding part of the omnidirectional visual content that is shown to the user changes accordingly. The rectilinear and stereographic projections - which are particular cases of the general perspective projection [7] - are typically used in the viewport rendering process [8][9].

Since a sphere is not a developable surface (i.e., a surface that can be flattened onto a plane without distortions), any mapping from a sphere to a plane introduces geometrical distortions on the resulting 2D image, e.g., objects are stretched (or sheared) and/or straight lines are bent. These distortions become more perceptually annoying as the field of view (FoV) increases, or when objects are close to the camera. Figure 1.4 shows an example of viewport images rendered from two different omnidirectional images, using the rectilinear and the stereographic projections, and a squared FoV of 110° (a FoV value often used in VR applications).

For the rectilinear projection, all straight lines in the visual scene remain straight in the viewport, thus without any bending; however, the objects close to the viewport borders are very stretched. As an example, in the *People* viewport the arm and shoulder of the boy holding a white light are geometrically distorted, being stretched or sheared, although the center of the image is acceptable. Interestingly, this stretching distortion is less visible for the *Buildings* viewport which is visually pleasing even at the viewport borders. On the other hand, the stereographic projection preserves, locally, the object shapes, but it severely bends the straight lines, a distortion also known as fisheye effect. This effect is illustrated in the stereographic viewports of Figure 1.4 where, for the *People* viewport, the boy holding a white light appears in the correct shape but the viewport is deformed globally due to the fisheye effect; also for the *Buildings* viewport, the horizontal and vertical lines are bent.

As shown by Figure 1.4, the visibility of geometric distortions depends on the image content. However, the perceptual impact of the geometric distortions has not been much studied in the literature. Thus, to study and assess these distortions is rather essential to characterize the viewport quality that can be offered by any omnidirectional image based application. This can be performed with subjective assessment tests, that are very time consuming and cannot be performed in real time, or with objective metrics, that automatically measure the viewport distortions, allowing to predict the viewport quality.

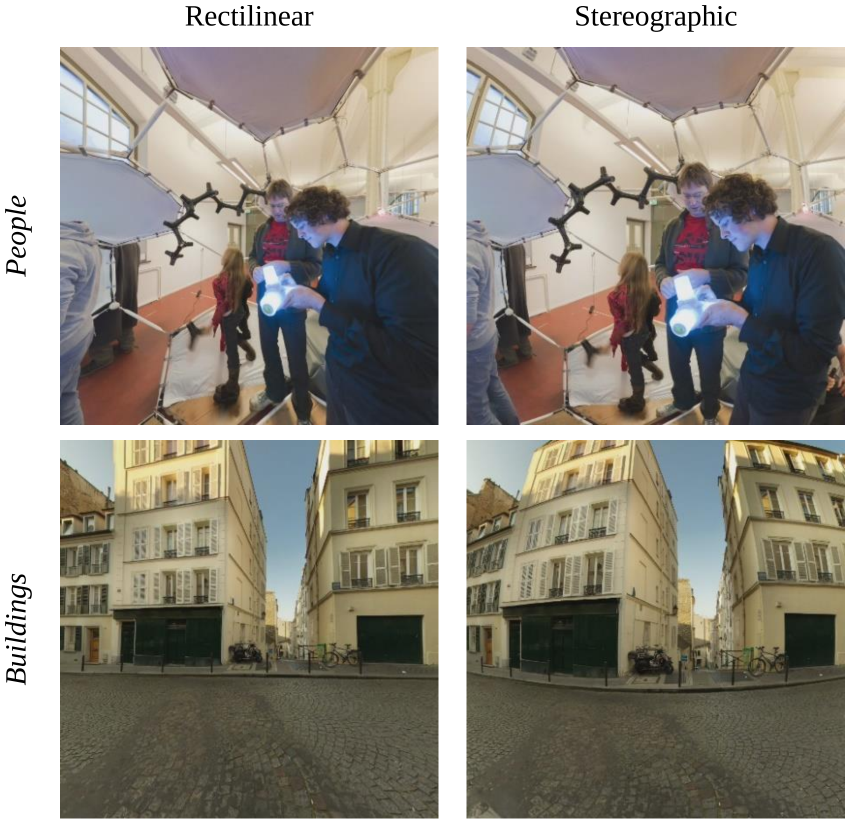


Figure 1.4. Viewport examples from two omnidirectional images, rendered with the rectilinear and stereographic projections using a square FoV of 110°.

Rectilinear and stereographic projections are content-unaware projections, meaning that the projection is always applied in the same way, regardless of the content to be projected. However, the viewport examples presented in Figure 1.4 show that the image content plays an important role on the perceived viewport quality. Accordingly, the projection should take the image content into account - i.e., being content-aware - to obtain a visually pleasant viewport image. The amount and type of geometric distortions can be controlled in most sphere to plane projections, including the general perspective projection (GPP) and the Pannini projection (PP), by the appropriate setting of the projection parameters. The GPP can be controlled by one parameter - the projection center - which can be used to obtain the rectilinear and stereographic projections, and also other intermediate projections (e.g., the Clarke, James, and orthographic projections). In the PP, and besides the projection center, an additional parameter - the vertical compression factor - can be adjusted, to decrease the bending of horizontal lines. Recently, the PP has been also used for viewport rendering of omnidirectional visual content [10]. Actually, this projection is more suitable for viewport rendering with a large FoV since it can generate viewports with less geometric distortions compared to the viewports obtained with rectilinear and stereographic [10].

To enhance the user's sense of immersion and engagement when exploring the omnidirectional visual content, the viewport FoV should be large. This is supported by several studies (e.g., [11][12]), showing that using a large field of view provides a more immersive and pleasant visual experience and maximizes the user's sense of presence, mainly because more visual information is available to cover the peripheral vision (i.e., closer to human FoV). However, the geometrical distortions introduced by the sphere to plane projection increase, and have more

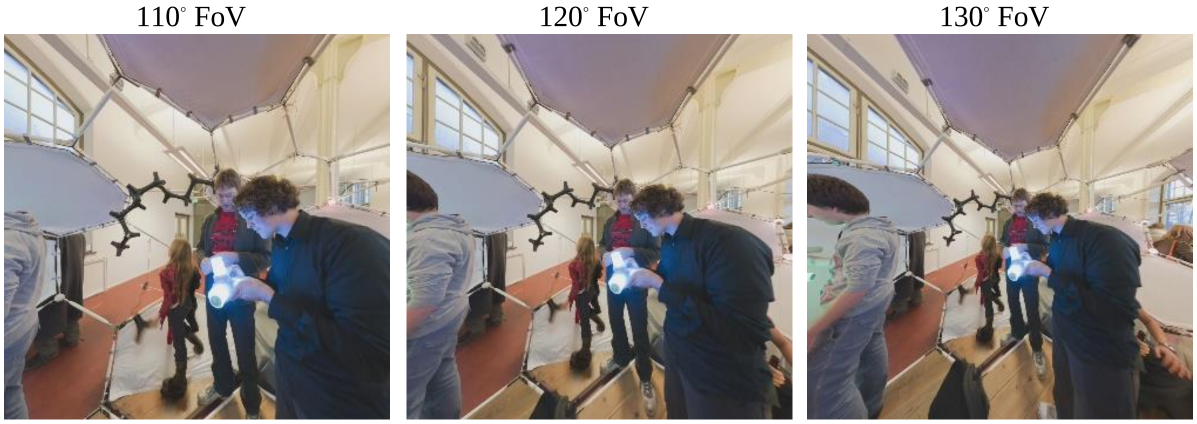


Figure 1.5. Example of viewports obtained with the rectilinear projection and varying FoV.

perceptual impact, as the viewport FoV increases. Figure 1.5 shows examples of rectilinear viewports rendered from the same omnidirectional image and viewing direction, with increasing FoV; the rectilinear projection was selected since it is rather popular in many VR applications. As shown, the stretching distortion and the perspective effect become more noticeable and annoying as the viewport FoV increases. Thus, it is important to find out which FoV should be used for viewport rendering that provides a good balance between immersivity and geometric distortion impact. This may improve the user's QoE.

1.2 Objectives

Considering that geometric distortions are introduced in the viewport during its rendering, due to the sphere to plane projection, and that:

- There are not many subjective quality assessment studies that assess the perpetual impact of those geometric distortions.
- There are not many subjective quality assessment studies that assess the effect of the FoV on the perceived viewport quality, and thus, with no clear evidence about the range of FoVs that should be used for viewport rendering.
- There is no objective quality metric able to automatically assess the impact of the geometric distortions in viewport images, and of predicting the viewport quality in the presence of those distortions.
- There are not many solutions that optimize the sphere to plane projections, aiming to minimize the geometric distortions of the viewport image.

The main objectives of this Thesis are:

1. Subjectively assess the perceptual impact of different types of geometric distortions, mainly stretching and bending, introduced in the rendering process.
2. Subjectively assess the FoV impact on the perceived quality of the viewport image, to determine the FoV that presents the best trade-off between user's immersive experience and the perceived visual degradations due to geometric distortions.
3. Develop content-aware objective quality metrics that automatically assess the geometric distortions in the viewport image.

4. Develop a procedure to optimize well-known projections used in the viewport rendering process, aiming to minimize the perceived geometric distortions, by adapting the projection to the image content.

1.3 Novel Contributions and Associated Publications

Following the main objectives defined above, research has been conducted to achieve them, with the following novel contributions:

1. Design and realization of subjective quality assessment studies targeting:

   i) The evaluation of the perceived geometric distortions when the general perspective and the Pannini projections are used for viewport rendering (see Chapters 3 and 5).

   ii) The evaluation of the FoV impact on perceived quality and to find the FoV that presents the best trade-off between user immersive experience and perceived geometric distortions, when the rectilinear projection is used for viewport rendering (see Chapter 3).

2. Design, implementation and assessment of two novel objective quality metrics:

   i) A content-aware metric to assess the perceptual impact of geometric distortions, in the rendered viewport. The metric relies on two sets of geometric distortion measures, namely, bending and stretching, that characterize the bending of straight lines and stretching of image regions (see Chapter 4).

   ii) An object-based metric to assess the perceptual impact of stretching objects shape, in the rendered viewport. The metric uses semantic segmentation to identify the relevant objects in the viewport and computes the stretching distortion for each object. Two distinct approaches were exploited and evaluated. The first one is based on shape measurements on the sphere and on the viewport, and the second is one based on Tissot indicatrices (see Chapter 5).

3. Design, implementation, and assessment of procedures for optimizing the projection parameters of the general perspective projection (GPP) and of the Pannini projection (PP), using the objective quality metrics proposed in Chapters 4 and 5. The optimum projection parameters are those that minimize the perceived geometric distortion in the viewport, adapting the projection to the viewport content (see Chapter 4 for GPP and Chapter 6 for PP).

The work developed in this Thesis led to the following publications:

1. **F. Jabar**, J. Ascenso, and M.P. Queluz, “Perceptual Analysis of Perspective Projection for Viewport Rendering in 360° Images,” *Proc. of IEEE International Symposium on Multimedia*, Taichung, Taiwan, Dec. 2017.

2. **F. Jabar**, M.P. Queluz, and J. Ascenso, “Objective Assessment of Line Distortions in Viewport Rendering of 360° Images,” *Proc. of the 1st IEEE International Conference on Artificial Intelligence and Virtual Reality*, Taichung, Taiwan, Dec. 2018.

3. **F. Jabar**, J. Ascenso, and M.P. Queluz, "Content-Aware Perspective Projection Optimization for Viewport Rendering of 360° Images," *Proc. of IEEE International Conference on Multimedia and Expo,* Shanghai, China, Jul. 2019.
4. **F. Jabar**, J. Ascenso, and M.P. Queluz, "Objective Assessment of Perceived Geometric Distortions in Viewport Rendering of 360° Images," *IEEE J. Sel. Top. Signal Process.*, vol. 14, no. 1, pp. 49-63, Jan. 2020, Doi: 10.1109/JSTSP.2019.2962970.
5. **F. Jabar**, J. Ascenso, and M.P. Queluz, "Field-of-View Effect on the Perceived Quality of Omnidirectional Images," *Proc. of the IEEE International Conference on Multimedia & Expo Workshops,* Athlone, Ireland, Jul. 2020.
6. **F. Jabar**, J. Ascenso, and M.P. Queluz "Object-Based Geometric Distortion Metric for Viewport Rendering of 360° Images", *IEEE Access,* vol. 10, no.1, pp. 13827-13843, Jan. 2022, Doi:10.1109/ACCESS.2022.3147699.
7. **F. Jabar**, J. Ascenso, and M.P. Queluz, "Globally and Locally Optimized Pannini Projection for Viewport Rendering of 360° Images", Submitted to J. Vis. Commun. Image Represent., Oct. 2022.

1.4 Structure of the Thesis

Considering the main objectives defined in Section 1.2, this Thesis is organized as follows:

- **Chapter 1** presents the context and motivation for this Thesis, the objectives and main novel technical contributions, and associated publications.
- **Chapter 2** presents the omnidirectional image (or video) system architecture, from acquisition to display, and reviews the state-of-the-art on sphere to plane projections, which constitute the main step in the viewport rendering process.
- **Chapter 3** describes the subjective evaluation studies performed to evaluate the perceptual impact of the geometric distortions and of the considered FoV for viewport rendering.
- **Chapter 4** introduces a novel content-aware objective quality metric to assess the perceptual impact of geometric distortions, on the rendered viewport. In this chapter, a procedure to globally optimize the general perspective projection (GPP), according to the viewport content - resulting in a content-aware GPP - is also proposed.
- **Chapter 5** introduces a novel object-based metric to assess the perceptual impact of objects stretching, on the rendered viewport.
- **Chapter 6** proposes procedures to optimize the Pannini projection for the viewport rendering with high FoV, resulting in a globally adapted Pannini projection (GA-PP), and on a globally and locally adapted Pannini projection (GLA-PP). While GA-PP aims to reduce the geometric distortions globally, in GLA-PP these distortions are globally and locally minimized.
- **Chapter 7** concludes the Thesis and points out some possible research paths for future work.

Chapter 2

State-of-the-Art on Sphere to Plane Projections

2.1 Introduction

Map projection is a process to flatten a spherical surface into a plane, to make a map; it involves the transformation of three dimensional (3D) coordinates, defining the spherical surface, into positions on a plane. Map projections have been used in cartography for a long time, to create a visual representation of the Earth’s surface on a plane (“Earth map”), which has been very useful for humans to navigate their way through the world. In the past, a large number of map projections have been developed by mathematicians and cartographers, trying to represent the Earth’s surface with correct shapes and dimensions. However, since a sphere is not a developable surface, all sphere to plane projections necessarily introduces geometrical distortions, by changing one or more geometric properties, such as distance, direction, shape or area; no sphere to plane projection can simultaneously maintain all these geometric properties. Thus, cartographers select a map projection according to the geometric property that should be preserved, at the expense of altering other properties. This means that the design of map projections is mostly focused on the characterization of geometric distortions.

Some of the sphere to plane projections developed for cartography in the past play a key role in the omnidirectional visual content delivery and thus, in the development of VR applications. Typically, to encode the omnidirectional visual content, the same coding standards of traditional 2D visual content are used and thus, a projection must be applied to obtain a planar representation of the spherical image. Furthermore, to display omnidirectional images or videos, a sphere to plane projection is applied to project a portion of the spherical image into a plane, resulting in the viewport image that is seen by the user. Therefore, the study, design and implementation of projections and their geometric distortions are very important for omnidirectional visual content processing. In particular, some projections may allow to save more bitrate for some target quality, or to improve the perceived quality of the viewports shown to users when navigating on this type of visual content.

The main objective of this chapter is to review the state of the art on sphere to plane projections; furthermore, to introduce some concepts related with the Thesis topic, a typical omnidirectional image/video system architecture is briefly described.

This chapter is organized as follows: Section 2.2 presents a typical omnidirectional image/video system architecture, from acquisition to display. Section 2.3 puts forward three distinct classifications of sphere to plane projections, considering: *i*) the used developable surface type; *ii*) the content characteristics that are preserved by the projection; *iii*) the projection content awareness, which is a new classification proposed in this Thesis. Sections 2.4 and 2.5 describe, respectively, the state-of-the-art on content-unaware and content-aware projections. Section 2.6 overviews related work on the subjective assessment of omnidirectional images and videos, and on metrics to objectively assess the geometric distortions resulting from sphere to plan projections. Section 2.7 concludes this chapter with some final remarks.

2.2 Omnidirectional Image System Architecture

Figure 2.1 depicts a typical omnidirectional image/video system architecture which illustrates how omnidirectional visual content is processed, from acquisition to display, i.e., the main processing steps. This type of architecture is rather suitable for many applications, but especially when omnidirectional images and videos need to be consumed by a wide range of users.

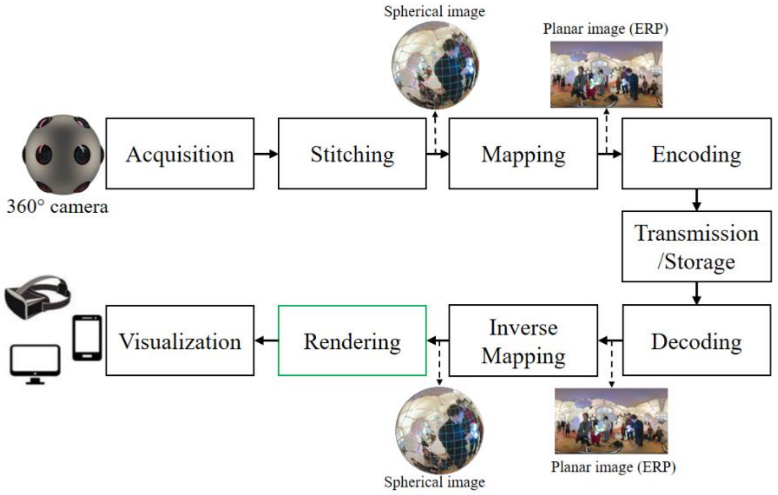


Figure 2.1. Omnidirectional image and video system architecture.

In the following, each step is briefly described:

1) Acquisition - Several acquisition devices have been developed in the past to capture omnidirectional visual content, from single camera systems to large arrays of cameras. Typically, when a single camera is used, a wide-angle or fisheye lens captures a large field of view, which is sufficient for some applications. However, a full 360° field of view, with high resolution, is difficult to obtain with such a device, and thus rotative arms or other optical arrangements (e.g., mirrors) may be used to acquire multiple images from different directions; however, this solution has the drawback of not capturing the entire environment at the same time [13]. Nowadays, the acquisition of omnidirectional visual content is typically done with a camera array, e.g., Facebook surround 360 [14] and GoPro Odyssey [15]. In this case, multiple synchronized cameras are embedded into a single device, where each camera is oriented to a different direction and acquires a fraction of the omnidirectional visual content. This allows the representation of the whole 360° (horizontal) × 180° (vertical) viewing range, providing a better acquisition than single camera devices. During acquisition, some visual distortions (or artifacts) may be introduced, such as optical distortions, Moiré effect, noise, and motion blur, that come mainly from individual camera sensors [16].

2) Stitching - After the acquisition, images from different cameras are stitched together to obtain a spherical representation of the visual scene, which is usually referred to as a viewing sphere (VS). Stitching is a process that aligns several images that were acquired from different viewing directions of the same visual scene, typically with some overlapping regions, e.g., 50% [17]. The stitching process has two fundamental steps: registration and blending. Registration is defined as the geometric alignment process between adjacent images. After, a blending process is performed to allow a smooth transition between the adjacent images that constitute the visual sphere. Some artifacts may be noticed in the output VS image, due to misregistration, different lighting conditions among images during acquisition, and object motion during acquisition. These artifacts can be visually annoying and may include blurring, visible seams, ghosting and broken edges, and even some geometric distortions (visible deformation on objects or part of it) [16].

3) Mapping - After stitching, the visual sphere is projected on a plane, to obtain a planar representation required by most image or video codecs (being it standard codecs, or not). Several projections, such as equirectangular projection (ERP) [18], cube map projection (CMP) [19], and pyramid projection [20], can be used for this purpose. Since the projection from the spherical to the planar representation (and the back projection to the spherical domain at the client side) involves resampling and interpolation, some visual distortion may be introduced, such as aliasing, blurring and ringing. Also, the coding performance is influenced by the used projection [20][21]. The coding efficiency of several projections has been already evaluated, and the hybrid equi-angular cube map projection (HEC) was identified as the best one [22].

4) Encoding - To store and transmit the omnidirectional visual content, state-of-the-art 2D video compression schemes, such as H.264/AVC [23], HEVC [24], and VVC [25], are often used to compress the visual data; the encoder input corresponds to the planar representation obtained in the previous step. Several artifacts, such as blocking, blurring, staircase, flickering, ringing, among others, are introduced in the planar image due to compression [16]. The visibility of these artifacts may also vary according to the projection used in the previous step. In recent years, several coding optimization tools have been developed to improve the coding efficiency for omnidirectional images [26]–[28].

5) Transmission - In this step, the content is transmitted from the sender to the receiver; this can be achieved using different approaches. The simplest one corresponds to the transmission of the entire visual sphere, represented in a 2D format (thus, as a 2D image), being mainly used for omnidirectional still images, and not so for videos, since it requires a large transmission bandwidth. For omnidirectional videos, more efficient approaches exist, e.g., viewport-adaptive streaming [29]–[31] and tile-based streaming [32]–[34]. The main idea of these approaches is to transmit the viewport that is observed by the user, at any time, with higher quality, and the remaining regions with lower quality, thus reducing the data rate without impact on quality. To enable the easy deployment of interoperable standardized streaming services for omnidirectional videos, the MPEG group has developed the Omnidirectional MediA Format (OMAF) - MPEG-I [1]: part 2. Depending on the streaming solution, different factors may influence the user’s QoE, such as transmission delay, rebuffering events, and video quality fluctuation [16][35].

6) Decoding - The decoding step performs the inverse operation of the encoder at the receiver side, to decompress the visual content.

7) Inverse Mapping - For rendering the omnidirectional visual content, a spherical representation is often used. Therefore, the transmitted planar content must be mapped into a sphere again, by applying the corresponding inverse mapping transformation.

8) Rendering - The rendering is the process of producing a visual representation that can be consumed by the users, using some suitable device. This process consists on the projection of a selected part of the spherical image - corresponding to the region observed by the user - onto a plane, resulting in the viewport that is shown to the users, as depicted in Figure 1.3b). In the omnidirectional image/video processing pipeline depicted in Figure 2.1, two different sphere to plane projections are used, one before encoding and another for rendering. The former is needed to represent the visual sphere, in a 2D format, to be used as input for the encoder, and the latter to obtain the viewport. The projection used before encoding should represent the visual sphere, in a 2D format, in the best way from the point of view of compression efficiency. However, the projection used for rendering should represent a selected part of the viewing sphere on a plane, with the best perceived quality, minimizing the subjective impact of geometric distortions. Perspective projections, e.g., rectilinear or stereographic, are often used for viewport rendering of omnidirectional content. The rectilinear keeps all straight lines in the visual scene also straight after projection but stretches objects shape. Stereographic preserves the object shapes locally but bends the straight lines (fisheye effect). There are some other projections proposed to map wide-angle images onto a plane (e.g., [36][37]), or for omnidirectional image rendering [10], but no projection can avoid geometric distortions. The perspective projections, and some of the recently proposed projections, will be described in Sections 2.4 and 2.5.

9) Visualization - In this last step, the rendered viewport is sent to the display device. Several display types can be used, including smartphones (or tablets), standard computer monitors, or head-mounted displays (HMDs). In a 2D monitor, the users can navigate on the omnidirectional visual content by changing the viewing direction through mouse movements or any other interactive device. In a smartphone, the viewing direction changes with the direction in which the smartphone is pointing to; the users can simply change the smartphone direction in the physical space to watch any part of omnidirectional visual content. The HMD is a wearable device worn on the head and has two displays close to the user's eyes. It has sensors to track the user's head movements, allowing the users to watch the different parts of the omnidirectional visual content, usually with 3-degree of freedom (yaw, pitch, roll), by moving their head. Typically, an HMD provides a more immersive experience compared to other visualization devices.

2.3 Classification of Sphere to Plane Projections

Several sphere to plane projections are available in the literature, which were designed along the years for different purposes and using different approaches, starting with the oldest cartographic Earth's map representations [38], to the more recent mapping of wide-angle images [39] and the rendering of omnidirectional visual content [9][10]. Although the term "projection" is used to describe the various transformations that enable the representation of a spherical surface on a planar map, not all are true projections, in the geometric sense of the word. Indeed, there are two broad classes of projections: true projections, and those that are solely based on mathematical transformations. In the former, points on the sphere can be projected on a plane using projection lines, linking the projection center to the point to be projected; in the latter case, this projection geometry cannot be drawn.

Classically, sphere to plane projections are classified according to the projection surface type or according to the geometric properties that are preserved after projection [40]. However, in recent years, several new sphere to plane projections were developed - mainly for photography and omnidirectional image or video - allowing to devise an additional classification, according to the influence of the sphere content, on the projection. Accordingly, the sphere to plane projections can be classified based on the following dimensions:

1) **Classification by projection surface** - The projection is classified based on the developable surface in which the spherical surface is projected onto; it applies only to true projections. The developable surface is a surface that can be unwrapped (or unfolded) into a plane without stretching, tearing, or shrinking. The most used developable surfaces are cylinder, cone, and plane. Most of the projections use a single developable surface; however, there are some projections that use a mix of developable surfaces.

2) **Classification by preserved geometric properties** - The projection is classified based on the geometric properties that are preserved. In cartography, the most important properties to be preserved are area, direction, shape, and distance. These properties are also important for human perception, and they need to be preserved as much as possible.

3) **Classification by content awareness** - The projection is classified according to its dependency on the content. The projection is content-aware if the projection procedure (e.g., the projection equations and/or parameters) depends on the content that is being projected; otherwise, it is content-unaware. In the latter, there is a univocal correspondence between the positions on the sphere and the resulting positions on the plane, after projection, regardless of the image content.

The following sections detail the projection classes introduced above.

2.3.1 Classification by Developable Surface

Figure 2.2 depicts sphere to plane projections using the three most common developable surfaces, i.e., cylindrical, conical, and planar.

Regarding this classification dimension, the following sub-classes can be identified:

- **Cylindrical projections** - In cylindrical projection, a cylinder is wrapped around the sphere and the spherical surface is projected on the cylinder surface; after, the cylinder “is cut” along one of the meridians and unwrapped to obtain the final cylindrical projection, as in Figure 2.2. Depending on how the cylinder is placed relative to the sphere, the cylindrical projection can be normal, transverse, or oblique. The normal case is suited for making a map where equatorial regions are less distorted, and the transverse case for presenting north and south regions with less distortions. Furthermore, the cylindrical projection can be tangent, when the cylinder surface touches the sphere along a circular line (also called as standard line), or secant, when the cylinder surface slices the sphere surface and touch the sphere on two standard lines. After projection, the regions near the standard line (one line in the tangent case, or two lines in the secant case) have the lowest distortions, and the distortion increases as the distance to the standard line increases. The secant case is used when it is required to reduce the distortions for some regions in the map. As an important propriety of all cylindrical projections, parallels and meridians result in straight vertical and horizontal lines, respectively (thus, perpendicular to each other) and with meridians of constant spacing (for meridional intervals of constant spacing in the sphere). Besides being often used in Earth mapping, cylindrical projections have been also exploited for wide-angle or panoramic images, and more recently for omnidirectional images. Some examples of cylindrical projection are detailed in Section 2.4.

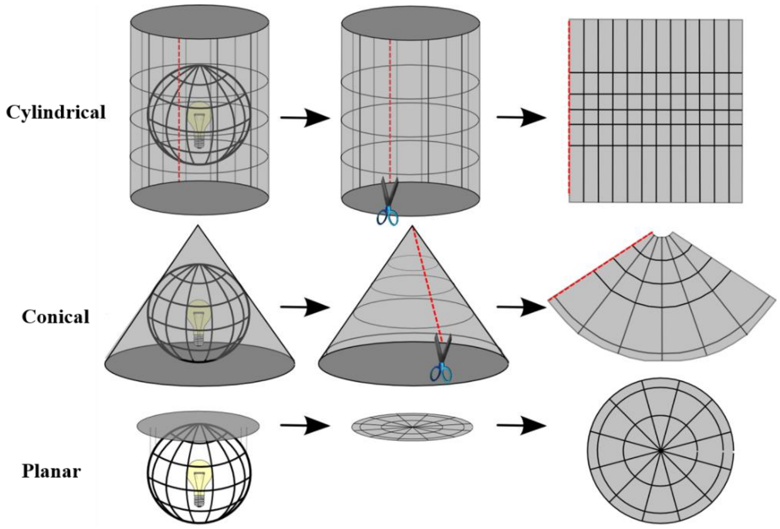


Figure 2.2. Projections based on a developable surface: cylindrical, conical, and planar (based on [41]).

- **Conical projections** - In conical projections, a cone is wrapped around the sphere and the sphere surface is projected onto the cone surface; after, the cone "is cut" along one of the meridians and unwrapped in a plane to produce the final conical projection, as in Figure 2.2. The cone touches the sphere in just one standard line (or parallel line), in the tangent case, and on two standard lines (also parallel lines), in the secant case. The meridians result in straight lines, meeting at the center point (point located at the center of the map), and parallels result in circular arcs centred on the centre point. Figure 2.3 presents an example of an Earth map resulting from a conical projection. Conical projections are suited for hemispheric maps of Earth, but not for a complete Earth map, neither for photography nor omnidirectional images.

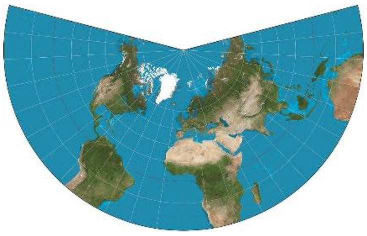


Figure 2.3. Lambert conformal conic Earth map [42].

- **Planar projections** - The planar projection projects the sphere onto a plane which is tangent to the sphere at (or secant near) the poles (polar aspect, *cf.* Figure 2.2), the equator (equatorial aspect), or some points in-between (oblique aspect). Planar projections may also use more than one plane as a developable surface, e.g., multiple plane projections [43] and cube map projection [19]. The latter, and besides ERP, is also often for mapping omnidirectional visual content before coding. Planar projections are also often used in photography and for the rendering of omnidirectional visual content. Relevant planar projections are detailed in Sections 2.4 and 2.5.

- **Hybrid projections** - In this case, the projection uses more than one developable surface type, for example, cylinder and plane. Typically, this projection is done in two steps: first, projection of the sphere surface onto the first developable surface (e.g., cylinder); second, projection of the first developable surface onto the second surface (e.g., plane). This type of projection has been developed in recent years for reducing the geometric distortions in wide-angle or panoramic images. Examples of this projection are proposed in [37][44]. In [37], the sphere surface is first projected onto a swung surface, followed by the projection of the swung content onto a plane. The swung surface is created using a 2D profile curve which is rotated around an axis of revolution (more details are provided in Section 2.5.7). In [44], the sphere surface is projected onto a cylindrical surface, and the content on the cylinder surface is then projected onto a plane. These projections are detailed in Sections 2.5.7 and 2.4.7, respectively.

An important property that is common to any of the aforementioned projection types, is that the geometric distortion is less near the points, or lines, where the developable surface touches or intersects the sphere. Accordingly, the surface placement relatively to the sphere will directly affect the map positions with the highest and the lowest amount of distortion, and its choice is dependent on the sphere regions, and directions, where the geometric proprieties need to be preserved.

2.3.2 Classification by Preserved Characteristics

Since a spherical surface is an undevelopable surface, no sphere to plane projection can simultaneously maintain all geometric properties; in cartography, area, distance, shape, and direction are the four most important properties that should be preserved [45]. The projection preserves the area if the area relationships before and after projection are maintained; however, no projection can preserve the object shapes and their area at the same time, i.e., area and shape are mutually exclusive. The projection preserves the distance, if the relationships of the distances before and after projection are maintained; however, no projection can maintain the distance for all projected points. The projection preserves the direction (where direction, also called azimuth, is measured in degrees relatively to the geographical north) if the directions from any point to every other point before and after projection are kept; no projection can have the correct direction for all projected points.

Considering the most recent applications, such as wide-angle images and omnidirectional image rendering, besides the aforementioned properties the projection should also keep the straightness of the lines that are also straight in the visual scene, since the human perception is highly sensitive to the distortion of those lines. In the omnidirectional representation of the scene, straight lines in the visual scene are always over great circles of the sphere, and because the arc of a great circle between two points is the shortest surface path between them. Accordingly, the projection preserves the straightness of the lines if any great circle is projected as a straight line.

Depending on the preserved properties, a projection can be classified as:

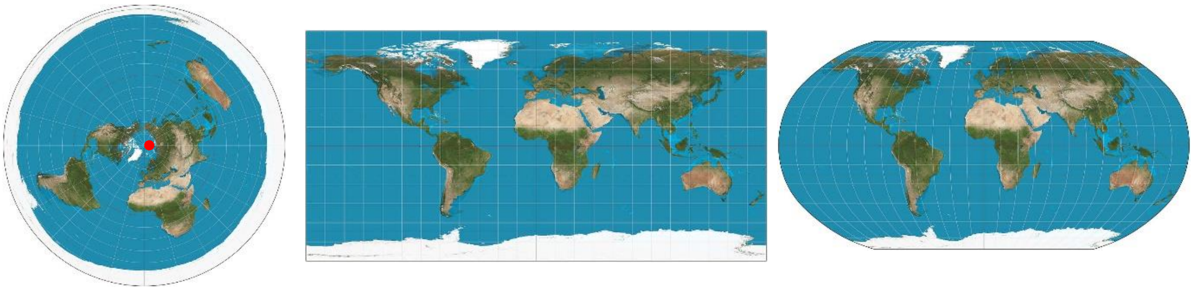
- **Conformal** - Conformal projections are designed to maintain conformality, meaning that the shapes (and the angles) are locally preserved; also, the distance is preserved in all directions around a projected point. Since the angles are locally preserved, when a conformal projection is used to create the Earth’s map the projected parallel and meridian intersect at 90° angles on the plane. Small areas, e.g., the area of a small city, are mapped with the correct shape; however, large areas, e.g., a continent, have a wrong shape. The projection depicted in Figure 2.3 is a conformal projection. Mercator is another well-known conformal projection which is fully described in Section 2.4.5.

- **Equidistant** - Equidistant projections are designed to maintain the distance along one or more lines or from one or two points to all the other points on the plane. However, it is not possible to maintain the correct distance for all projected points on the plane. Figure 2.4a) depicts an equidistant projection, where the distance along a line from the central point (red point in Figure 2.4a) to any other point is preserved. Note that no projection can maintain the distance to and from all points on a map. The equirectangular projection (ERP), described in Section 2.4.3, is another example of an equidistant projection.

- **Azimuthal** - It corresponds to a planar projection, having the projection plane tangent to the sphere. Because directions from the point of tangency are preserved, these projections are also known as *azimuthal.* Furthermore, all great circles that cross the tangency point are projected as straight lines [46]. The most popular azimuthal projections are orthographic, stereographic, and rectilinear (or gnomonic) projections. Rectilinear and stereographic are typically used for rendering omnidirectional visual content and are detailed in Section 2.4.6.

- **Equal-area** - Equal-area projections are designed to maintain the relative area of regions before and after projection, at the expense of distorting other properties, such as shape, angle, and/or distance. An equal-area projection can be equidistant, but never conformal [46]. Figure 2.4b) depicts the Lambert equal-area projection, which is fully described in Section 2.4.4.

- **Balanced** - Balanced projections are designed such that no specific property is preserved, but a balance between different properties is obtained. This means that instead of preserving the shape, area, angle, or distance, extreme distortions are avoided in any of the geometric properties. Figure 2.4c) presents the Robinson projection, which is a balanced one. Examples of this case are described in Sections 2.4.6, 2.4.7, and 2.5.



a) b) c)

Figure 2.4. a) Azimuthal equidistant projection [47]; b) Lambert cylindrical equal-area projection [48]; c) Balanced Robinson projection [49].

2.3.3 Classification by Content Awareness

Content awareness is an important dimension of the projection classification, especially considering new applications such as wide-angle photography or omnidirectional image rendering, where the projection procedure may depend (or not) on the content being projected. In this sense, a projection can be classified as:

- **Content-unaware** - A content-unaware projection projects the spherical surface (or a part of it) on the plane without considering the content characteristics, i.e., the projection is not adapted to the content; however, for the most relevant applications, such as visualizing wide-angle images or rendering omnidirectional visual content, this may lead to images with very noticeable geometric distortions (e.g., stretching), and with an unpleasant quality. Content-unaware projections are always applied in the same way and thus cannot be used to mitigate the perceptual impact of some geometric distortions. Several content-unaware projections are detailed in Section 2.4.

- **Content-aware** - Content-aware (also known as content-preserving, content-dependent, or content-adaptive) projections consider the image content, to preserve the visual properties of some image regions and structures. In this case, the projection is adapted to the content, e.g. guaranteeing that straight lines are kept straight in the projected image. Recently, several content-aware projections have been proposed in the literature. Some of these projections are detailed in Section 2.5.

Due to the importance of this dimension and its relevance for the objectives of this Thesis, the state-of-the-art next described is organized in the two projection classes defined in this section.

2.4 Content-unaware Projections

Several content-unaware projections were developed in the past for cartography purposes; however, some of them became popular, and are used, in recent applications. Table 2.1 summarizes the content-unaware projections most used for mapping wide-angle images, or to perform viewport rendering of omnidirectional visual content [8]–[10][39][50]. They are classified according to the used developable surface (for true projections) and preserved characteristic, as described before. Moreover, the typical application of each projection is also presented. The rectilinear, stereographic, and orthographic projections are particular cases of the general perspective projection (GPP), and the Pannini projection was proposed in [44], for mapping wide-angle images.

Table 2.1. Selected content-unaware projections and their classification by projection surface (CPS), by preserved characteristic (CPC), and typical application.

Projection	CPS	CPC	Typical application
Central Cylindrical	Cylindrical	Azimuthal	Wide-angle photography, architectural photography
Equirectangular	n.a. (Cylindrical-type)	Equidistant	Cartography, wide-angle photography, mapping omnidirectional visual content
Lambert - Equal Area	Cylindrical	Equal-area	Cartography, wide-angle photography, mapping omnidirectional visual content
Mercator	n.a. (Cylindrical-type)	Conformal	Cartography, wide-angle photography
Transverse Mercator	n.a. (Cylindrical-type)	Conformal	Cartography, wide-angle photography
Rectilinear	Planar	Azimuthal	Cartography, photography, architectural photography, omnidirectional visual content rendering
Stereographic	Planar	Azimuthal, Conformal	Cartography, photography, omnidirectional visual content rendering
Orthographic	Planar	Azimuthal	Cartography
Pannini	Hybrid	Balanced	Wide-angle photography, omnidirectional visual content rendering

The following sections describe and analyze the projections referred to in Table 2.1. The reference coordinate systems, used to formalize the projections, are first introduced.

2.4.1 Reference Coordinate Systems

Consider the sphere depicted in Figure 2.5, with 3D Cartesian coordinates (X, Y, Z), centered at point O and with unit radius. Each point on the sphere can also be defined by the longitude (ϕ), with origin on the Z-axis and with a range $[-\pi, \pi]$, and by the latitude (θ), with origin on the XZ plane and with a range $[-\pi/2, \pi/2]$.

The spherical to 3D Cartesian coordinates transform can be described by

$$X = \cos(\theta)\sin(\phi) \tag{2.1}$$
$$Y = \sin(\theta) \tag{2.2}$$
$$Z = \cos(\theta)\cos(\phi) \tag{2.3}$$

and the 3D Cartesian to spherical coordinates transform is given by

$$\phi = \tan^{-1}\frac{X}{Z} \tag{2.4}$$
$$\theta = \tan^{-1}\frac{Y}{\sqrt{X^2 + Z^2}}\ . \tag{2.5}$$

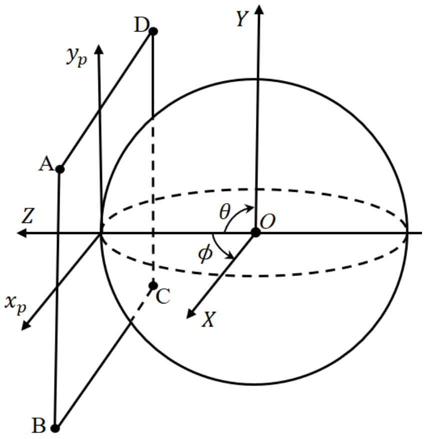


Figure 2.5. Sphere and projection plane coordinate systems.

The projection plane, denoted as ABCD in Figure 2.5, has 2D Cartesian coordinates (x_p, y_p), is perpendicular to the Z-axis and is tangent to the sphere at $Z = 1$.

Consider a point on the sphere, with spherical coordinates (ϕ, θ); this point is projected on the plane using the forward projection, $(x_p, y_p) = Proj(\phi, \theta)$; inversely, for each projected point on the plane, the corresponding point of the sphere can be obtained using the backward projection, $(\phi, \theta) = Proj^{-1}(x_p, y_p)$.

2.4.2 Central Cylindrical Projection

In central cylindrical projection (also known as perspective cylindrical projection) a vertically positioned cylinder is wrapped around a unit radius sphere, tangent along the equator line, and the sphere surface is projected on the cylinder surface from a perspective point (or projection centre) located at the sphere centre (*O*), as depicted in Figure 2.6a). After, the cylinder is cut vertically at Z= -1, and unwrapped to obtain the final projection, as depicted in Figure 2.6b).

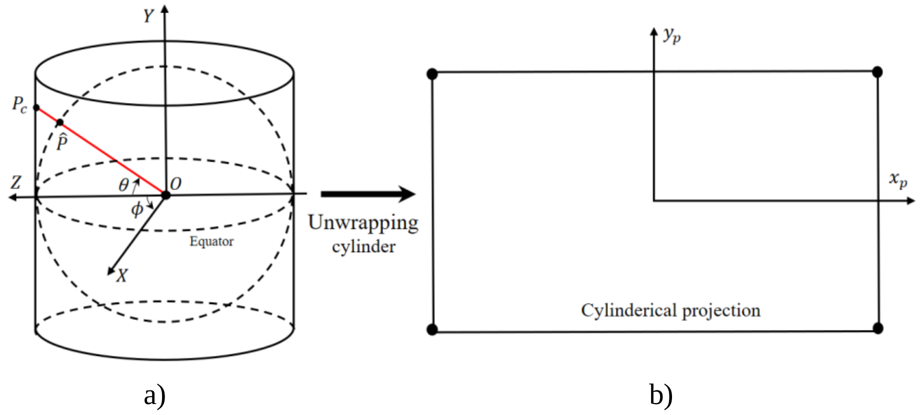


Figure 2.6. a) Central cylindrical projection geometry; b) Cylindrical projection after unwrapping the cylinder.

The forward projection is formally described by

$$x_p = \phi \tag{2.6}$$

$$y_p = \tan(\theta) \tag{2.7}$$

and the backward projection is given by

$$\phi = x_p \tag{2.8}$$

$$\theta = \tan^{-1}(y_p)\,. \tag{2.9}$$

This projection keeps the vertical lines on the visual scene (lines coincident with the meridians on the sphere) straight after projection, but the horizontal lines are bent, except those that coincide with the equator line. Also, the stretching (in the horizontal direction) increases from the equator to the poles. This projection is often used for panoramic images with a large HFoV, but should not be used if the VFoV is also large (due to stretching) [39].

2.4.3 Equirectangular Projection

The equirectangular projection (ERP), also known as equidistant cylindrical projection or plate carrée, is not a true projection (in the geometric sense of the word), directly mapping the latitude and longitude coordinates of the sphere, on the horizontal and vertical coordinates of the plane. The forward and backward projection equations are simply described by

$$\phi = x_p \tag{2.10}$$

$$\theta = y_p\,. \tag{2.11}$$

This projection keeps the vertical lines straight, but the horizontal lines are bent, except those that coincide with the equator line. Also, the objects close to the sphere poles are projected with significant stretching in the horizontal direction. Since this is common to the cylindrical central projection, the ERP projection is considered as a cylindrical-type projection [46]. This projection is often used for mapping omnidirectional visual content before coding [20].

2.4.4 Lambert Cylindrical Equal-area Projection

The Lambert equal-area projection is a cylindrical projection, invented by Johann Heinrich Lambert in 1772. Like in the central cylinder projection, it projects the sphere on a cylindrical surface that is wrapped around the sphere, tangent along the equator line. After, the cylinder is unwrapped to obtain the final projection. The main difference to the central cylindrical projection is that the sphere is projected on the cylinder surface using projection lines parallel to the equator plane.

The forward projection is described by

$$x_p = \phi \tag{2.12}$$

$$y_p = \sin(\theta) \tag{2.13}$$

and the backward projection is

$$\phi = x_p \tag{2.14}$$

$$\theta = \sin^{-1}(y_p)\,. \tag{2.15}$$

As the name suggests, this projection maintains the relative area of the objects. Also, it has the same effect on the vertical lines as the previous projections, and also introduces horizontal stretching. It is often useful for wide-angle or panoramic images, but less useful for panoramic images with a large VFoV (due to stretching) [39]. It has been also used for mapping omnidirectional visual content before coding [20].

2.4.5 Normal and Transverse Mercator Projections

The Mercator projection was developed by the geographer and cartographer Gerardus Mercator, in 1569, and it is not a true projection, although being derived from the central cylindrical projection. It is one of the most used projections in Earth cartography, mainly because the projection along the y-axis coordinates is defined to guarantee conformality, and to represent lines of constant course, known as *rhumb* lines, as straight segments that conserve the angles with the meridians; this makes it quite useful for nautical navigation.

The Mercator forward projection is described by

$$x_p = \phi \tag{2.16}$$

$$y_p = \ln\left(\tan\left(\frac{\theta}{2} + \frac{\pi}{4}\right)\right) \tag{2.17}$$

and the backward projection is given by

$$\phi = x_p \tag{2.18}$$

$$\theta = 2\left[\tan^{-1}\left(\exp(y_p)\right) - \frac{\pi}{4}\right]. \tag{2.19}$$

Being a cylindrical-type projection, the vertical lines are straight, but the horizontal lines are bent, except the horizontal lines that coincide with the equator line. Furthermore, the Mercator projection increases the relative area of objects as the latitude increases, generating area distortion. Accordingly, it is useful for panoramic images with a large HFoV, but is not a good solution for images with a large VFoV. The transverse Mercator projection is similar to the normal Mercator projection, except that it derives from a cylindrical projection where the cylindrical surface is horizontally oriented and tangent to the prime meridian (meridian line at $\phi = 0$). In this case, the forward projection is described by

$$x_p = \tanh^{-1}[\sin(\phi)\cos(\theta)] \tag{2.20}$$

$$y_p = \tan^{-1}[\sec(\phi)\tan(\theta)] \tag{2.21}$$

and the backward projection is given by

$$\phi = \tan^{-1}\left[\sinh(x_p)\sec(y_p)\right] \tag{2.22}$$

$$\theta = \sin^{-1}\left[\operatorname{sech}\left(x_p\right)\sin(y_p)\right]. \tag{2.23}$$

The horizontal lines are straight, but the vertical lines are bent (except the vertical lines that coincide with the prime meridian line). Like the normal Mercator, the transverse Mercator is a conformal projection, i.e. the object shapes are locally preserved. However, the relative area of the objects increases as the longitude increase, generating area distortion. Thus, the transverse Mercator projection may be a good solution to project spherical content with a large VFoV [39].

2.4.6 General Perspective Projection

In the general perspective projection (GPP), points on the sphere are projected on a plane tangent to the sphere, using projection lines emanating from the projection center, located on the Z-axis and at a distance d from the sphere center (*O*), as depicted on Figure 2.7 (d is accounted towards the negative sense of Z); in this figure, *P* is the projection center, and $\hat{P}$ is a point on the sphere projected as a point $\bar{P}$ on the plane.

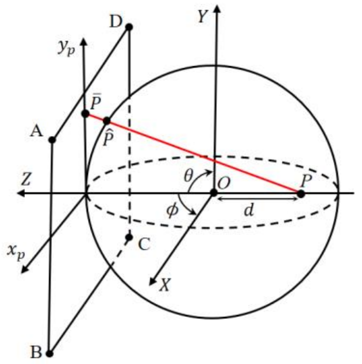


Figure 2.7. General perspective projection (GPP).

The GPP forward projection is given by

$$x_p = (1+d)\frac{\cos(\theta)\sin(\phi)}{\cos(\theta)\cos(\phi)+d} \tag{2.24}$$

$$y_p = (1+d)\frac{\sin(\theta)}{\cos(\theta)\cos(\phi)+d} \tag{2.25}$$

and the backward projection is

$$\phi = \tan^{-1}\frac{q\,x_p}{q(1+d)-d} \tag{2.26}$$

$$\theta = \tan^{-1}\frac{q\,y_p}{\sqrt{(q\,x_p)^2+(q(1+d)-d)^2}} \tag{2.27}$$

where q is given by

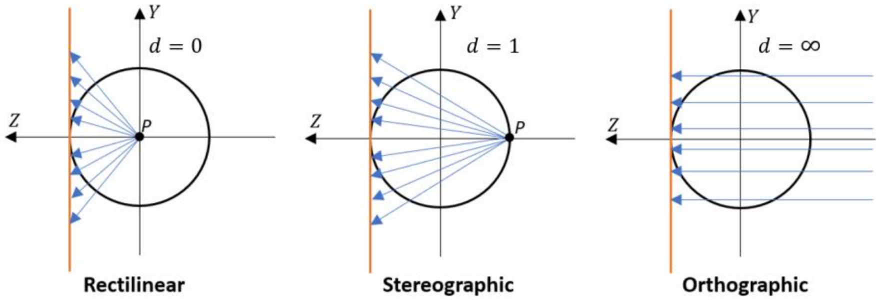
$$q = \frac{d(d+1)+\sqrt{\left(x_p^2+y_p^2\right)(1-d^2)+(d+1)^2}}{x_p^2+y_p^2+(d+1)^2}. \tag{2.28}$$

Different perspective projections can be obtained by moving the projection center along the Z-axis, i.e., by varying the value of d in Figure 2.7. The popular ones are defined below:

i) **Rectilinear** - The rectilinear projection has a projection center located in the center of the sphere i.e., $d = 0$, as depicted in Figure 2.8a). This projection is often used in photography and for the rendering of omnidirectional images [9][8]. Since all great circles are projected as straight lines [51], straight lines in the visual scene are also straight after projection (i.e., lines are not bent), which creates a perceptually appealing viewport when used for omnidirectional image rendering. However, stretching is present and increases with the distance to the tangency point; this stretching effect is even noticeable for FoVs which are not too large, e.g., 100°. Due to the stretching effect, this projection is less suitable for applications that require a large FoV.

ii) **Stereographic** - The stereographic projection has a projection center located on a position opposite to the tangency point between the sphere and the projection plane i.e., $d = 1$, as depicted in Figure 2.8b). This projection is conformal, being object shapes locally preserved. In this projection, the vertical and horizontal lines are bent, except for straight radial lines crossing the tangency point, which straightness is kept (these lines correspond to great circles crossing the tangency point). The bending of horizontal and vertical lines creates an effect known as a fisheye effect, making it less suitable for many applications, e.g. wide-angle photography. This projection is also used for the rendering of omnidirectional content [9], but much less often than the rectilinear one.

iii) **Orthographic** - The orthographic projection has the projection center located at infinity i.e., $d = \infty$, as in Figure 2.8c), resulting on projection lines that are orthogonal to the projection plane. It is neither equal-area nor conformal, thus objects are mapped with significant distortion in terms of area and shape. Like the stereographic projection, only the lines over great circles that cross the tangency point are projected without bending. The other lines are much more bent than in the stereographic projection; thus, it is hardly used in photography and not used at all for omnidirectional image rendering.



a) b) c)

Figure 2.8. Illustration of three well-known GPP instances: a) rectilinear; b) stereographic; c) orthographic.

2.4.7 Pannini Projection

The Pannini projection (PP) was proposed in [44] to map wide-angle images into a flat surface. This projection was derived from an analysis of a painting style popular during the 18th century called as *vedutismo* - the art of painting a highly detailed visual scene with a wide field of view without visible geometric distortions. The PP was named after Italian *vedutisti* painter, Giovani Paolo Pannini.

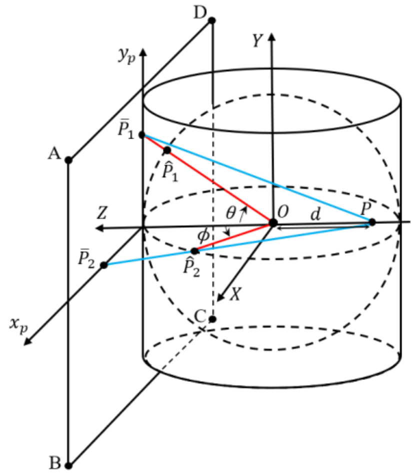


Figure 2.9. Pannini projection of two points, $\widehat{P}_1$ and $\widehat{P}_2$. The red lines project the points from the sphere to the cylinder surface; the blue lines project the points from the cylinder surface to the plane.

Consider the Pannini projection represented in Figure 2.9, showing a vertically oriented cylindrical surface, whose axis coincides with the Y-axis, tangent to the sphere along the equator line, and the projection plane ABCD. Points $\hat{P}$ on the sphere are projected on the plane in two steps: *i)* projection from the sphere surface to the cylindrical surface with a projection center located in the center of the sphere (red lines in Figure 2.9), as in a central cylindrical projection; *ii)* projection from the cylindrical surface to the plane, with a certain value of d (blue lines in Figure 2.9).

In this projection, all vertical and radial lines (i.e., those that cross the tangency point) are kept straight, while other line orientations (including horizontal lines) are bent. To reduce the bending of horizontal lines, a vertical compression transformation (VC) can be applied, at the expense of bending the radial lines and/or stretching some image regions [44].

The Pannini forward projection is given by

$$x_p = \mathrm{S}\sin(\phi) \tag{2.29}$$

$$y_p = (1 - vc)(\mathrm{S}\tan(\theta)) + vc\left(\frac{\tan(\theta)}{\cos(\phi)}\right) \tag{2.30}$$

where

$$S = \frac{d + 1}{d + \cos(\phi)}, \tag{2.31}$$

and $vc \in [0,1]$ is the vertical compression factor value. Several projections can be obtained by varying d in the range [0,1], namely rectilinear projection, when $d = 0$, and stereographic Pannini, when $d = 1$. Varying vc from 0 to 1, enforces the horizontal lines to be less bent; however, all radial lines become bent and/or image regions are stretched. Setting $vc = 0$ the Pannini projection is referred to as basic Pannini.

The Pannini backward projection is described by

$$\phi \ = \tan^{-1}\left(\frac{x_p}{S\cos(\check{\phi})}\right) \tag{2.32}$$

$$\theta = \tan^{-1}\left(y_p \,/\left[(1-vc)S + \frac{vc}{\cos(\check{\phi})}\right]\right) \tag{2.33}$$

where (2.34)

$$\cos(\check{\phi}) = \frac{-kd + \sqrt{k^2d^2 - (k+1)(kd^2-1\,)}}{k+1} \tag{2.35}$$

$$k = \frac{x_p^2}{(d+1)^2}\,. \tag{2.36}$$

2.4.8 Content-unaware Projections Qualitative Evaluation

In this section, a qualitative comparison between the projections listed in Table 2.1, and described in the previous sections, is presented. The viewports were rendered from the same omnidirectional image (presented in Figure 2.10 in equirectangular format), taken from the Salient360! Dataset [52], with a spatial resolution of 7500×3750 pixels, and for the same viewing direction. The horizontal FoV was set to 110°, and the viewport spatial resolution was set to 856×856 pixels (aspect ratio of 1), as recommended in [53]. Since the projections were described in the continuous spatial domain, and the viewports are defined in the discrete spatial domain, the conversion from these two domains - required for the viewport rendering - is firstly described.

Figure 2.10. Omnidirectional image used for producing viewports with different projections.

A. Viewport Rendering Procedure

Consider a point on the sphere, with spherical coordinates (ϕ,θ), that is projected on a viewport point, with cartesian coordinates (x_p,y_p), using the forward projection $(x_p,y_p) = Proj(\phi,\theta)$; thus, a point on the viewport image, (x_p,y_p), can be projected onto a sphere point, (ϕ,θ), using the backward projection $(\phi,\theta) = Proj^{-1}(x_p,y_p)$. Let consider the spherical coordinate system presented in Figure 2.5 and, for this initial description, that the front viewport is being observed; this viewport corresponds to the viewing direction $(\phi = 0, \theta = 0, \psi = 0)$, where ψ is the rotation angle around Z-axis (i.e., roll angle). Consider the spherical image and the front viewport, as depicted in Figure 2.11a), that is projected on the plane, resulting in a viewport image depicted in Figure 2.11b). In this figure, (x_p,y_p) are the 2D image plane coordinates, in length units; (u,v) are the 2D image sampling coordinates, also in length units; the image sampling points are represented by orange dots; (m,n), are the pixel position coordinate*s* corresponding, respectively, to the columns and rows of the viewport image.

The viewport position (m,n) can be rendered by applying the following four steps:

1) **Compute the viewport size** - Consider Figure 2.11a), and a point, $\hat{P}$, on the sphere that is positioned at the limit of the vertical field of view, F_v, thus has spherical coordinate

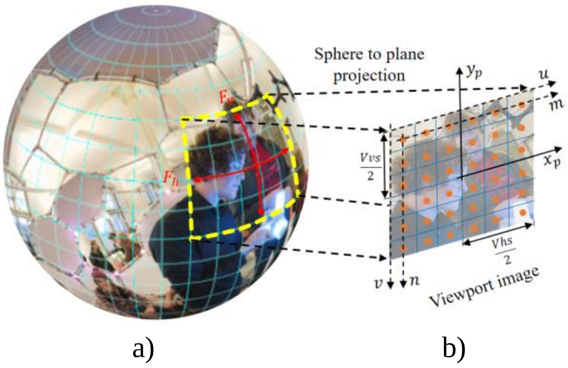


Figure 2.11. a) Spherical image and observed sphere region; b) Viewport image with the related coordinate systems.

$(\phi,\theta)=(0,\frac{F_v}{2})$. Its projection point on the plane, $\bar{P}$, is consequently positioned at the vertical limit of the viewport, having Cartesian coordinates $(x_p,y_p)=(0,\frac{V_{vs}}{2})$, where V_{vs} is vertical viewport size in length units. V_{vs} can be computed as

$$\left(0,\frac{V_{vs}}{2}\right)=Proj\left(0,\frac{F_v}{2}\right). \tag{2.37}$$

Applying a similar rational to the horizontal field of view, results in

$$\left(\frac{V_{hs}}{2},0\right)=Proj\left(\frac{F_h}{2},0\right), \tag{2.38}$$

where V_{hs} is the viewport horizontal size in length units; F_h is the horizontal field of view. Since to obtain a viewport with a specific aspect ratio, $AR=\frac{V_{hs}}{V_{vs}}$, it is not possible to define F_h and F_v independently, firstly it is necessary to select a value for F_h (or F_v), then to compute V_{hs} (or V_{vs}) and finally, to compute V_{vs} (or V_{hs}), from the desired AR.

2) **Compute the plane coordinates, $(\boldsymbol{x_p},\boldsymbol{y_p})$** - To compute the plane coordinates, the image sampling coordinates, (u,v), need to be firstly computed from the pixel positions, (m,n). The (m,n) and (u,v) coordinates are related by

$$u=(m+0.5)\frac{V_{hs}}{W_{vp}}, \qquad 0\le m<W_{vp} \tag{2.39}$$

$$v=(n+0.5)\frac{V_{vs}}{H_{vp}}, \qquad 0\le n<H_{vp} \tag{2.40}$$

where W_{vp} and H_{vp} are, respectively, the viewport width and height, in pixels. The (x_p,y_p) and (u,v) coordinates are related by

$$x_p=u-\frac{V_{hs}}{2} \tag{2.41}$$

$$y_p=-v+\frac{V_{vs}}{2}. \tag{2.42}$$

At this point, the coordinates (x_p,y_p) can be obtained for every pixel on the viewport.

3) **Compute the spherical coordinates, $(\boldsymbol{\phi},\boldsymbol{\theta})$** - For each position (x_p,y_p), the spherical coordinates (ϕ,θ) are computed using the backward projection, $(\phi,\theta)=Proj^{-1}(x_p,y_p)$. To obtain a viewport oriented according to a generic viewing direction $(\phi_{VD},\theta_{VD},\psi_{VD})$, the

spherical coordinates (ϕ, θ) obtained for the front viewport are converted to Cartesian coordinates (X, Y, Z), as described in Section 2.4.1, and (2.43) should then be applied:

$$\left(\acute{X}, \acute{Y}, \acute{Z}\right)^T = R\ (\phi_{VD}, \theta_{VD}, \psi_{VD})\ (X, Y, Z)^T \tag{2.43}$$

where $(\acute{X}, \acute{Y}, \acute{Z})$ are the sphere positions in Cartesian coordinates correspondent to the viewport oriented according to $(\phi_{VD}, \theta_{VD}, \psi_{VD})$, and $R(\phi_{VD}, \theta_{VD}, \psi_{VD})$ is the rotation matrix considering 3DoF, given by:

$$R(\phi_{VD}, \theta_{VD}, \psi_{VD}) = \begin{bmatrix} [\cos(\phi_{VD})\cos(\psi_{VD})] & [-\sin(\phi_{VD})\sin(\theta_{VD})\cos(\psi_{VD}) - \cos(\theta_{VD})\sin(\psi_{VD})] & [\sin(\phi_{VD})\cos(\theta_{VD})\cos(\psi_{VD}) - \sin(\theta_{VD})\sin(\psi_{VD})] \\ [\cos(\phi_{VD})\sin(\psi_{VD})] & [-\sin(\theta_{VD})\sin(\phi_{VD})\sin(\psi_{VD}) + \cos(\theta_{VD})\cos(\psi_{VD})] & [\sin(\phi_{VD})\cos(\theta_{VD})\sin(\psi_{VD}) + \sin(\theta_{VD})\cos(\psi_{VD})] \\ [-\sin(\phi_{VD})] & [-\cos(\phi_{VD})\sin(\theta_{VD})] & [\cos(\phi_{VD})\cos(\theta_{VD})] \end{bmatrix}. \tag{2.44}$$

Then, $(\acute{X}, \acute{Y}, \acute{Z})$ is transformed back to spherical coordinates, as described in Section 2.4.1.

4) **Compute the viewport pixel values** - Finally, the information contained at the spherical coordinates obtained in Step 3), is transferred to the correspondent viewport pixel, at position (m, n). For the equirectangular representation of the sphere (equirectangular image (*ERI*)), the relationship between spherical coordinates, (ϕ, θ), and pixel coordinates, (m_{ERI}, n_{ERI}) of the ERI image is given by

$$m_{ERI} = \frac{\phi W_{ERI}}{2\pi} + 0.5(W_{ERI} - 1) \tag{2.45}$$

$$n_{ERI} = 0.5(H_{ERI} - 1) - \frac{\theta H_{ERI}}{\pi}, \tag{2.46}$$

where W_{ERI} and H_{ERI} are, respectively, the width and height of the equirectangular image, in pixels. Since (m_{ERI}, n_{ERI}) obtained from (2.45), (2.46) may have fractional values, the bilinear interpolation is used.

B. Quality Evaluation

Figure 2.12 depicts examples of the viewport obtained for each projection. For the Pannini projection, a projection center at $d = 1$ was considered, to preserve the object conformality as much as possible. As shown in Figure 2.12, all projections produce stretching and/or bending distortions in the viewport image; no projection can preserve all lines and object shapes simultaneously. Accordingly, the following analysis can be made based on the resulting geometric distortions:

- **Horizontal line bending** - Excluding transverse Mercator, rectilinear, and Pannini $(d = 1, vc = 1)$, all projections bend the horizontal lines; among these, the GPP with $d = 0.25$ has the least horizontal line bending, followed by the GPP with $d = 0.5$, stereographic, and basic Pannini $(d = 1)$. In the Central cylindrical, Equirectangular, Lambert equal-area, Mercator, and orthographic projections, the horizontal lines are too much bent.

- **Vertical line bending** - Excluding transverse Mercator, stereographic, orthographic, GPP $(d = 0.25$ and $d = 0.5)$, all projections keep the straightness of vertical lines. Among the projections that bend the vertical lines, the GPP with $d = 0.25$ has the least bending, followed by the GPP with $d = 0.5$, and stereographic. The transverse Mercator and orthographic projections result in strong vertical line curvature.

- **Small objects deformation** - The small objects, such as plates and bowls on the table of Figure 2.12, close to the top and bottom of the image, are stretched along the vertical direction in the central cylindrical projection, and along the horizontal direction in the equirectangular and Lambert equal-area projection. This does not happen for Mercator, transverse Mercator, stereographic, and basic Pannini ($d = 1$), since they are all conformal projections (object shapes are locally preserved). In rectilinear projection ($d = 0$) the objects close to the viewport borders are very stretched. In the GPP with $d = 0.25$ and $d = 0.5$, the object shapes are less stretched compared to the rectilinear case and, as d increases from 0.25 to 0.5, the object conformality increases (but lines are more bent). In the orthographic projection, the objects are too much deformed, particularly at the viewport borders. In Pannini ($d = 1, vc = 1$), the objects on the right and left side of the image are stretched along the vertical direction.

- **Large objects deformation** - Large objects, such as the white table in Figure 2.12, are deformed too much for central cylindrical, equirectangular, lambert equal-area, Mercator, rectilinear, and orthographic, but much less for other projections; however, in transverse Mercator the people on the left and right side of the image are deformed, but less than orthographic. In general, the objects in GPP ($d = 0.25$), GPP ($d = 0.5$), basic Pannini ($d = 1$), and Pannini ($d = 1, vc = 1$) appear less deformed compared to other projections, resulting in more pleasant viewports.

In summary, all projections produce stretching and/or bending distortions in the viewport. However, GPP and PP have advantages over the other projections by producing viewports with less visible geometric distortions. Furthermore, these projections allow to control the perceived geometric distortion - in type (bending or stretching) and strength - by varying the projection parameters.

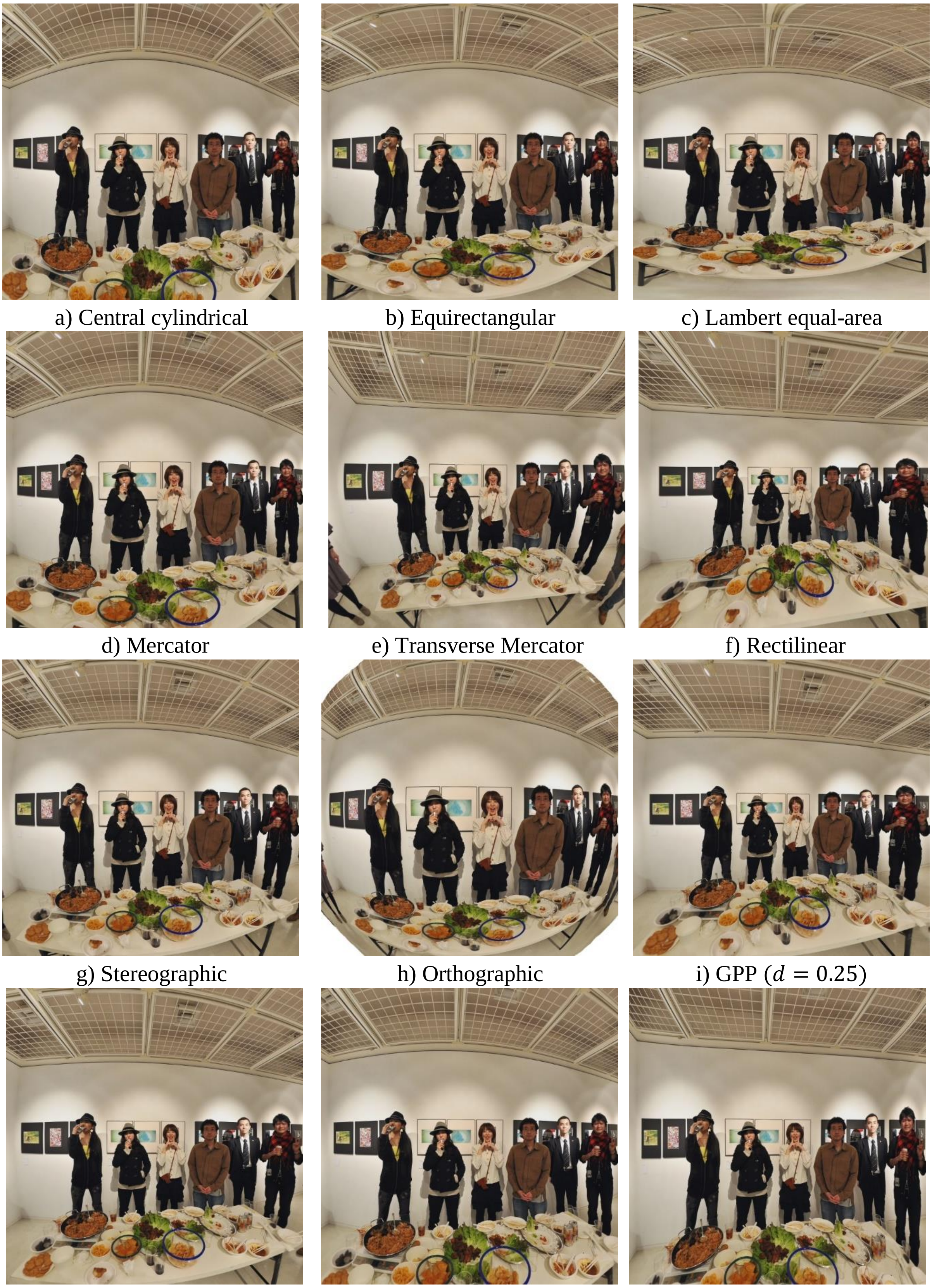
2.5 Content-aware Projections

As described earlier, content-aware projections are adapted to the content of the omnidirectional image, targeting a perceptually attractive viewport. Table 2.2 shows the classification of content-aware projections available in the literature, according to the developable surface; moreover, the targeted application of each projection is also presented. According to the preserved characteristics, these projections are classified as balanced. Note that two projections were proposed by Kim et al. in [10]: optimized Pannini projection (Kim et al. [10] in Table 2.2) and multiple optimized Pannini projection (Kim* et al. [10] in Table 2.2).

However, since there are a few specific classes that are exclusive of content-aware projections, two additional dimensions for classifying these projections were defined, namely: *i)* how the projection is adapted to the visual content and, *ii)* the number of projection planes used to render the viewport image. The classes organized by each dimension are described next:

i) Classification by content adaptation - The projection can be adapted to the image content locally, regionally, or globally:

  - **Locally adapted** - The projection parameters can change at the pixel level, i.e., from one pixel position to another pixel position of the viewport image, aiming to minimize the geometric distortions with a very flexible and localized procedure. Examples of locally adapted projections are briefly reviewed in Sections 2.5.1-2.5.3.



a) Central cylindrical b) Equirectangular c) Lambert equal-area

d) Mercator e) Transverse Mercator f) Rectilinear

g) Stereographic h) Orthographic i) GPP ($d = 0.25$)

j) GPP ($d = 0.5$) k) Basic Pannini ($d = 1$) l) Pannini ($d = 1, vc = 1$)

Figure 2.12. Examples of viewport obtained for several content-unaware projections.

Table 2.2. Content-aware projections with the corresponding classification and targeted application.

Projection	Classification by projection surface	Targeted application
Carroll et al. [36]	Planar	Wide-angle photography
Kopf et al. [54]	Hybrid	Wide-angle and panoramic photography
Shih et al. [55]	Planar	Wide-angle photography
Zelni-Manor et al. [43]	Planar	Wide-angle photography
Kim et al. [10]	Hybrid	Omnidirectional visual content rendering
Kim* et al. [10]	Hybrid	Omnidirectional visual content rendering
Chang et al. [37]	Hybrid	Wide-angle photography
Chang et al. [56]	Hybrid	Wide-angle and panoramic photography

- **Regionally adapted** - The projection parameters can change at the region level, i.e., from one region to another region of the viewport image. The aim is to minimize the geometric distortions in some image regions (usually corresponding to salient regions) and thus different projection parameters are used for the different regions that are being projected. Examples of regionally adapted projections are briefly reviewed in Sections 2.5.4 and 2.5.6.

- **Globally adapted** - The projection is adapted to the entire viewport image, meaning that the projection parameters do not change. This projection uses global distortion measures and aims to minimize the geometric distortions for all parts of the viewport with a single set of parameters. Examples of globally adapted projections are briefly reviewed in Sections 2.5.5, and 2.5.7-2.5.8.

ii) Classification by the number of projection planes - The sphere content can be rendered using just one projection plane, or several intermediate projection planes, as described next:

- **Single plane** - A single projection plane is defined for projecting the sphere content. The projection can be locally, regionally, or globally adapted to the image content. Examples of content-aware, single plane, projections are briefly reviewed in Sections 2.5.1, 2.5.3, 2.5.5, 2.5.7-2.5.8.

- **Multiple planes** - Several intermediate projections planes are defined, being a part of the sphere projected in each one. The projection can be locally, regionally, or globally adapted to the image content. Those planes are then combined to obtain the final projection. Examples of multiple planes projections are briefly reviewed in Sections 2.5.2, 2.5.4, and 2.5.6.

Usually, content-aware projections try to minimize the perceived geometric distortions in the image regions that attract more the human attention. Geometric distortions of lines and of salient regions (corresponding to important objects, such as human faces) may have a high perceptual impact, and thus deserve special attention in the projection procedure. Typically, content-aware projections minimize the geometric distortions for the entire image, or for some image regions, through an optimization technique, where the cost function to be minimized includes some terms which act as constraints. In the following, some of the most often used constraints are described:

- **Straight line constraint** - The straight lines in the visual scene should remain straight after projection. This constraint is defined over every, or just for some, straight lines in the image. Accordingly, the straight lines need to be identified in the image, either manually by the user, or automatically, by applying line detection techniques.
- **Conformality constraint** - The shapes of regions/objects should be preserved or presented in a perpetually attractive way. The regions/objects can be identified automatically using some saliency/object detection technique, or manually by the user.
- **Smoothness constraint** - To satisfy the two previous constraints, the projection parameters may do not change smoothly, leading to abrupt changes in some regions and structures in the final image. The smoothness constraint is defined to avoid abrupt changes in the geometric properties (such as scale or orientation) of image regions.

Typically, a combination of constraints is used to construct the cost function. Table 2.3 presents the content-aware projections classification - for the projections listed in Table 2.2 - according to content adaptation, the number of projection planes, the constraints that are used by each projection and the need of user interaction.

Table 2.3. Content-aware projection classification based on content adaptation and the number of projection planes, as well as used constraints.

Projection	Classification by content adaptation	Classification by number of planes	Conformality constraint	Line constraint	Smoothness constraint	User interaction
Carroll et al.[36]	Locally adapted	Single plane	✓	✓	✓	✓
Kopf et al. [54]	Locally adapted	Multiple planes	-	-	✓	✓
Shih et al. [55]	Locally adapted	Single plane	✓	✓	✓	-
Zelnik-Manor et al. [43]	Regionally adapted	Multiple planes	-	-	-	✓
Kim et al. [10]	Globally adapted	Single plane	✓	✓	-	-
Kim* et al. [10]	Regionally adapted	Multiple planes	✓	✓	-	-
Chang et al. [37]	Globally adapted	Single plane	-	✓	-	-
Chang et al. [56]	Globally adapted	Single plane	✓	✓	-	-

As shown in Table 2.2, most of the content-aware projections were developed to reduce the geometric distortions in wide-angle or panoramic images. However, these projections and the involved procedures may play also an important role in the context of viewport rendering of omnidirectional images. The following sections provide a summarized description of the projections listed in Table 2.3; for qualitative comparison purposes, the last section contains examples of rendered viewports for a subset of those projections.

2.5.1 Locally Adapted Projection for Wide-angle Images

In [36], a spatially varying projection for mapping wide-angle images (up to HFoV of 180°) was proposed. The projection is locally adapted based on a set of conformality and line constrains measures, computed from the image, seeking to minimize the geometric distortions. First, the user needs to specify the type of input wide-angle image and its FoV, then the input image is projected on a spherical surface. The sphere content is then sampled in longitude and latitude with a step size $(\Delta\phi, \Delta\theta)$, with $\Delta\phi = \Delta\theta$, resulting in a mesh of N points, where each point is indexed by ij and parameterized by the spherical coordinates (ϕ_{ij}, θ_{ij}). The sphere is mapped on the image plane, resulting in a mesh of points parameterized by Cartesian coordinates (x_{ij}, y_{ij}), as shown in Figure 2.13.

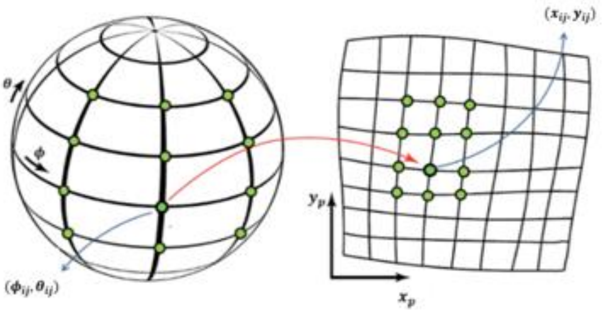


Figure 2.13. Representation of sampled points on the sphere and of their projection on the image plane (based on [36]).

In the following, the main steps of this projection are briefly described:

- **Selection of straight lines by user** - For the input image, and through a user interface, the user selects manually (i.e., drawing over the input image) the lines in the visual scene that should be kept straight in the final image, after projection; additionally, the user assigns one of the following line constraints to each selected line: fixed direction (horizontal or vertical); general direction. The orientation of lines with fixed direction should be kept after projection, while the orientation of lines with general direction is allowed to change. The selected line set is denoted by L, where each line $l \in L$ consists of a set of points. The subsets of lines with fixed direction and with general direction are denoted, respectively, as L_f and L_g.

- **Projection calculation** - For each (ϕ_{ij}, θ_{ij}), the corresponding (x_{ij}, y_{ij}) is computed through a least-square optimization technique, where a cost function, referred to as total energy, E_t, is minimized, aiming at the lowest possible distortion. The energy E_t is composed by three sub-energies: *i)* conformality energy, E_c, which aims to measure the shape distortion; *ii)* line straightness energies, E_{lo} and E_{ld}, which measure the distortion of straight lines; and *iii)* smoothness energy, E_s, which aims to limit the scale and orientation changes in the E_t optimization. The E_s term is described in [36], whereas the E_c, E_{lo}, E_{ld} terms are detailed in Section 2.6.2. The energy E_t results from a weighted sum of the conformality, smoothness, and line energies:

$$E_t = w_c^2 \, E_c + w_s^2 E_s + w_l^2 \left(\sum\nolimits_{l \in L_f} E_{lo} + \sum\nolimits_{l \in L_g} E_{lo} + \sum\nolimits_{l \in L_g} E_{ld} \right), \tag{2.47}$$

where w_c, w_s, and w_l are the weighting parameters for the corresponding energy terms. In (2.47), E_{lo} aims the line bending minimization while keeping the line direction specified by the user (if any); on the other hand, E_{ld} aims the minimization of the bending, regardless of the line

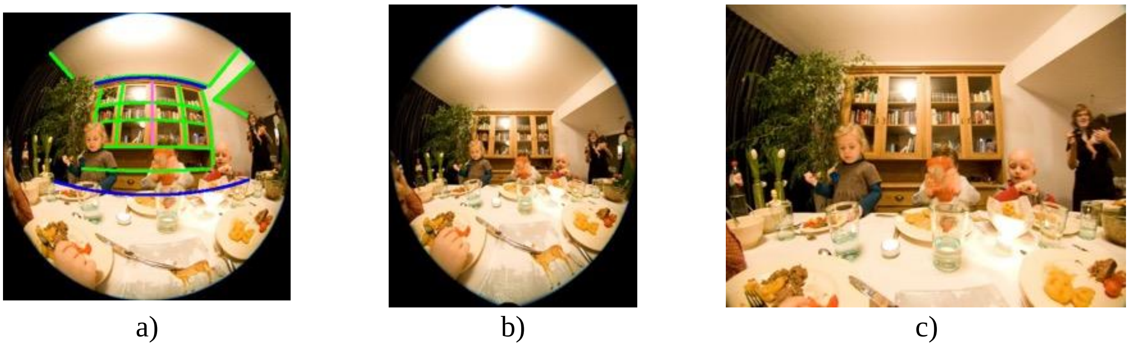


Figure 2.14. a) Fish-eye input image with lines selected by the user; output image b) before cropping and c) after cropping (based on [36]).

direction. The optimized mesh is produced by iteratively minimizing E_t, using a least-square optimization technique, seeking the lowest possible distortion. The final image is rendered by warping the input image according to the optimized mesh, using bilinear interpolation.

Figure 2.14 depicts an input image, where the straight lines selected by the user are highlighted in green (lines with general direction), pink (lines with vertical direction), and blue (lines with horizontal direction). This image was captured using a fish-eye lens, having vertical and horizontal FoVs of 180°. The images resulting from the described method, before and after manual cropping, are presented in Figure 2.14b) and Figure 2.14c), respectively. As shown in Figure 2.14c), the output image is very realistic and does not present visible distortion since the shape of the objects is preserved, and most of the straight lines are projected without bending.

The projection has the advantage of producing very impressive results for images with large FoV; however, it has some disadvantages: *i)* user interaction is required; *ii)* when a large number of lines with different directions are specified by the user, the likelihood of projecting all of them without bending is reduced; *iii)* lacks an automatic stopping condition and thus, the user must evaluate the result visually and decide if the procedure should, or not, iterate again; *iv)* the user must crop the output manually to obtain a rectangular image, which may lead to information loss.

2.5.2 Locally Adapted Projection for Panoramic Images

In [54], a locally adapted projection was proposed for mapping wide-angle or panoramic images, aiming to reduce the geometrical distortions, particularly in the perceptually important parts of the image. The final image is obtained by using the rectilinear projection for user specified regions and performing a seamless transition to a cylindrical projection over the rest of the image. In the following, the main steps of this projection are briefly described:

- **Selection of image regions** - The input image (wide-angle or panoramic) is projected to a cylindrical surface, using a central projection, which is then unwrapped to a plane and shown to the user. Then, the user specifies the regions that need to be preserved. Figure 2.15a) depicts a cylindrical image (after unwrapping to a plane) with user specified regions. To have the desired orientations and region sizes on the final image, the user is able to: *i)* change the orientation of each region by rotating it around the centroid; *ii)* change the size of each region. At this step, the positions, on the cylinder surface, of user specified regions, and their desired orientations and sizes, are known.

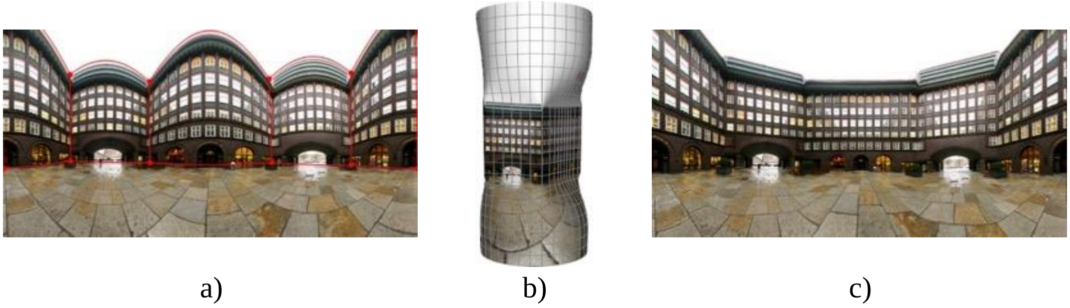


Figure 2.15. a) Input wide-angle image and user specified regions; b) use the planar surface for user specified regions and cylinder surface for other regions; c) final output image (based on [54]).

- **Projection surfaces generation** - The regions previously defined on the cylindrical surface are projected onto a plane, using rectilinear projection, *cf.* Figure 2.15b). This process introduces orientation discontinuities (dramatic changes in image regions and structures), at the regions close to the user specified region boundaries. This problem was addressed with a least-square optimization procedure. The cost function is defined to satisfy the user specified constraints, described in the first step, and to smooth the transition between user specified regions as well as between the planar and cylindrical surfaces. This step outputs the image presented in Figure 2.15b).

- **Final image generation** - The projection surface obtained in the previous step is on the cylinder surface and needs to be unwrapped to a plane, to obtain the final projection; however, this surface cannot be unwrapped to a plane without introducing some distortions (it is not a pure cylindrical surface). In this case, the surface is unfolded to a plane using a surface parameterization technique called as Most Isometric Parametrization (MIPS), proposed in [57], that allows to transform the obtained surface into a plane without deforming it. The final output image, depicted in Figure 2.15c), is rendered by warping the input image according to the surface generated by MIPS, using bilinear interpolation.

An advantage of this method is that the lines of the visual scene are kept straight. However, this method has several disadvantages: *i)* it requires user interaction; *ii)* if the user specified regions have a large area, it may lead to objects stretching in the final result (as in the rectilinear projection); *iii)* straight lines may appear bent in the user specified regions; *iv)* orientation of regions may change slightly relatively to the orientation specified by the user, and thus may lead to the deformation of some vertical lines.

2.5.3 Locally Adapted Projection for Wide-Angle Selfy Images

In [55], a locally adapted projection was proposed to undistort human faces in photographic wide-angle images (with HFoV in the range of 70°-120°), acquired from camera phones. The input image is obtained with a perspective projection, thus straight lines keep their straightness, but the objects (e.g., faces) may be stretched. The final image is obtained by locally adapting a stereographic projection located on faces and performing a seamless transition to the background regions. In the following, the main steps of this projection are briefly described:

- **Person segmentation** - The person segmentation method proposed in [58] was applied to the input image to identify the persons in the image (*cf.* Figure 2.16). Since the main purpose was to correct faces and hair, a face mask was generated by intersecting the person mask with a rectangular face bounding box obtained with a face detector. Then, the face bounds were

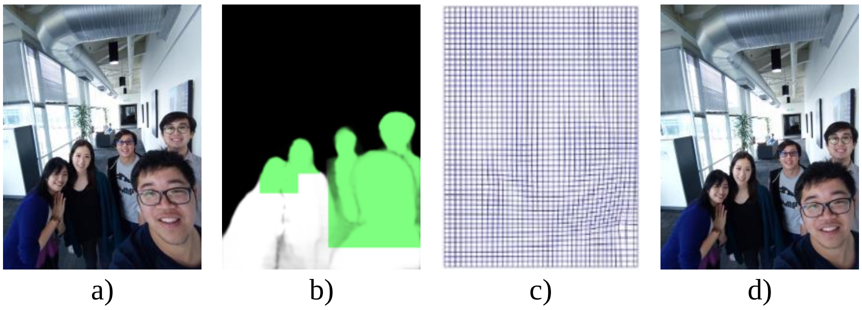


Figure 2.16. a) Input image with a horizontal FoV of 97°; b) subject mask segmentation with identified facial regions; c) optimized mesh; d) final output (based on [55]).

extended to cover the hair and other parts of the face regions. As an example, the face mask corresponding to Figure 2.16a) is shown in green colour in Figure 2.16b).

- **Stereographic mesh** - A uniform grid mesh $M_p = \{\boldsymbol{p_i}\}$ was defined for the input image, consisting on a vertex set $\{\boldsymbol{p_i}\}$, where $\boldsymbol{p_i}$ refers to a 2D mesh coordinate. Then, the stereographic mesh, denoted by $M_u = \{\boldsymbol{u_i}\}$, is created by applying a procedure proposed in [59], to every $\boldsymbol{p_i}$ inside a facial region (identified in the previous step). Since the stereographic projection is a conformal projection, in M_u the faces shapes are thus preserved, while straight lines remain straight in the background. However, this process creates visual distortions at the face boundaries.

- **Mesh optimization** - A procedure similar to the optimization described in Section 2.5.1 was used to obtain a smooth transition between faces and the background. An optimized mesh, denoted as M^* and aiming at the lowest possible distortion, is obtained by iteratively minimizing a cost function using a least-square optimization procedure; the cost function is a weighted sum of energy terms that account for the faces conformality and for the straightness of the background lines. The output mesh corresponding to Figure 2.16a) is depicted in Figure 2.16c).

- **Warping -** The final output image is obtained by warping the input image according to the optimized mesh. In this process, the optimized mesh is used to map pixels from the input image to the new positions in the output image. The interpolation procedure proposed in [60] was used for warping. The final output image, corresponding to the input image of Figure 2.16a), is depicted in Figure 2.16d).

This projection has the advantage of being fully automatic; however, it has some disadvantages: *i)* the geometric distortion correction is only performed on facial regions; *ii)* correcting only the face, without the rest of the body (e.g., human shoulders), may create an unnatural look; *iii)* straight lines and objects may be somewhat distorted if they are close to the face regions.

2.5.4 Multiple Perspective Projections for Wide-Angle Images

In [43], two multiple perspective projections, being one object-based, were proposed to map wide-angle images (up to HFoV of 180°) onto a plane, aiming to reduce geometric distortions caused by using a single perspective projection. Both projections are regionally adapted, and are briefly described in the following:

i) Multiple rectilinear projections - Instead of projecting the sphere content onto a single plane, multiple tangent planes are used, each one covering a limited field of view.

Figure 2.17 shows an example of multiple rectilinear projections with three tangent planes, located at different points of the sphere. The main steps of this projection are:

- **Selection of plane position by user** - A simple user interface was designed to allow the user to select the position of the tangent planes. The input image is presented to the user in an equirectangular format. The user must carefully choose the center of the tangent planes and the region covered by each, to reduce the geometric distortions.

- **Rectilinear projection** - For each tangent plane, the rectilinear projection is used to map the corresponding regions of the sphere to that plane. Afterwards, the tangent planes are arranged on a single flat surface to obtain the final projection.

Using this technique, orientation discontinuities may occur between the tangent planes, leading to the creation of visual artifacts. For example, the objects and straight lines located between two tangent planes may appear distorted.

ii) Multiple object-based rectilinear projections - This solution was proposed to reduce the orientation discontinuities in some image regions and structures introduced by the multiple rectilinear projections, previously described. The main steps of this projection are:

- Manually, obtain a foreground-background segmentation mask for the input image, in equirectangular format, and cut out the foreground objects. The GIMP [61] implementation of Intelligent Scissors [62], which requires user interaction, was used for this segmentation.

- Fill the holes in the background caused by cutting out the foreground objects using a texture propagation technique [63]. Then, project all content with a filled background to a plane using multiple rectilinear projections, as previously described.

- Project each object using a rectilinear projection centred on the object, and then paste objects onto the background to obtain the final image; the objects are placed according to their positions on the input image.

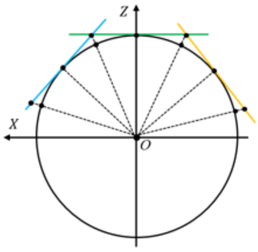
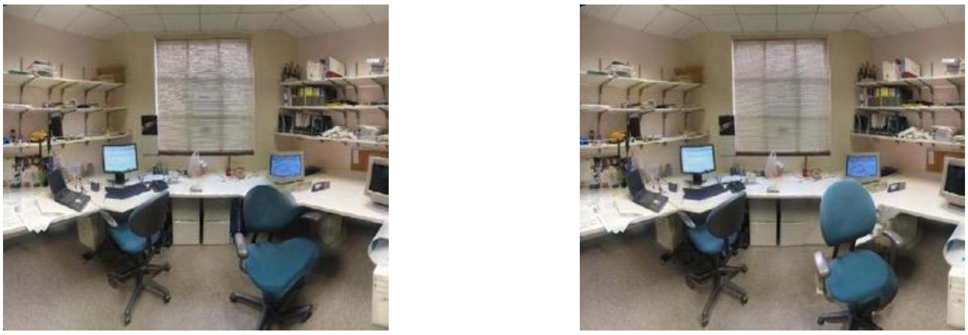


Figure 2.17. Top view of multiple rectilinear projections.

Figure 2.18 shows an example of images rendered with the two multiple rectilinear projection approaches. As shown, the straight lines appear straight for both projections. However, in the first approach (in Figure 2.18a) some objects on the left side (e.g., monitor, keyboard, and a part of the table), and the chair and monitor on the right side, are clearly distorted due to the orientation discontinuities. The multiple object-based rectilinear projections (in Figure 2.18b) can correct the distortion of the chair but not on the other objects.

The main advantages of both projections are the preservation of straight lines, and the fact that they use the rectilinear projection only for a limited field of view, which reduces the geometric (mostly stretching) distortions compared to a single rectilinear projection, covering the whole



a) Multiple rectilinear projections b) Multiple object-based rectilinear projections

Figure 2.18. Examples for the multiple rectilinear projections and multiple object-based rectilinear projections (based on [43]).

image. However, they present also some disadvantages: *i)* user interaction is required; *ii)* orientation discontinuities are introduced; *iii)* rectilinear projection can distort the objects if they are close to the projection plane borders, even for limited FoVs.

2.5.5 Optimized Pannini Projection for Omnidirectional Visual Content Rendering

In [10], an automatic procedure to optimize the Pannini projection [44] was proposed and applied for viewport rendering of omnidirectional visual content. The Pannini projection is globally adapted, where the parameters (d, vc) are found based on an optimization procedure that relies on automatically detected lines and salient points, i.e., points centered on perceptually relevant regions (e.g., a human face). The optimization procedure attempts to minimize a cost function, which combines line straightness and conformality measures. In the following, the main steps of this projection are briefly described:

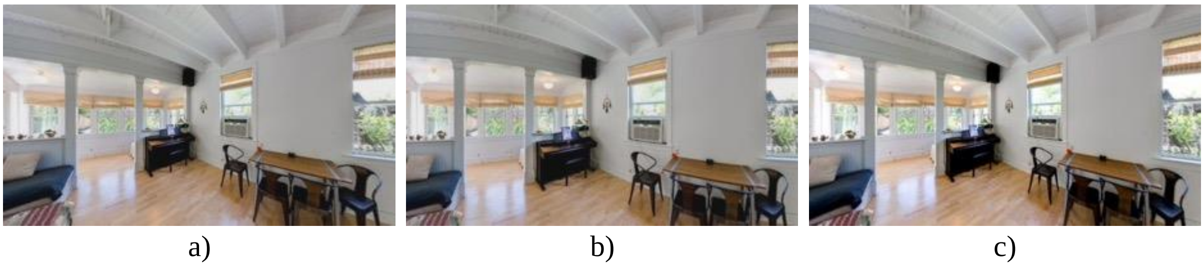
- **Compute line straightness measure** - The selected part of the sphere corresponding to the viewport is projected onto a plane using rectilinear projection; then, straight lines are automatically detected in the viewport using the line detector proposed in [64]. After, the line straightness measure, E_{ld}, originally proposed in [36] and already referred to in Section 2.5.1, was computed for each detected line.

- **Compute conformality measure** - The salient regions in the viewport are automatically detected using the saliency detection model proposed in [65]. For each salient region, a salient point is defined, centred at the salient regions. Then, the conformality measure, E_c, originally proposed in [36] and already referred to in Section 2.5.1 was computed for each salient point, allowing to preserve the shape of the region around that point.

- **Pannini parameters optimization** - The Pannini projection parameters (d, vc) that minimize the geometric distortions in the viewport were obtained by minimizing the cost function E, given by

$$E = w_l \sum\nolimits_{l \in L} E_{ld} + w_c \sum\nolimits_{p \in SP} E_c \, , \tag{2.48}$$

where L and SP are, respectively, the detected lines set, and the salient points set; w_l and w_c are the weighting parameters of corresponding terms. The final projection is obtained by iteratively minimizing E using a gradient descent technique, aiming for the lowest possible distortion.



a) b) c)

Figure 2.19. a) Pannini with $d = 0.5, vc = 0$; b) Pannini with $d = 1, vc = 0$; c) Optimized Pannini (based on [10]).

Figure 2.19 presents viewports rendered with a horizontal FoV of 150° using the Pannini projection with fixed parameters $(d = 0.5, vc = 0)$ and $(d = 1, vc = 0)$, as well as the viewport obtained with the optimized Pannini projection. As can be figured out, the optimized Pannini projection produced a viewport with a much lower amount of line bending and stretching distortions than the Pannini with fixed parameters.

The main advantage of the optimized Pannini projection is not requiring any user interaction; however, it lacks a local adaption to the content, since it minimizes only global distortions and thus stretching and/or bending may be still visible for some image regions and structures. Moreover, the crowdsourcing subjective test conducted in [10] showed that, on average, the Pannini projection with fixed parameters achieved higher quality scores than the optimized Pannini.

2.5.6 Multiple Optimized Pannini Projections for Omnidirectional Visual Content Rendering

In [10], a fusion of multiple Pannini projections was proposed for rendering omnidirectional visual content; it is regionally adapted and builds on the work of the optimized Pannini projection described in the previous section. However, in this case, the final rendered image results from the fusion of multiple optimized Pannini projections, each centered in a salient point. In the following, the main steps of this projection are briefly described:

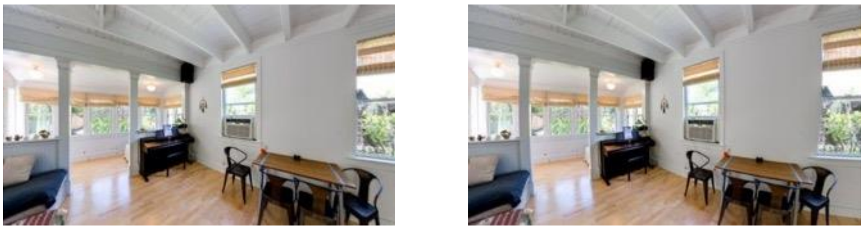
- **Salient points computation** - First, the fraction of the sphere corresponding to the viewport is forward projected onto a plane, using a rectilinear projection. Then, a saliency detection technique is applied to the projected image, to obtain the saliency map. For each salient region, a saliency point is defined and positioned in the region centroid. Finally, the corresponding spherical positions of the salient points are computed, using the rectilinear backward projection.

- **Global and local projection** - Instead of using a single projection, a global optimized Pannini projection (Gproj), and a set of regionally optimized Pannini projections (RProj), are used. The parameters of the Gproj and RProj projections are obtained with the technique described in the previous section. The Gproj includes the whole viewport and is centred on the viewport center, thus aligned with the viewing direction; it aims to project the regions close to the viewport center with low distortion. Each RProj is centred at a salient point, with the aim of representing the regions around that point with lower distortion (compared to Gproj), in the viewport image.

- **Projection alignment and scaling** - Before obtaining the final image, for each RProj the salient local region is aligned with the corresponding regions of the Gproj, allowing to combine Gproj and RProj. Scaling is used to adjust the size of the local region to match the size of the corresponding region in Gproj. This process guarantees that the region covered by RProj has a similar size to the corresponding region on the Gproj and thus, the object (contained inside these regions) boundaries are somewhat matched and a better fusion of Gproj and RProjs can be obtained.

- **Projection fusion** - To obtain the final viewport, the Gproj and RProj projections are fused, by combining the spherical coordinates of Gproj and RProj. To obtain the final viewport with less geometric distortions, the fusion varies the influence of each projection along the viewport. In this case, image regions near a salient point are heavily influenced by its corresponding RProj, considering that the salient local region is projected with less distortion when its regional projection is used, and image regions near the viewport center are heavily influenced by the global projection, which has minimum geometric distortion in this area.

Figure 2.20 presents two viewports, rendered from the same omnidirectional image and viewing direction, with a horizontal FoV of 150°, and using the optimized Pannini projection, described in the previous section, and the multiple optimized Pannini projection, described in this section. As can be observed, the multiple optimized Pannini projection produces a viewport with less geometric distortion than the viewport obtained with the optimized Pannini; in particular, the chairs and the table in the bottom-right of Figure 2.20 appear less distorted for the multiple optimized Pannini compared to the optimized Pannini.



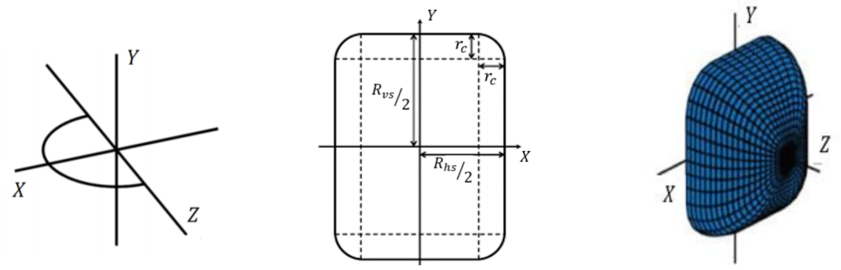
a) b)

Figure 2.20. Example of viewports obtained using the projections proposed in [10]: a) Optimized Pannini projection; b) Multiple optimized Pannini projection (based on [10]).

The main advantage of the multiple Pannini projection is to not require user interaction; however, it has some disadvantages: *i)* for each considered salient point, the optimized Pannini projection has to be regionally optimized (a large number of parameters need to be found for all salient points), which increases complexity; *ii)* when there are multiple salient points close to the same linear structure, this structure may be strongly distorted (because different projections are used over it); *iii)* any misalignment or incorrect scaling may result in some visual artifacts in the final viewport image; *iv)* geometric distortions on the fused viewport are not measured and thus the local projections are not jointly optimized, which may result in viewports with still some geometric distortions (bending or stretching).

2.5.7 Rectangling Stereographic Projection for Wide-angle Images

In [37], the rectangling stereographic projection was proposed for mapping wide-angle images (up to HFoV of 180◦) onto a plane, while minimizing the geometric distortions. It uses a swung surface, which is a generalization of a surface of revolution in which the rotation around an axis, of an initial profile curve, is guided and scaled by a trajectory curve. Figure 2.21a)-c) illustrate the profile curve, a rounded rectangle defining the trajectory curve and the resulting swung surface, respectively. The projection is globally adapted and is obtained by projecting



a) b) c)

Figure 2.21. a) Profile curve; b) trajectory curve; c) resulting swung surface (based on [37]).

the sphere content on the swung surface using rectilinear projection, followed by the projection of the swung surface content on the final projection plane, using a stereographic projection. The construction of the swung surface is the key of this projection design and is responsible for the reduction of the geometrical distortions. The trajectory curve corresponds to a rounded rectangle, and is defined with the following geometrical parameters: roundness of corners, r_c, and aspect ratio, $R_{ar} = R_{vs}/R_{hs}$, where R_{vs} and R_{hs} are, respectively, the rounded rectangle height and width (*cf.* Figure 2.21b). These parameters define the swung surface and, therefore, how the content is projected. In the rectangling stereographic projection, these parameters were optimized based on the image content, namely some detected lines of the visual scene.

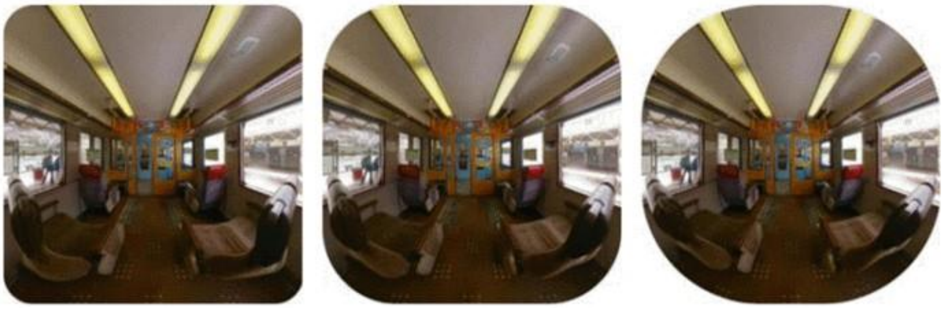
In the following, the main steps of this projection are briefly described:

- **Line detection** - The sphere content is first projected on a cube, using the cube map projection [19]. The line detector proposed in [64] is applied to each cube face, resulting in a set, L, of detected straight lines. Then, lines are projected back onto the sphere to obtain their positions in the spherical domain.

- **Swung surface optimization** - Due to the swung surface characteristics, the vertical lines close to the top and bottom of the image, and the horizontal lines close to the left and right of the image, may be projected with some bending. Therefore, the swung surface (i.e., parameters R_{ar} and r_c) is optimized by minimizing the number of lines of L that, after their projection on the plane, are positioned in regions that bent them.

- **Sphere to plane projection** - The content of the sphere is projected onto the already optimized swung surface, using the rectilinear projection. After, the content on the swung surface is projected onto the plane, which is tangent to the swung surface, using a stereographic projection, resulting in the final planar image. The stereographic projection is used since it preserves the local shapes.

Figure 2.22 depicts rendered images using the described projection, with a fix R_{ar} value and varying r_c. As can be observed, for a r_c value of 0.2, the geometric distortions are higher near the image borders. For example, the sofas on the left and right sides of Figure 2.22a) appear more deformed compared to an image obtained with a higher r_c value. However, as stated in [37], an image with a rectangular boundary is more attractive.



a) $r_c = 0.2$ b) $r_c = 0.5$ c) $r_c = 0.8$

Figure 2.22. Results obtained with the proposed method in [37], using fixed R_{ar} and varying r_c (based on [37]).

The main advantage of this projection is being fully automatic; also, since the content of the swung surface is projected onto the plane using the stereographic projection, object shapes are in general preserved. However, straight lines may appear bent for regions close to the image borders. Moreover, this projection only minimizes global distortions and thus lacks local adaption to the image content.

2.5.8 Swung to Cylinder Projection for Panoramic Images

In [56], the swung to cylinder projection was proposed for mapping wide-angle or panoramic images. The geometrical construction and the steps of this projection are similar to the projection described in the previous section. However, in this case the swung surface content is projected on a cylinder surface, tangent to the sphere at the equator line, instead onto a plane, and the final projection is obtained by unwrapping the cylinder surface. Using a cylinder instead of a plane allows projecting an image with a horizontal field of view covering the full 360°. Besides the swung surface parameters, r_c and R_{ar}, two additional parameters were introduced: the projection center, d, and the normal curvature of the projection cylinder, $c_k = 1/c_r$, where c_r is the cylinder radius. The parameter d allows the projection center to change, and c_k allows the final projection surface to change from a planar to a cylindrical surface. If $d = 1$ and $c_k = 0$, the final projection surface is a plane and the projection is equivalent to the rectangling stereographic projection, described in the previous section. In swung to cylinder projection, the parameters d, c_k, r_c, R_{ar}, were optimized based on the image content, namely some detected lines and salient regions. In the following, the main steps of this projection are briefly described:

- **Line and saliency detection** - The same procedure of the previous technique is applied to obtain a set of lines defined on the sphere. In addition, to detect salient regions, the gradient magnitude (a simple measure of visual saliency) is computed for the input image and used as a saliency map.

- **Sphere to cylinder projection** - The content of the sphere is projected onto the cylinder in two steps: first, the content of the sphere is projected onto the swung surface using the rectilinear projection; then, the content on the swung surface is projected onto the cylinder, using projection lines emanating from the projection center d.

- **Projection parameters optimization** - The projection parameters are optimized in two steps: in the first step, the parameters (d, c_k) are found by minimizing a cost function, based on a set of conformality and line straightness measures, that account for distortion of salient regions and bending of straight lines, respectively. After determining the best (d, c_k), the

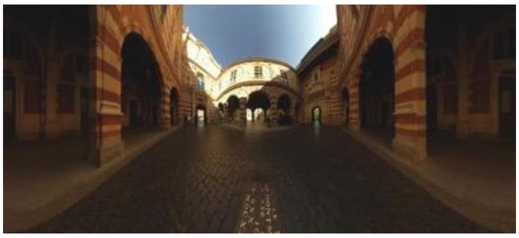


Figure 2.23. Example of panorama with a HFoV of 360°, obtained with the swung to cylinder projection with optimized parameters ($\boldsymbol{d = 0.6, c_k = 0.6, r_c = 0.75, R_{ar} = 3}$) (based on [56]).

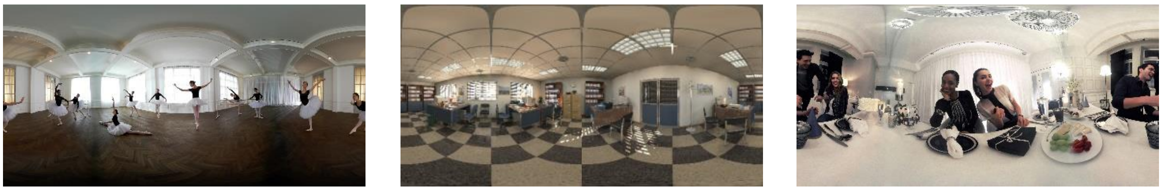
parameters (r_c, R_{ar}) are optimized using a procedure similar to what was described in the previous section.

Figure 2.23 illustrates the final image, with a FoV covering the full 360°, obtained with the swung to cylinder projection.

This projection has the advantage of being fully automatic; moreover, it can represent the visual scene with a wide-angle (up to HFoV of 360°) better than the projection described in the previous section. However, this projection has some disadvantages: *i)* it has a higher number of parameters compared to the projection described in the previous section (4 *vs* 2), which increases complexity; *ii)* as the projection of the previous section, it lacks local adaptation to the content and thus the horizontal and radial lines may be bent, reducing the user perceived quality.

2.5.9 Qualitative Evaluation of Content-Aware Projections

This section presents a qualitative comparison of some selected fully automatic (i.e., user interaction is not required), content-aware projections. These projections were previously described and classified in Table 2.3, and are the following ones: rectangling stereographic projection [37], optimized Pannini projection [10] and multiple optimized Pannini projection [10]. Not all projections were included in this study since the source code was not available for some, while others do not work for all types of images (e.g., the method of Section 2.5.3 requires faces to work properly). Five content-unaware projections are also considered, for comparison purposes, namely: rectilinear, stereographic, basic Pannini with $d = 0.5, vc = 0$, basic Pannini with $d = 1, vc = 0$ (or stereographic Pannini), and Pannini with $d = 1, vc = 0.5$. For each projection under evaluation, three viewports were obtained from three omnidirectional images (presented in Figure 2.24 in equirectangular format), available in the dataset of [10]. Each viewport has a HFoV of 150° and a spatial resolution of 960×540 pixels ($AR = 16/9$).



a) *Dance* (3840 × 1920) b) *Office1* (3200 × 1600) c) *Dinner 1* (4000 × 2000)

Figure 2.24. Omnidirectional images and their spatial resolution.

Figure 2.25 shows three viewport examples, obtained for all the projections under evaluation, allowing the following analysis based on the resulting geometric distortions:

- **Horizontal line bending -** Except for the rectilinear projection, in which all straight lines in the visual scene remain straight in the viewport, all projections bend the horizontal lines, although with different strengths. In Pannini with $d = 1, vc = 0.5$, the horizontal lines are only slightly bent (due to vertical compression) at the cost of bending the radial lines and/or stretching objects vertically at the left and right sides of the viewport. In the optimized Pannini, the horizontal lines are less bent than in the multiple optimized Pannini. The amount of horizontal lines bending present in the viewports obtained using the multiple optimized Pannini was not expected, since this projection is content-aware.

- **Vertical line bending -** All vertical lines in the visual scene remain straight in the viewport for most of the projections, being the stereographic, rectangling stereographic, and multiple optimized Pannini the exception. In rectangling stereographic, the vertical lines are less bent than in stereographic, particularly for the lines in the regions close to the viewport center. In the multiple optimized Pannini, the vertical lines remain straight in *Dance* and *Office 1* viewports, but not in the *Dinner 1* viewport (e.g., the vertical floor lamp on the left side of the *Dinner 1* viewport is bent).

- **Small objects deformation** - Excluding stereographic and Pannini ($d = 1, vc = 0$), all other projections deform small objects, notably the objects close to the viewport borders. Stereographic and Pannini ($d = 1, vc = 0$) are conformal projections and preserve objects locally but not globally; e.g., in the *Dinner 1* viewports of these projections, while the white table is deformed globally, the plates on the table are not deformed.

- **Large objects deformation** - The large objects (e.g., people, chairs, tables), are too much stretched towards the viewport borders, in the rectilinear projection; this projection has also a strong perspective effect (or tunnel effect), where the objects close to the viewport center appear further away from the camera, compared to those on the image borders. In stereographic and rectangling stereographic, the objects are globally too much deformed, particularly in *Office 1* viewport. In Pannini ($d = 1, vc = 0.5$), the objects on the left and right side of the viewports are stretched vertically. In the basic Pannini, the object stretching is reduced when d varies from 0.5 to 1, at the cost of more bending on horizontal lines. In optimized Pannini, the object shapes are preserved in the *Dance* viewport, but stretched in the *Office 1* and *Dinner 1* viewports, e.g., the chair on the left side of *Office 1* viewport and the boy’s arm on the left side of the *Dinner 1* viewport. In the multiple optimized Pannini, the object shapes are better preserved than in the optimized Pannini for *Dance* and *Office 1* but deformed in the *Dinner 1* viewport (e.g., the white table).

From the results of Figure 2.25, it is possible to conclude that no content-aware projection produces viewports without visible stretching and/or bending distortions. However, viewports produced by content-aware projections have less geometric distortion than those resulting from content-unaware projections. Also, among content-aware projections, optimized Pannini and multiple optimized Pannini produced viewports with less geometric distortion and thus the viewport quality is more pleasant.

2.6 Quality Assessment of Sphere to Plane Projections

This section overviews the state of the art on subjective and objective quality assessment of omnidirectional images, with the focus on the geometric distortions introduced by the rendering process.

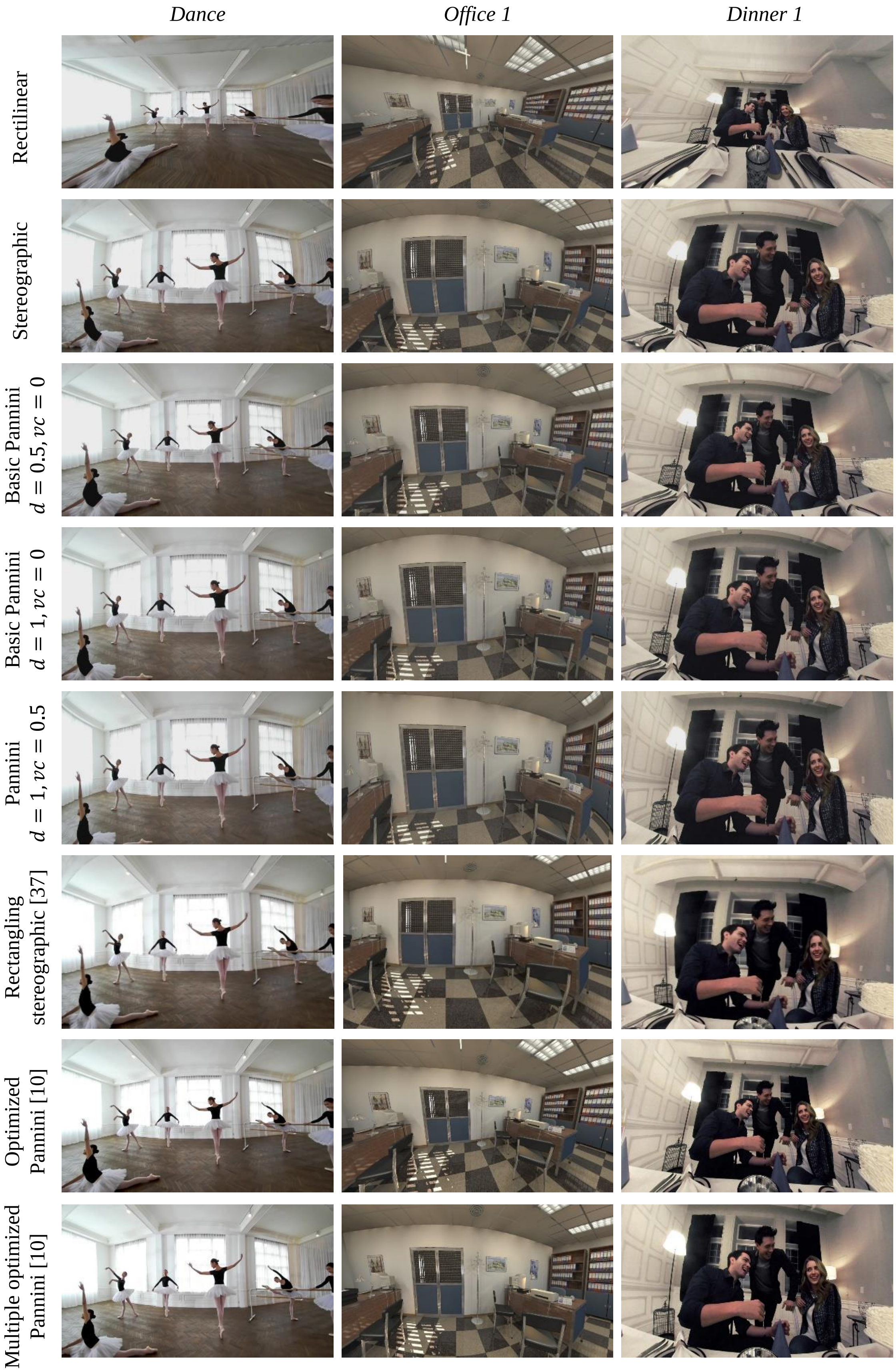


Figure 2.25. Examples of viewports obtained for the selected content-aware and content-unaware projections using a horizontal FoV of 150°. The viewports for optimized Pannini and multiple optimized Pannini were obtained from [10].

2.6.1 Subjective and Objective Quality Assessment

Omnidirectional image/video quality can be assessed using subjective or objective methods. The subjective quality assessment targets the perceptual quality evaluation by humans, while the objective quality assessment targets perceptual quality evaluation based on computational models, that can predict the perceived quality.

Since humans are the target consumers of the omnidirectional visual content they are also, and naturally, the most reliable source to obtain a quality score for an omnidirectional image/video. Therefore, the subjective evaluation tests are conducted with several viewers, that are asked to express their opinion about a set of stimuli - images or videos - and the resulting quality scores are regarded as the ground truth data about the image/video quality. However, this type of quality assessment is costly, requires the viewers availability to do the test, and it is impossible to be conducted in real-time. Therefore, subjective tests are mainly performed to obtain ground truth data to be used in the development, and evaluation, of objective quality metrics.

The design of a subjective quality assessment test requires some basic components that must be selected depending on the application and aim of the test:

- **Test environment** - Typically, the subjective tests are performed in a dedicated laboratory room or online through a web-based platform [66].
- **Visualization device** - For omnidirectional images, different types of displays can be used, such as HMDs, smartphones, or personal computers with a 2D display [67]. In the case of smartphones, an additional mobile VR headset is often used, such as the Google Cardboard (Mobile VR). In this case, the smartphone is slipped into the headset, and then the user wears it to explore the content [68].
- **Test materials** - The test material is the content selected for the subjective experiment, such as omnidirectional images/videos and rendered viewports.
- **Test methodology** - The test methodology specifies how the subjective experiment should be conducted, notably how the test materials are shown to the observers and how the opinion scores should be collected. Several test methodologies are available, which are defined in international standards (mostly from ITU) [69]–[74].
- **Test Subjects** - According to the Recommendation ITU-R BT 500.13 [70], a minimum of 15 subjects should be used for a subjective assessment test to be statistically meaningful. However, less than 15 subjects are also possible for studies with a very limited scope.
- **Data Analysis** - After collecting the subjective scores, the subjective scores are processed by applying some statistical analysis tools, as recommended in ITU standards [69]–[74].

In recent years, several subjective quality assessment studies have been conducted with omnidirectional images and video, to assess the perceptual impact of artifacts introduced by compression, transmission, stitching, or by the display. However, little has be done on the subjective impact of geometric distortions introduced by the rendering process. Table 2.4 summarizes the most relevant subjective quality test studies with omnidirectional images/video.

Table 2.4. Subjective quality assessment of omnidirectional image/video.

Method	Artifacts Source	Used display
Pavan et al. [75], Krzysztof et al. [76], Jia et al. [77]	Stitching	HMD
Jill et al. [53][78], Ashutosh et al. [79], Vladyslav et al. [80]	Compression	Computer monitor
Raimund et al. [81]	Transmission	HMD
Raimund et al. [82]	Transmission	Computer monitor
Matt et al. [21], Mai et al. [83], Bo et al. [84], Francisco et. al [85]	Compression	HMD
Evgeniy et al. [68]	Compression	Mobile VR
Wenjie et al. [86]	Display	HMD, Computer monitor
Kim et al. [10]	Rendering	Computer monitor

Regarding the objective quality assessment, most of the objective quality metrics developed so far for omnidirectional visual content aimed to assess the perceptual impact of compression artifacts (e.g., [21], [87]–[95]), or stitching artifacts (e.g., [75]–[77]). Except for a few geometric distortion measures (described in the next section), which were developed for cartography or wide-angle images, there is no objective quality metric designed to automatically assess the perceptual impact of the viewport geometric distortions, due to the sphere to plan projection involved on the rendering process.

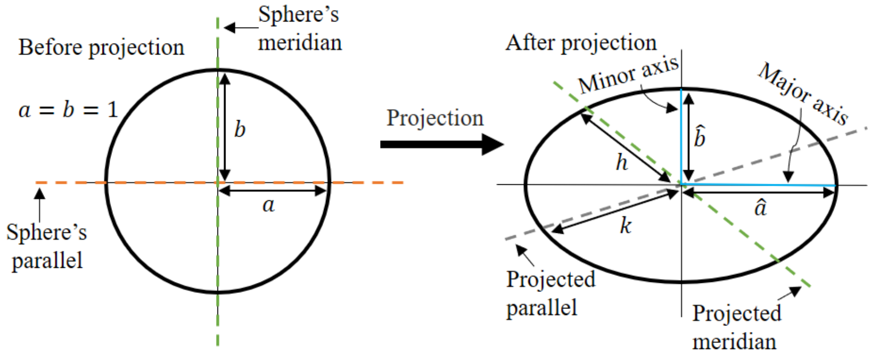
2.6.2 Sphere to Plane Geometric Distortion Measures

In Earth cartography, Tissot indicatrices [96] have been used for years by cartographers to evaluate and compare distortion on different Earth map projections. The metric is based on the geometric relationship between a circle on the sphere and its projection on the plane, called as indicatrix or ellipse of distortion. This indicatrix is obtained after projection, in the map, an infinitely small circle defined on the sphere; the relationship between the major and minor axis of the resulting ellipse, after projection, enables to compute the local scale (or distance), area, and angle distortions, at the projected point.

Figure 2.26 depicts an infinitesimal unit circle defined on the sphere, and its corresponding Tissot indicatrix after projection on the plane; $\hat{a}$ and $\hat{b}$ are, respectively, the Tissot indicatrix semi major and minor axis; h and k correspond, respectively, to the scale factor along the projected sphere meridian and parallel. The details about the computation of the parameters $\hat{a}$, $\hat{b}, h$, k, and angular deformation $\acute{\theta}$, which represents the angle between the projected meridian and parallel at the projected point, are presented in Chapter 4 (Section 4.2.1).

If the projection is conformal (e.g., stereographic), shapes and angles are locally preserved, and the ellipse is a circle; otherwise, the ellipse has a major axis and a minor axis which are directly related to the scale distortion and to the maximum angular deformation. When $\acute{\theta} = 90°$, $\hat{a} = h$ and $\hat{b} = k$. The shape distortion, t, maximum angle deviation, ω, and amount of inflation or deflation in the area, s, are given by

$$t = \frac{\hat{a}}{\hat{b}} \tag{2.49}$$



a) b)

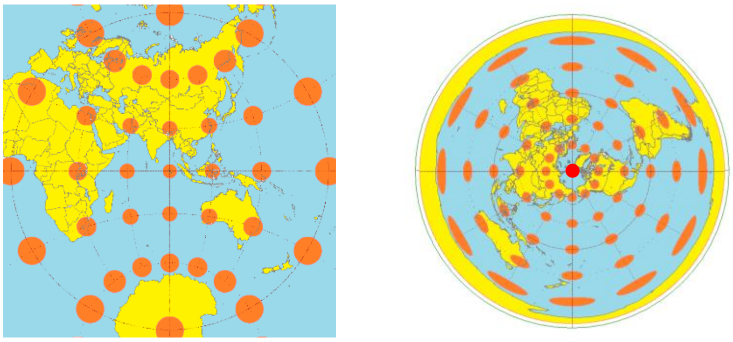
Figure 2.26. a) Infinitesimal unit circle defined on the sphere; b) Its corresponding Tissot indicatrix after projection on the plane.

$$\omega = 2\sin^{-1}\left(\frac{|\hat{a} - \hat{b}|}{\hat{a} + \hat{b}}\right) \tag{2.50}$$

$$s = \hat{a} \times \hat{b}\,. \tag{2.51}$$

A conformal projection has $t = 1$ and $\omega = 0$; since $\hat{a} = \hat{b}$ and $\acute{\theta} = 90°$, each ellipse degenerates into a circle with the radius $\hat{a}$ (or $\hat{b}$) being equal to the scale factor (or stretching) in any direction. If the projection is equal-area, area relationships are locally preserved, and the areas of the circle on the sphere and projected ellipse are the same, and thus $s = 1$.

To show the distortions across an Earth map, the Tissot indicatrices are typically placed over the intersections between projected meridians and parallels, as depicted in Figure 2.27 for the stereographic and azimuthal equidistant projections. For the stereographic, all Tissots indicatrices are circles since this projection is conformal; however, they have different areas, showing that it is not an equal-area projection. For the azimuthal equidistant map projections, the scale is constant along all radial lines (lines crossing the central point, highlighted in red in Figure 2.27b), which is possible to visualize with the Tissot indicatrices, since all indicatrices minor axis have the same length and are always oriented along a radial line. Also, it shows that the shape and area distortions increase with the distance to the central point.



a) b)

Figure 2.27. Tissot indicatrix for a) stereographic projection and b) azimuthal equidistant projection (based on [97]).

The Tissot indicatrix is very useful in the study and evaluation of map projections. More importantly, it quantifies the local scale, area, angular, and shape distortions precisely at each projected point. The Tissot metric could be also used to characterize the geometric distortions introduced during viewport rendering; however, it has some drawbacks: *i)* it does not evaluate the global distortion, but only local (pointwise) distortions; *ii)* it is content independent, i.e., the image content is not considered to evaluate the perceived distortion; *iii)* it cannot measure the bending of the straight lines, which is a geometric distortion type with a strong perceptual impact on the users.

More recently, a few geometric distortion measures were proposed in the context of content-aware projections for wide-angle images, seeking to minimize the resulting geometric distortions. As described in Section 2.5.1, in [36] local conformality and line straightness measures were proposed and used to adapt the projection locally to the image content and thus reducing the geometric distortions. The conformality measure is computed based on Cauchy-Riemann equations [98], which are verified by a conformal projection

$$\frac{\partial x_p}{\partial \theta} = -\frac{\partial y_p}{\partial \phi} \times \frac{1}{\cos(\theta)}\,, \quad \frac{\partial y_p}{\partial \theta} = \frac{\partial x_p}{\partial \phi} \times \frac{1}{\cos(\theta)}\,. \tag{2.52}$$

Accordingly, the conformality measure, denoted as E_c in (2.47), is given by

$$E_c = \left(\frac{\partial y_p}{\partial \phi} + \cos(\theta)\frac{\partial x_p}{\partial \theta}\right)^2 + \left(\cos(\theta)\frac{\partial y_p}{\partial \theta} - \frac{\partial x_p}{\partial \phi}\right)^2. \tag{2.53}$$

For a conformal projection E_c has a value close to zero. The global conformality measure can be obtained by summing the local measures computed for all projected points.

The line straightness measure is computed based on the geometry of straight lines and aims to preserve the image linear structures. The line straightness measure, denoted as E_{ld} in (2.47), is given by

$$E_{ld} = \frac{\sqrt{\left(x_p^s - x_p^e\right)^2 + \left(y_p^s - y_p^e\right)^2}}{\sqrt{\left(x_p^s - x_p^e\right)^2 + \left(y_p^s - y_p^e\right)^2} + \left|x_p^s\left(y_p^e - y_p^m\right) + x_p^e\left(y_p^m - y_p^s\right) + x_p^m\left(y_p^s - y_p^e\right)\right|}, \tag{2.54}$$

where (x_p^s, y_p^s), (x_p^e, y_p^e), and (x_p^m, y_p^m) correspond, respectively, to the line start, end, and middle points of the projected line. Note that this measure is normalized by the line length, and thus for a straight line E_{ld} has a value close to 1, and for a curved line E_{ld} has a value lower than 1. The global line straightness measure can be obtained by averaging the line straightness measures computed for all projected lines. In (2.47), and besides E_{ld}, another line distortion measure is described, E_{lo}, which is similar to E_{ld}, except that E_{lo} considers the line direction into accounts. Thus, E_{lo} aims the line bending minimization while keeping the line direction specified by the user (if any); while E_{ld} aims the minimization of the bending, regardless of the line direction. Note that, E_{ld} and E_{lo} need to be coherent with other terms used in (2.47) and (2.48), thus both E_{ld} and E_{lo} need to be subtracted from 1 and then used in (2.47) and (2.48).

However, these measures have some drawbacks: *i)* similar to Tissot, the conformality measure is content independent; *ii)* the line straightness measure requires user interaction to manually identify the perceptually important straight lines in the scene. Also, these measures were not validated with respect to perceived geometric distortions. Thus, there is no evidence that these measures are well correlated with user perception of geometric distortions. These measures, without validation, were also used in [10] to optimize the Pannini projection parameters.

In (2.53), the partial derivatives need to be obtained beforehand. An alternative way to compute a conformality measure, without requiring the partial derivatives, was proposed in [10]. In this case, four points are sampled, on the spherical image, around each salient point, *i*; the sample points are some degrees apart (e.g., 0.5 degrees), to the left, right, up, and down of the salient point. When the points are projected on the plane, the distances d_i^l, d_i^r, d_i^u, and d_i^d, between the i-th projected salient point and its four projected sample points, are then computed. From these distances, the conformality measure, $\hat{E}_c$, is obtained as

$$\hat{E}_c = \frac{1}{N}\sum_i \left(\frac{\mathrm{Min}(d_i^r, d_i^l, d_i^u, d_i^d)}{\mathrm{Max}(d_i^r, d_i^l, d_i^u, d_i^d)}\right), \tag{2.55}$$

where N is the total number of salient points.

2.7 Final Remarks

This chapter presented the omnidirectional image/video processing pipeline and reviewed the state of the art on sphere to plane projections and on the quality assessment of the resulting geometric distortions. Several content-unaware and content-aware projections were described, and qualitatively evaluated. The qualitative evaluation showed that the different projections present a tradeoff between the different types of geometric distortions, and no projection can avoid geometric distortions. Moreover, in general, content-aware projections have less geometric distortions than content-unaware projections.

Finally, the quality assessment of omnidirectional images, with the focus on the geometric distortions, was reviewed both in terms of subjective quality studies and objective quality assessment metrics. This showed that there is little work on the perceptual impact of geometric distortions resulting from sphere to plan projections, and the related literature is rather scarce. Accordingly, this Thesis seeks to fill this gap by assessing, through subjective experiments, and quantifying, through the development of objective metrics, the geometric distortions introduced during the viewport rendering of omnidirectional visual content.

Chapter 3

Subjective Assessment of the General Perspective Projection

3.1 Introduction

As mentioned in Chapters 1 and 2, objects stretching (or shearing) and bending of straight lines are the two main geometric distortion types introduced during the viewport rendering of omnidirectional visual content, due to the sphere to plane projection (*cf.* Figure 1.4, Figure 2.12, Figure 2.25). However, there are not many subjective quality assessment studies in the literature that evaluate the perceptual impact of these distortions. Therefore, the first step in this chapter is to study and evaluate, through a subjective tests campaign, the perceptual impact of geometric distortions, using the general perspective projection (GPP) for viewport rendering. As mentioned in Chapter 2, GPP includes the rectilinear (or gnomic) and stereographic projects, which are the most used solutions for viewport rendering. Furthermore, the GPP allows to vary (in type and strength) the geometric distortions, by controlling the GPP projection center value; thus, a wide range of geometric distortions can be obtained, enabling the subjective assessment of their impact and thus the creation of a diverse and rich dataset, needed for the development and validation of an objective quality metric.

To enhance the user sense of immersion and engagement when exploring omnidirectional visual content, the viewport FoV should be large. However, the geometric distortions become more perceptually disturbing when a large FoV is used. Thus, the used FoV may play an important role on the user's QoE. So far, there are not many subjective quality assessment studies that assess the FoV effect on the perceived viewport quality; in particular, there is no clear evidence about the range of FoVs that should be used for viewport rendering, nor its dependency on the image content type. Thus, the second step in this chapter focuses on the study and evaluation of the FoV impact on the perceived quality of omnidirectional visual content rendering. This study is also based on a subjective tests campaign with viewport images, in this case, rendered with the rectilinear projection.

In this context, this chapter addresses the two main objectives:

- Subjectively assess the perceptual impact of geometric distortions introduced by the GPP, notably stretching of objects and bending of straight lines.
- Subjectively assess the FoV impact on the perceived quality of the viewport image, to: *i)* evaluate its eventual dependency on the image content; *ii)* determine the FoV that presents the best trade-off between user's immersive experience and the perceived visual

degradations due to geometric distortions, and when the rectilinear projection is used for viewport rendering.

Since, as stated in Chapter 1, the target application scenario for this Thesis is the visualization of omnidirectional images on typical 2D displays (e.g., smartphones and personal computers), both studies are accomplished using 2D screens.

This chapter is organized as follows: Section 3.2 describes the subjective assessment of geometric distortions impact. Section 3.3 describes the subjective assessment of the FoV impact. Finally, in Section 3.4, some final remarks are presented.

3.2 Subjective Assessment of the Geometric Distortions Impact

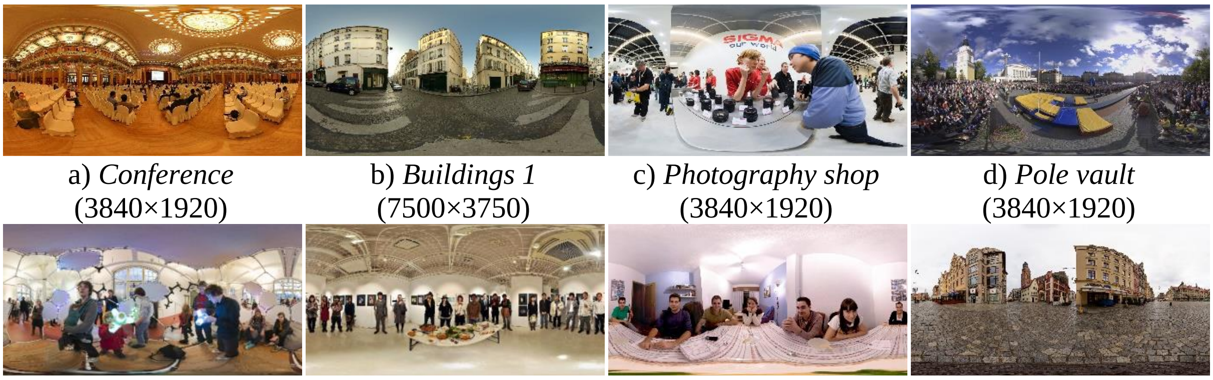
This section describes the subjective assessment of the geometric distortions impact, using the GPP for viewport rendering. To the best of our knowledge, this is the first subjective test to evaluate the geometric distortions introduced during rendering. It is important to note that subjective tests are typically very time-consuming, and in this case they were conducted in several sessions. In each session, new omnidirectional images were included and evaluated, and the resulting quality scores and viewport images were added to the GPP viewport dataset, which was made publicly available [99].

The rest of this section is organized as follows: Sections 3.2.1 and 3.2.2 describe, respectively, the considered omnidirectional image dataset and the subjective evaluation methodology. Sections 3.2.3 and 3.2.4 present, respectively, the subjective test results, and the analysis of the subjective scores, together with the main conclusions taken from the results.

3.2.1 Dataset

The subjective assessment of the GPP was conducted using eight omnidirectional images, in equirectangular format, taken from the Salient360! dataset [52]. The images, and their spatial resolutions, are depicted in Figure 3.1. The selected images have different types of content, including indoor and outdoor scenes, objects near and far away from the camera, the presence or absence of people, and horizontal and vertical lines in the scene. Thus, they exhibit different types of dominant distortions (from bending to stretching) when the GPP is used for viewport rendering, with different values of the projection center, d. For each image, the viewports were rendered for three different viewing directions: front view ($\phi_{VD} = 0, \theta_{VD} = 0, \psi_{VD} = 0$), 45° to the right ($\phi_{VD} = 45, \theta_{VD} = 0, \psi_{VD} = 0$), and 45° to the left ($\phi_{VD} = -45, \theta_{VD} = 0, \psi_{VD} = 0$); together, the three selected views cover a large part of the viewing sphere where the users attention is often attracted for, as the regions located around latitude zero (equatorial line) and inside, or close to, the front viewport [100].

For each viewing direction, ten viewports were produced, each one corresponding to a pair (d,FoV), with $d \in \{0,0.25,0.5,0.75,1\}$ and FoV $\in \{90°, 110°\}$. Higher values of d have not been considered, since the amount of distortion (fisheye effect) introduced by these projections is visually annoying to many viewers, and also to limit the test duration. The FoV of $90°$ and $110°$ were selected since these values are often used in VR applications. The viewports were rendered using the GPP, and with a spatial resolution of 856×856 pixels ($AR = 1$); besides being recommended in [53] for subjective tests, this resolution allows the simultaneously display of two viewports, side by side, in typical monitors.



a) *Conference* (3840×1920)

b) *Buildings 1* (7500×3750)

c) *Photography shop* (3840×1920)

d) *Pole vault* (3840×1920)

e) *Museum* (3840×1920)

f) *Dinner 2* (7500×3750)

g) *Friends* (3840×1920)

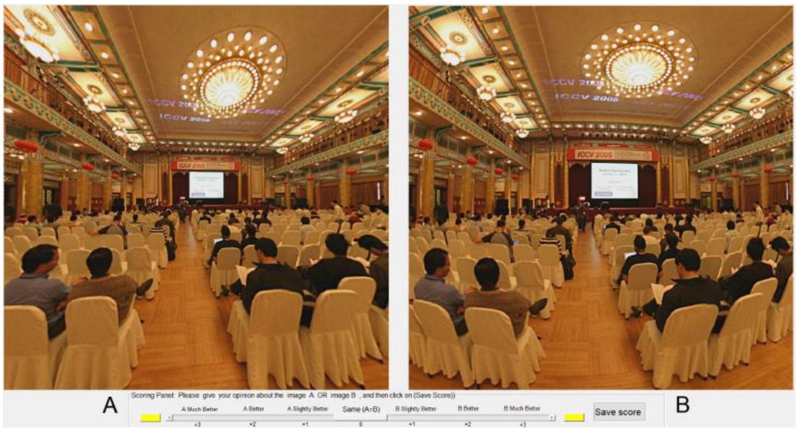
h) *Buildings 2* (7500×3750)

Figure 3.1. Omnidirectional images used in the subjective tests, and their spatial resolution.

3.2.2 Subjective Evaluation Method

The Stimulus Comparison Adjectival Categorical Judgment (SCACJ) [70] was selected as the evaluation method, since it is easier for observers to select the most pleasant viewport, in a pair of viewports rendered with different projections, than to directly rate the viewports. In this case, two viewports are shown simultaneously side by side, and the observer is asked to give his opinion about the quality of displayed viewports, giving a score to the most pleasant viewport using the following comparison scale: slightly better (+1), better (+2), much better (+3), or 0 in case no difference was detected. Since the rectilinear projection $(d = 0)$ is typically used for the rendering of omnidirectional images and videos, the resulting viewports were used as the reference stimulus.

A subjective assessment interface, including a graphical user interface (GUI), was designed to perform the visualization of the viewport images and to collect the associated subjective scores. Two viewport images were shown side by side, as depicted in Figure 3.2, one being the reference viewport $(d = 0)$, and the other being the viewport under evaluation. To minimize the contextual effects, the viewports were shown in random order and position, such that the position of the reference is either on the right side or on the left side, and the viewport pairs from the same omnidirectional image were never consecutively displayed. A total of 192 stimuli was evaluated by each observer (4 (d) × 2 (FoV) × 3 (viewing directions) × 8 (images) = 192).



Scoring Panel: Please give your opinion about the image A OR image B , and then click on (Save Score))

Figure 3.2. Subjective assessment interface designed to perform the visualization of the viewport images and to collect the associated subjective scores.

Before the subjective test, the observers were asked to read short instructions about the test procedures, so that they could understand the task. Subsequently, more detailed instructions were shown on the screen during a short training session, right before the test, to familiarize the observers with the projection distortions characteristics and the evaluation interface; the viewports used in this training session were not used for the actual test. During the test phase, the observer gave a score between +1 (slightly better) and +3 (much better) to the most pleasant viewport, or a score of 0 in case of similar quality. This score was associated to the viewport under evaluation, if it was considered the most pleasant one; otherwise, the symmetric score was given to it. Thus, any viewport (except the reference ones) got a final score between −3 and +3.

The subjective test was conducted with a 2D display, as recommended in the MPEG group [8], using a Full HD monitor, with a native resolution of 1920×1080 pixels. In total, 20 observers, aged between 22 and 35 years, were asked to participate in the subjective evaluation. The participants were seated at the distance, from the monitor screen, of approximately three times the picture height, as suggested in [73]. The omnidirectional images, the rendered viewports, and the resulting subjective score values were made available in [99].

3.2.3 Subjective Tests Results

During the subjective test, every viewport (except the reference ones) got a final score between −3 and +3; these scores were then normalized to the interval [0,10], as recommended in [70]. Outliers detection was applied according to the guidelines in [70], but it was verified that none of the viewers' scores deviated strongly from others (i.e., no outliers were detected). The comparative mean opinion score (CMOS) [71] was then computed for each viewport, according to

$$\mathrm{CMOS}_i = \frac{1}{O}\sum_{j=1}^{O} \mu_{\mathrm{ij}}\,, \tag{3.1}$$

where CMOS_i is the resulting CMOS for the viewport (or stimulus) i, μ_{ij} is the score given by observer j to the viewport i, and O is the total number of observers after outlier's removal. The reliability of the subjective assessments was determined by computing the 95% confidence interval associated with the CMOS scores, i.e., $[\mathrm{CMOS}_i\ - \delta_i,\ \mathrm{CMOS}_i + \delta_i]$, where:

$$\delta_i = 1.96\frac{\sigma_i}{\sqrt{O}} \tag{3.2}$$

and σ_i is the standard deviation of the scores for each stimulus, given by

$$\sigma_i = \sqrt{\sum_{j=1}^{O} \frac{\left(\mathrm{CMOS}_i - \mu_{ij}\right)^2}{O-1}}. \tag{3.3}$$

Figure 3.3 depicts the resulting CMOS values for each viewport and associated 95% confidence intervals. The smallest confidence intervals occurs when one of the comparing viewports shows an evident global distortion (e.g., stretching or bending) relatively to the other; the largest confidence intervals result for those cases where each comparing viewport shows a different distortion type, and the viewer gives a score based on which type is most/least annoying for him.

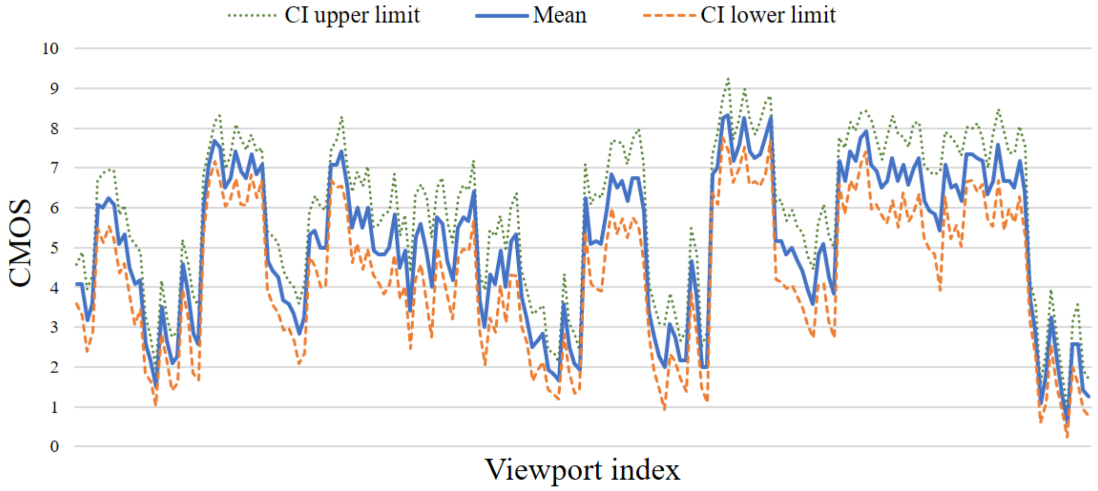


Figure 3.3. CMOS values for the evaluated viewports, with associated 95% confidence interval (CI) limits.

Figure 3.4 shows the CMOS values for each considered image and FoV (averaged over the three viewing directions), versus projection center, d; the GPP particular cases for $d = 0, 0.25, 0.5, 0.75, 1$ are referred to as pr_0, pr_1, pr_2, pr_3 and pr_4, respectively. Due to the subjective scores normalization, a CMOS of 5 represents a viewport quality undistinguished from the reference one (pr_0); a score of 0 represents a viewport quality much worse than the reference; a score of 10 represents a viewport quality much better than the reference.

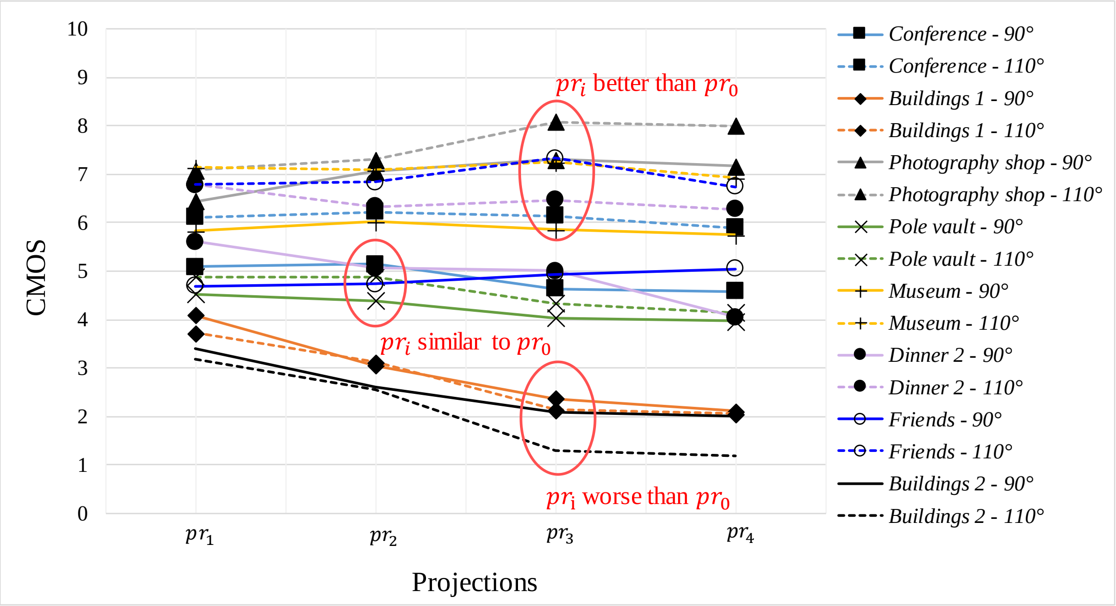


Figure 3.4. CMOS values as a function of the projection type, for the considered test images and FoV.

3.2.4 Subjective Tests Analysis

As shown by Figure 3.4, the best projection depends strongly on the image content and the rectilinear projection (pr_0), often used in omnidirectional viewing systems, is not always the best. Three main groups can be identified, regarding how the viewport quality resulting from a given projection, pr_i, compares with the viewport quality for the rectilinear projection, pr_0:

- **G1)** $\boldsymbol{pr_i}$ **better than** $\boldsymbol{pr_0}$ - which happens for the cases where the stretching associated to pr_0 is the dominant distortion, as in those viewports with regions of interest close to the viewport borders, and where the stretching is maximum.

- **G2)** $\boldsymbol{pr_i}$ **similar to** $\boldsymbol{pr_0}$ - which happens whenever there is no dominant distortion type, and the subjects do not show an overall preference for any particular projection.

- **G3)** $\boldsymbol{pr_i}$ **worse than** $\boldsymbol{pr_0}$ - which happens for the viewports where the bending is the dominant distortion, typically viewports containing long straight lines (e.g., *Buildings 1* and *Buildings 2*). These lines bend when d approaches 1, which negatively influences the perceived quality.

To evaluate if the difference in CMOS values between these three groups is statistically significant, and following the procedure suggested in [101], the analysis of variance (ANOVA) with three groups and a significance level of 0.05 was applied per projection, and for each FoV. The resulting *p*-value, shown in Table 3.1, is always lower than 0.05, which confirms that the separation between the three groups is statistically significant. This result also allows to conclude that the impact of projection type (i.e., pr_i), is dependent on the considered image.

To further evaluate if the dependency of CMOS values from the FoV is statistically significant, a paired sample T-test was applied comparing two sets of samples, one containing the viewports with FoV of 90° and another containing the viewports with FoV of 110°. This procedure, also suggested in [101], is illustrated in Figure 3.5. The test results indicate that the null-hypothesis, i.e. that the two sets have the same mean, can be rejected with a *p*-value of 0. This confirms that the FoV has a significant impact on the perceived geometric distortion.

Finally, to evaluate if the dependency of CMOS values from the projection center is statistically significant (for the same content and FoV), the previous procedure was applied, but now comparing the viewports pairs corresponding to the same image and viewing direction, resulting from a pair of projections, (pr_i, pr_j), with $i \neq j$ (in Figure 3.5, pr swaps with FoV). In this case, the percentage of projection pairs resulting in viewports with significantly different perceived quality is quite dependent on the images group, being 40% for G1, 10% for G2 (a low percentage was expected in this case, since pr_i has a similar quality to pr_0, as shown by Figure 3.4), and 90% for G3.

From this analysis, it is possible to conclude that, in the viewport rendering of omnidirectional images:

- The rectilinear projection, often used for viewport rendering, is not always the projection that leads to the best viewport quality.

- The viewport content has an important impact on the perceived distortions, and thus condition the best projection, to be used.

- The projection type (d value), the considered FoV, and the viewport content characteristics, are the three main influence factors of the perceived geometric distortions.

Table 3.1. Resulting p-value for ANOVA test.

	p-value per projection			
FoV	pr_1	pr_2	pr_3	pr_4
90º	1.19E-04	3.42E-06	0	0
110º	0	0	0	0

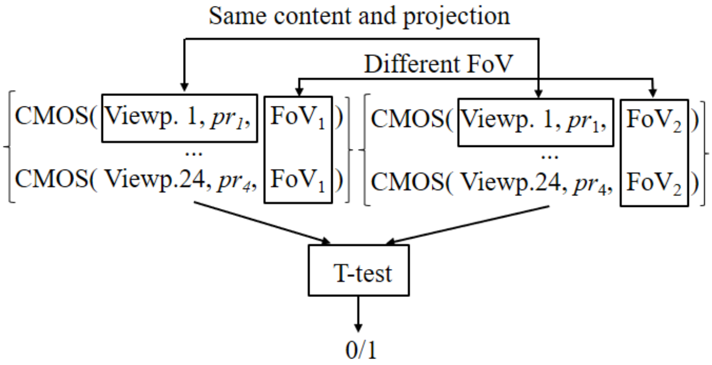

Same content and projection
Different FoV
CMOS(Viewp. 1, pr_1, FoV$_1$)
...
CMOS(Viewp.24, pr_4, FoV$_1$)
CMOS(Viewp. 1, pr_1, FoV$_2$)
...
CMOS(Viewp.24, pr_4, FoV$_2$)
T-test
0/1

Figure 3.5. T-test procedure for evaluating if there is a significant impact of the FoV on the CMOS scores (based on [101]).

It should be mentioned that, on the first subjective test session, where a subset of the omnidirectional images dataset was used, namely *Conference*, *Buildings 1*, *Photography shop*, and *Pole vault*, the FoV of 75° was also considered, besides the FoVs of 90° and 110°. Figure 3.6 shows the resulting CMOS values for these images (averaged over the three viewing directions) versus considered projection. While it is usually assumed that pr_0 (rectilinear projection) performs well for a FoV of 75°, from Figure 3.6 it can be concluded that, for some images, other projections perform better than rectilinear, notably $pr_1$to pr_4 for *Conference* with FoV of 110°, and pr_1 to pr_4 for *Photography shop* with FoVs of 75°, 90°, 110°. In fact, even for a FoV of 75°, pr_0 yields noticeable stretching distortion in objects that are simultaneously located at the viewport borders and close to the camera.

To limit the subsequent subjective tests duration, preventing the results from being influenced by the viewers fatigue, the FoV of 75° was not kept for the remaining omnidirectional images; also, larger FoVs are closer to the human FoV.

The subjective test results allowed to conclude that the FoV value has a statistically significant impact on the perceived geometric distortions. Accordingly, further studies were conducted to acquire a better understanding of the FoV influence on the QoE, and also to find out a possible dependence, from the viewport content, of the FoV value that maximizes the QoE (since this dependency was verified for the GPP projection center). These studies are presented on the next section.

3.3 Subjective Assessment of the FoV Impact

The subjective tests described on the previous section, showed that the GPP projection type (or d value) that minimizes the geometric distortions is dependent on the viewport content. This section evaluates, through an additional subjective tests campaign, if a similar dependence exists for the viewport FoV; furthermore, the impact of the FoV on the users's QoE is further investigated.

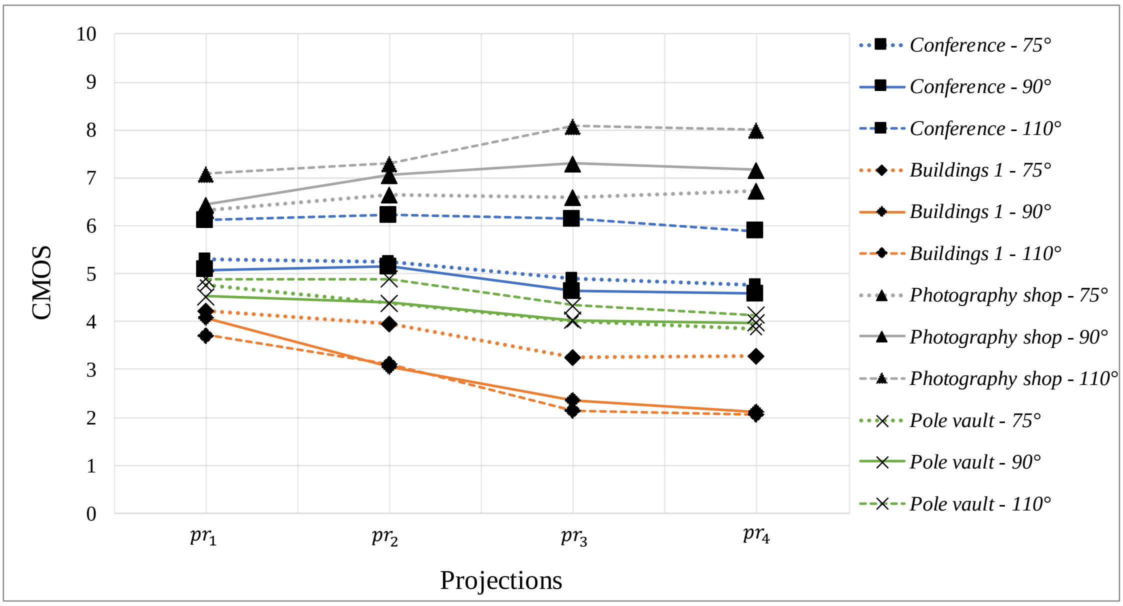


Figure 3.6. CMOS values as a function of the projection type, for the four omnidirectional images, with FoV of 75°, 90°, and 110°.

As mentioned in Chapter 1, the geometric distortions increase with the used FoV. Despite of this, the rendering of omnidirectional visual content should provide an immersive visual experience and maximize the users sense of presence. As shown by several studies (e.g., [11][12]), to meet these requirements large FoVs should be used. Recently, most of the VR applications that make use of omnidirectional images are targeting a FoV close to human FoV (e.g., [102][103]), aiming to provide a better QoE to the users. In fact, in the area of virtual reality, the abstract concept of immersion can be measured by relating the human FoV, and the observed area on the sphere that is shown to the user [104]. For the horizontal direction, the human FoV is on the range 200°-220° for monocular vision, and around 114° for binocular vision; the vertical FoV is on the range 130°-135°.

Since the FoV has an important role on the user's QoE, the study and evaluation of the FoV impact are much needed. So far, few works (e.g., [11][12][105][106]) have considered the influence of the FoV on user's QoE. In [105], the authors evaluated the viability of high FoVs (namely, 110°, 140° and 170°) in computer graphics rendition. Three different projection methods, namely rectilinear, Panini, and stereographic, were considered. However, the main goal was to compare the impact, on the rendered images quality, of the three considered projection methods, with varying FoV; furthermore, the test conditions only considered the projection of 3D virtual environments on 2D displays. Also, during the subjective assessment tests, the rendered content was presented to the viewers as still images, corresponding to the viewport on a fix and predefined viewing direction. In [11], a driving simulator was used to assess the impact of FoV on the user's experience; four FoVs were consider, namely 60º, 100º, 140º, and 180º. The test included a full-size car, and a virtual world was created around the car using a set of projection walls and cameras. The results indicated that the user's presence and enjoyment increased with the FoV; however, FoVs beyond 140º could conduct to simulator sickness. Additional studies on the FoV impact were carried out in [12][106], with FoV values of 60º and 100º in [12], and varying between 10º and 110º in [106]; in both works, 3D virtual environments were rendered on 2D displays. The experimental results showed that the users needed less time to achieve visual search tasks, if a large FoV is used.

In all previous works, no conclusions were drawn about the best FoV for omnidirectional visual content rendering. Therefore, the subjective test campaign presented in this section was specifically designed with the objective of finding the FoV value that presents the best trade-off between immersivity and geometric distortions perception, and to assess its dependency from the image content. Furthermore, a method to generate navigation videos from 360^o images, using real head motion (navigation path or scanpath), is proposed; besides simulating the user navigation on omnidirectional images, these videos allow the comparison of results across different subjects for each subjective test condition. Moreover, evaluating the user immersivity provided by different FoVs should be done preferable with varying viewing direction (i.e., not using "still images"), which is the natural way to navigate the omnidirectional visual content. One way to do this is to generate navigation videos from 360^o images. This is also supported by some work conducted by the MPEG group [107].

The rest of this section is organized as follows: Section 3.3.1 describes the proposed procedure to create a viewport video from real head motion; Section 3.3.2 is dedicated to the subjective tests procedure, including subjective test results and analysis.

3.3.1 From Head Motion Data to User Navigation Video

This section details the procedure to create a video that simulates the user navigation when exploring an omnidirectional image. First, the method to select a representative navigation path, from all the scanpaths available for a given image, is detailed. Then, the viewport projection and the video creation are described.

A. Navigation Path Selection

In [100][108], a dataset of omnidirectional images, designated by Salient360!, was proposed. The dataset includes also the scanpaths from head movement of several observers (per image), recorded while the observers explored the images with HMD, and the visual saliency maps obtained from the recorded scanpaths. Each scanpath contains 100 samples, taken with a fix time period, t_s, where the i^{th} sample corresponds to a viewing direction, VD_i, with spherical coordinates (ϕ_i, θ_i) (*cf.* Figure 1.3a).

Figure 3.7 presents one of the omnidirectional images available in Salient360!, the corresponding saliency map, and the navigation paths of three observers, drawn in green, yellow, and black over the saliency map. On the saliency map, the hottest coloured regions indicate the most salient regions and the blue areas indicate least salient regions.

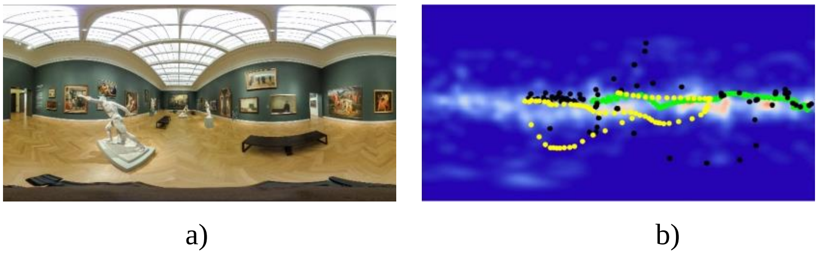


Figure 3.7. a) An omnidirectional image in equirectangular format; b) The saliency map generated from the head motion of several observers, and the navigation path for three different observers.

Considering the final application of the simulated navigation video - subjective assessment of the FoV effect on QoE - from the available paths, the selected path (per image) is the one that best fulfil the following criteria: *i)* slow temporal variation; *ii)* wide coverage of the omnidirectional image, thus without being stuck in the same positions; *iii)* close to the salient regions, since distortions over the salient regions have more perceptual impact than in other regions.

Accordingly, for each navigation path, three features were extracted: velocity, $\mathcal{v}$, wideness, $\mathcal{w}$, and saliency, $\mathcal{S}$, defined as:

- **Navigation path velocity, $\boldsymbol{\mathcal{v}}$ -** the distance between adjacent viewing directions, $VD_i(\phi_i, \theta_i)$, and $VD_{i+1}(\phi_{i+1}, \theta_{i+1})$, over the unit sphere, is given by

$$d_i = \cos^{-1}(\sin\theta_i \sin\theta_{i+1} + \cos\theta_i \cos\theta_{i+1} \cos(\phi_i - \phi_{i+1}))\,, \tag{3.4}$$

  and the velocity of the head movement between VD_i and VD_{i+1}, is given by $\mathcal{v}_i = d_i/t_s$. The global path velocity value, $\mathcal{v}$, is computed by

$$\mathcal{v} = \frac{1}{N-1}\sum_{i=1}^{N-1} \mathcal{v}_i\,, \tag{3.5}$$

  where N is the number of samples in the path.

- **Navigation path wideness, $\boldsymbol{\mathcal{w}}$ -** The distance between adjacent viewing directions over the path can be computed using (3.4), and the summation of all these distances gives the path length. However, a high path length does not necessarily represent a wide path, since an observer may just navigate in a small area of the viewing sphere, without fully exploring its content. Instead, the path wideness, $\mathcal{w}$, is computed as the distance between the path starting point, $VD_1(\phi_1, \theta_1)$, and its farthest viewing point.

- **Navigation path saliency, $\boldsymbol{\mathcal{S}}$ -** Geometric distortions within salient regions are perceptually more relevant than in other regions. Therefore, the selected path should have most of its sample points close to salient regions. To compute $\mathcal{S}$, each data point (ϕ_i, θ_i) of the navigation path is first projected on the equirectangular image saliency map. The relationship between spherical coordinates, (ϕ_i, θ_i), on the unit radius sphere, and pixel coordinates, (m_{ERI}, n_{ERI}), on the equirectangular representation of the sphere, is given by (2.45) and (2.46). In Salient360! dataset, to obtain the saliency map, an isotropic 3.34-degree Gaussian foveation filter, centred in a set of viewport locations, was applied to all scanpaths. Therefore, for each resulting pixel position, (m_{ERI}, n_{ERI}), a saliency value is obtained by applying a similar Gaussian filter over the saliency map; thus, not only the saliency at (m_{ERI}, n_{ERI}), but also the saliency at neighbouring locations, are considered. Finally, the path saliency, $\mathcal{S}$, is computed by averaging the saliency values obtained along the path.

After having computed $\mathcal{v}, \mathcal{w}, \mathcal{S}$ for all the navigation paths available for a given omnidirectional image, each feature is normalized by the corresponding maximum range (found on the computed feature values). The selected path, SP, is then obtained by

$$SP = \max_k(\alpha_1 \mathcal{S}_k + \alpha_2 \mathcal{w}_k + \alpha_3 \frac{1}{\mathcal{v}_k}) \tag{3.6}$$

where $\mathcal{S}_k, \mathcal{w}_k, \mathcal{v}_k$ are the normalized features computed for the k^{th} navigation path, and

α_1, α_2, α_3 are weighing parameters that control the importance of each feature. For the results presented in this Thesis, $\alpha_1 = 2$, $\alpha_2 = \alpha_3 = 1$. These values were obtained experimentally, by visual inspection of the selected navigation paths, and resulting navigation videos, for some omnidirectional images.

By applying the described procedure to the three navigation paths depicted in Figure 3.7b), the colored in green was the selected one; this path goes through the salient regions and explores well the content across the image. After selecting the navigation path, using all the available navigation paths, the corresponding video is created, as described in the next section.

B. Navigation Video Rendering

For each viewing direction, the corresponding viewport is obtained by projecting a fraction of the omnidirectional image on the image plane, using the rectilinear projection. The viewport rendering process is formally described in Chapter 2 (Section 2.4.8.A). In this study, only the rectilinear projection was considered, since: *i)* it is the most used projection for VR applications; *ii)* to evaluate if the best FoV is dependent, or not, on the image content, the chosen d value should not be critical; *iii)* including, in the subjective test, several omnidirectional images, different projections and FoV values, would result in a too large number of viewports to be evaluated by the observers, causing the observers tiredness and/or fatigue.

Since the navigation paths available in Salient360! were recorded with a sampling period, t_s, of 0.25 s, the number of viewing directions per second is just four. Thus, to produce a navigation video with a reasonable frame rate, and sufficient to reproduce a continuous head motion, more viewing points per second are required; these were obtained by linearly interpolating fourteen additional viewing points between each pair of adjacent samples of the recorded paths, resulting in 60 viewing points per second. These were then halved to obtain navigation videos with 30 frames/s.

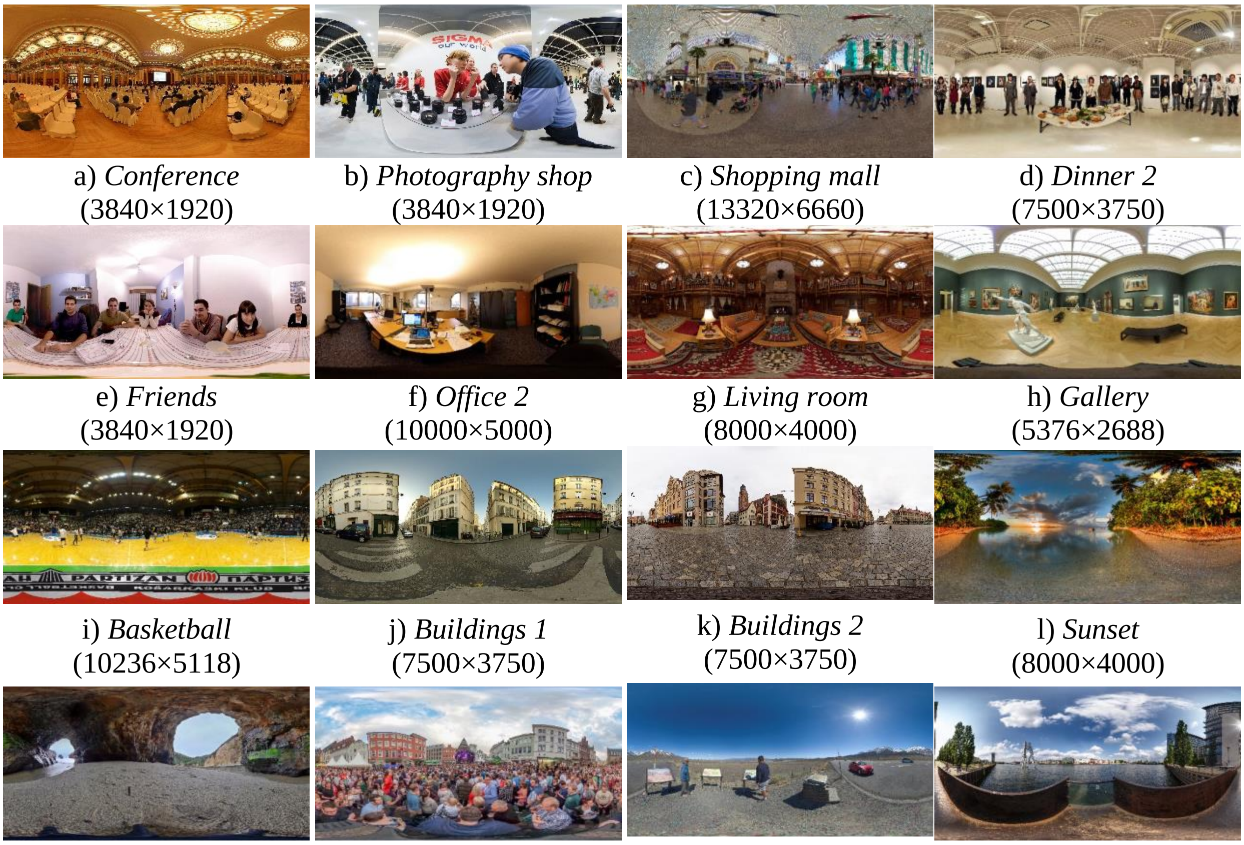
It is important to mention that the proposed procedure to generate navigation videos from omnidirectional images could be potentially used as a basis for other interesting applications, such as omnidirectional video summarization and cinematography, where 2D videos are produced from omnidirectional videos [109].

3.3.2 Subjective Assessment of the FoV Impact

This section presents the subjective test evaluation of the FoV effect on perceived quality. After describing the considered dataset and subjective test methodology, the processing of the resulting subjective scores is described. Finally, the subjective test results are analyzed.

A. Dataset

Sixteen omnidirectional images, taken from Salient360! dataset, were used in the subjective test. The images, and their special resolutions, are depicted in Figure 3.8. This set of images includes six images that were used in the previous subjective test, and ten additional images having other content types, such as indoor and outdoor scenes, presence or absence of people, objects close or far from the camera, and nature or urban environments. After selecting the navigation path for each image, as described in Section 3.3.1.A, six navigation videos were produced per image, as described in Section 3.3.1.B, with a length of 10 seconds each; the chosen frame resolution was $W_{vp} = 1816$ and $H_{vp} = 1020$ pixels, corresponding to a



a) *Conference* (3840×1920)
b) *Photography shop* (3840×1920)
c) *Shopping mall* (13320×6660)
d) *Dinner 2* (7500×3750)
e) *Friends* (3840×1920)
f) *Office 2* (10000×5000)
g) *Living room* (8000×4000)
h) *Gallery* (5376×2688)
i) *Basketball* (10236×5118)
j) *Buildings 1* (7500×3750)
k) *Buildings 2* (7500×3750)
l) *Sunset* (8000×4000)
m) *Cave* (5376×2688)
n) *Concert* (5376×2688)
o) *Desert* (16000×8000)
p) *River* (10000×5000)

Figure 3.8. Omnidirectional images used in the subjective tests, and their spatial resolution.

$AR = 16/9$, as recommended in [53]. The aspect ratio of 16/9 allows to show the navigation videos with a full screen size, being more immersive when compared to an aspect ratio of 1. Each video has a distinct horizontal field of view (F_h), with $F_h \in \{75°, 90°, 100°, 110°, 120°, 135°\}$. The resulting navigation videos are available in [110].

B. *Subjective Assessment Methodology*

A preliminary subjective test with few observers showed that it was difficult for them to concentrate on navigation videos if they are shown side by side, since the two videos have different FoV. Moreover, showing just one video at a time provides a better immersivity, since the whole display can be used for it. Accordingly, the two navigation videos under comparison were shown one after the other, and a simple assessment method – the pairwise comparison (PC) [69] – was selected. PC is commonly used for image and video quality assessment (e.g., [111][112]); it is simple and easy for the observers, since they just need to indicate the most pleasant navigation video, in a pair of videos rendered with different FoVs.

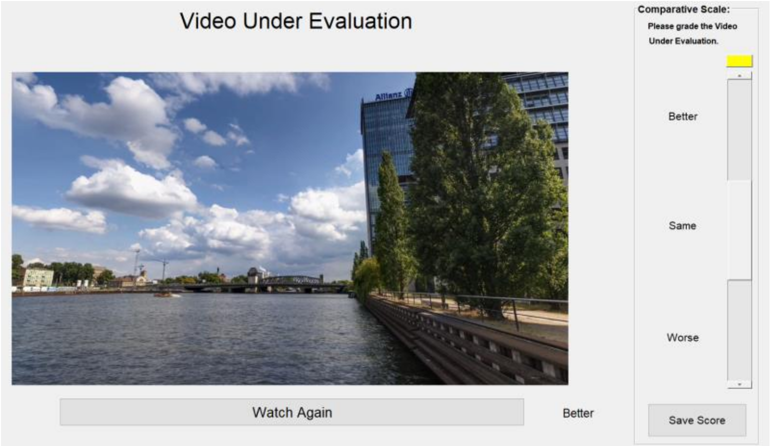
However, comparing all possible pairs is unfeasible due to the quadratic growth of comparisons, with the number of stimuli. To limit the test duration and avoid the observer fatigue, the navigation videos from the same omnidirectional image were arranged in pairs, according to the corresponding F_h value: $\{75^{o},90^{o}\}$, $\{90^{o},100^{o}\}$, $\{100^{o},110^{o}\}$, $\{110^{o}, 120^{o}\}$, $\{120^{o}, 135^{o}\}$, and only these pairs were compared. It is worthy to note that the same error made when comparing distant (in terms of perceived quality) stimuli has a higher impact on the final stimuli rank, than when comparing close stimuli. Accordingly, it is preferable to omit pairs with farther FoVs.

Moreover, decisions about pairs with closer FoVs have higher informative value and increase the discriminatory power of the test [113].

To form circular triads that help the outlier detection process (details are provided in next section), the test images were randomly split in two groups, and an additional comparison (pair of videos) was included, with F_h ∈{90º,110º} for the images in the first group, and F_h ∈{100º, 120º} for the images in the second group. As in the previous subjective tests campaign, before starting the evaluation of the videos, the observers read short written instructions about the test procedures and participated in a short training session to understand the objectives of the test and the evaluation interface. The viewport videos used in the training session were not used for the actual test.

During the test session, each video (or stimuli) of each pair of navigation videos was shown to the observers for 10 seconds, one after the other, and the observers were asked to rate the second video, with respect to the first video, according to the following grading scale: +1 (better), 0 (same), or -1 (worse); an asymmetric score is automatically associated to the first video. Thus, six comparisons were made for each omnidirectional image, shown in random order. It is worthy to note that the observers were allowed to watch the navigation videos multiple times, before rating them; as shown in Figure 3.9, when the observer clicks on “Watch Again”, both navigation videos were shown again, in the same order, one after another. To limit the test duration to half an hour at most, avoiding the observer’s fatigue, the subjective test was conducted in two separate sessions. Per observer, each test session took in average 25 minutes, plus 3 minutes for the training phase. In each test session, eight omnidirectional images were evaluated. In total, 96 comparisons (6 (pairs of FoVs) × 16 (images) = 96) were made. The number of observers was 23 in the first session and 21 in the second session, aged between 22 and 42 years. The experiment was conducted using a Full HD 2D computer monitor, with a native resolution of 1920×1080 pixels. The observers were seated in front of the monitor, at a distance of approximately three times the picture height, as suggested in [73]. A subjective assessment interface, depicted in Figure 3.9, that includes viewing and scoring panels, was designed to perform the visualization of the navigation videos and collect the associated subjective scores.

After the subjective test, the resulting subjective scores were processed according to the analysis described in the next section. The processed subjective scores are available in [110].



Please grade the Video Under Evaluation.

Figure 3.9. Subjective assessment interface designed to perform the visualization of the viewport videos and to collect the associated subjective scores. The scoring panel is shown only after the second video is shown.

C. PC Subjective Scores Processing

The PC scores are usually represented in a *wining frequency* matrix. This matrix contains the number of times that a given stimulus is selected against the other stimuli involved in the comparison. To rank the stimuli from highest to lowest preference, the winning frequency matrix must be translated to a continuous scale of preferences. In the following, the outlier's detection method and the procedure to translate the winning frequency matrix to a continuous scale, are described.

Outliers can be detected by computing the transitivity satisfaction rate, Ɽ, for each observer, from his/her comparison results. The transitivity rule is violated when a circular triad is formed among three stimuli, a, b, c under evaluation. If the grade 0 (i.e., "same" quality) was not allowed, the possible circular triads would be

$$\begin{gathered}(a > b) \cap (b > c) \cap (c > a) \\ (a < b) \cap (\mathrm{b} < c) \cap (c < a)\end{gathered} \tag{3.7}$$

where $a > b$ means that a was preferred over b. Allowing the grade 0, each case above give rise to three additional circular triads which, for the first case are (for the second case it will be similar)

$$\begin{gathered}(a > b) \cap (b > c) \cap (c = a) \\ (a > b) \cap (b = c) \cap (c > a) \\ (a = b) \cap (b > c) \cap (c > a)\end{gathered} \tag{3.8}$$

where $a = b$ represents a tie between a and b. Accordingly, the total number of considered circular triads per group of three stimuli is eight. The score reliability, $Ɽ_o$, of observer o, is given by

$$Ɽ_o = 1 - \frac{\delta_o}{\eta_o} \tag{3.9}$$

where δ_o is number of detected circular triads for that observer, and η_o is the total number of possible circular triads. If $Ɽ_o \leq\ 0.9$, the observer o is considered as an outlier, as recommended in [111], and his subjective scores are not further considered.

After outlier detection, the subjective scores from the observers are represented in the wining frequency matrix and then translated to absolute quality scores using the Bradley-Terry (BT) model [114]; this is one of the most popular approaches to convert winning frequencies obtained from PC experiment, to continuous scale scores.

Considering three discrete rates ("better", "same", and "worse"), the results obtained from O observers when evaluating K stimulus, can be represented by the winning frequencies, w_{ij}, $i, j = 1,2, \dots, K$, which represents the number of times stimulus i was preferred over stimulus j, where $w_{ij} + w_{ji} = O$ and $w_{ii} = 0$. The tie cases are equally split, meaning that if the observer chooses the option "same" a score of 0.5 is given to each stimulus. The BT score for stimulus i is defined by

$$S_i = \log(q_i) \tag{3.10}$$

where q_i can be considered as the quality score for stimulus i, $q_i > 0$ and $\sum_{i=1}^{K} q_i = 1$. The probability of selecting the stimulus i against stimulus j, is estimated by

$$P_{ij} = P(i > j) = \frac{w_{ij}}{O} = \frac{q_i}{q_i + q_j}. \tag{3.11}$$

The parameters q_i can be estimated by maximizing the log-likelihood [114]

$$L(q_1, q_2, \dots, q_K) = \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} P_{ij} \log\left(\frac{q_i}{q_i + q_j}\right). \tag{3.12}$$

The 95% confidence interval (CI) for the estimated values of $\log(qi)$, can be computed as [115]

$$\left(\log q_i - 1.96\frac{\sqrt{\sigma_{ii}/\alpha}}{q_i}, \log q_i + 1.96\frac{\sqrt{\sigma_{ii}/\alpha}}{q_i}\right) \tag{3.13}$$

where $\alpha = \sum_{i<j} c_{ij}$, c_{ij} is the number of comparisons between stimuli i and j, σ_{ii} is the i^{th} diagonal element of the $(K$+1)×$(K$+1) covariance matrix

$$\Sigma = \begin{bmatrix} \Lambda & \mathbf{1} \\ \mathbf{1}' & 0 \end{bmatrix}^{-1}, \qquad \text{where } \Lambda = [\lambda_{ij}] \tag{3.14}$$

and

$$\begin{aligned} \lambda_{ii} &= \frac{1}{q_i} \sum_{j \neq i} \frac{q_j c_{ij}}{\alpha(q_i + q_j)^2}, \qquad i = 1,2, \dots, K \\ \lambda_{ij} &= \frac{-c_{ij}}{\alpha(q_i + q_j)^2}, \qquad i \neq j, \qquad i,j = 1,2, \dots, K \end{aligned} \tag{3.15}$$

It is worthy to note that the confidence interval given by (4.13) corresponds to the BT fitting model and does not represent the observers' confidence. Also, since the BT model estimates the scores for each stimulus from the distance between pairs of stimuli, the estimation error is propagated; thus, a larger confidence interval will result for stimuli with scores farther away from the stimulus with the highest computed score.

In this work, the outliers were detected for each subjective test session separately, and by applying (3.7) to (3.9); two outliers were detected in each test session, and their subjective scores were not further considered. As detailed in Section 3.3.2.A, for each considered omnidirectional image six navigation videos were produced, each video having a distinct F_h, $F_h \in \{75°, 90°, 100°, 110°, 120°, 135°\}$. Accordingly, a 6×6 wining frequency matrix was built for each omnidirectional image, containing the wining frequencies for the considered paired comparisons. The quality score of each navigation video was then estimated by applying (4.10), (3.11), and (3.12). Finally, the 95% confidence interval was computed using (3.13).

D. Subjective Test Results and Analysis

Table 3.2 presents the preferences probabilities between compared FoVs, computed by (3.12) and averaged for the corresponding stimuli; Figure 3.10 and Figure 3.11 depict, respectively, the resulting BT scores and corresponding 95% CI, obtained per image.

Figure 3.10 shows that, in general, when the FoV varies from 75º to 110º, the experienced quality increases. This is consistent with the results from other works (e.g. [11][12][106]), that showed an enhancement of the user experience when the FoV increases up to a certain value. On the other hand, the quality scores decrease for FoV values above 110º, since the geometric distortions resulting from the sphere to plane projection became too much evident and

Table 3.2. Preference Probabilities for the compared FoVs.

FoV	75º	90º	100º	110º	120º	135º
75º	-	0.09	-	-	-	-
90º	**0.91**	-	0.18	-	-	-
100º	-	**0.82**	-	0.35	-	-
110º	-	-	**0.65**	-	**0.66**	-
120º	-	-	-	0.34	-	**0.87**
135º	-	-	-	-	0.13	-

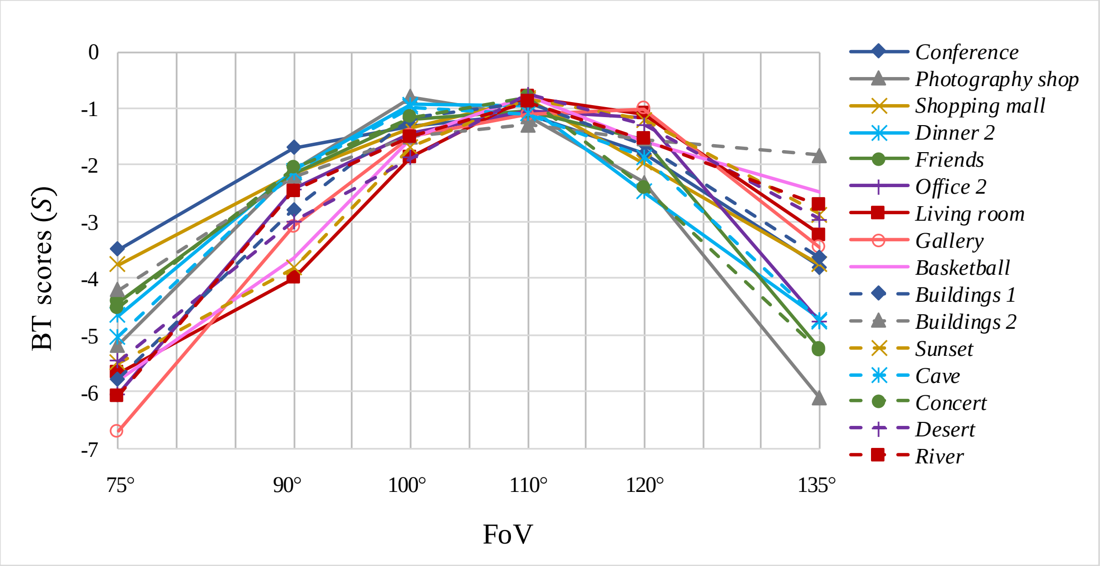


Figure 3.10. BT scores vs. FoV for each considered omnidirectional image.

annoying. This is objectively shown by Figure 4.2 of Chapter 4, that depicts the evolution of some stretching distortion measures, as a function of the FoV.

The FoV of 110º was selected as the optimum FoV for most of the considered images; the exceptions to this are the images *Photography shop*, *Dinner 2*, and *Cave*, for which the preferred FoV was 100º (although with a marginal difference compared to the FoV of 110º). The images *Photography shop* and *Dinner 2* have human faces quite close to the camera, and their geometric distortion becomes rather severe for the FoV of 110º. Regarding the image *Cave*, it shows the interior of a cave structure that has a round shape, whose distortion is quite visible for a FoV of 110º.

Although the BT scores do not show the same increasing or decreasing rate for the different types of image content, no statistically significant difference was found, not even between indoor and outdoor images for which a clear separation was rather expected.

As expected, Figure 3.11 shows that the CI increases for stimuli with FoVs farther away from the preferred FoV (110º), due to error propagation. This could be reduced with a more complete winning frequency matrix, requiring more pairs of compared FoVs during the subjective tests, at the cost of a higher test duration.

In summary, the subjective test results suggest that the best trade-off between user immersive experience and geometric distortions perception is achieved for a FoV close to 110º, regardless of the image content. However, the optimum FoV could have a different value if other

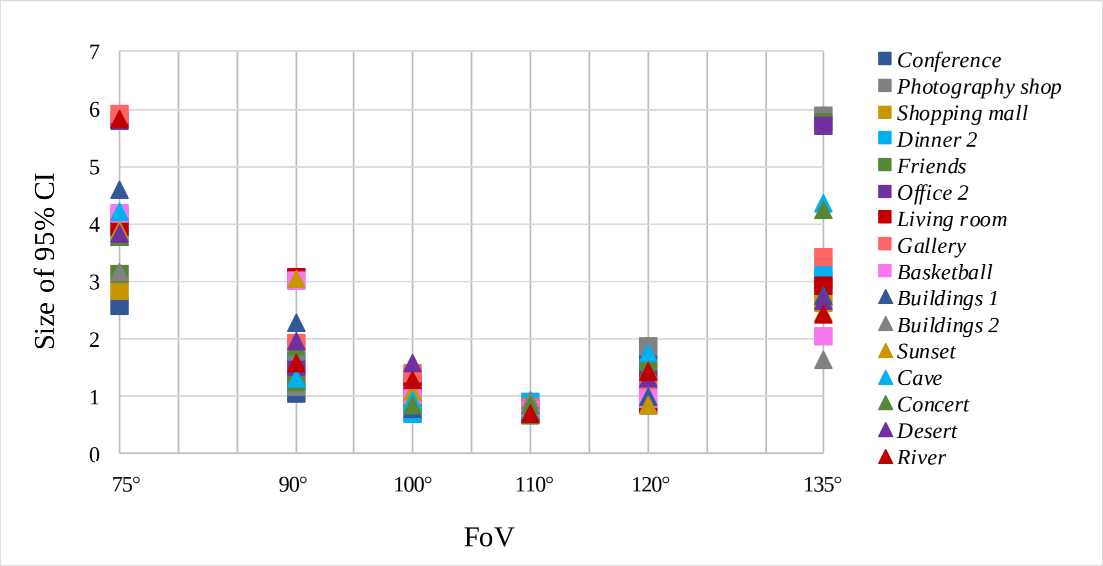


Figure 3.11. The 95% CI for each considered omnidirectional image and FoV.

projection than the rectilinear one (e.g., GPP with $d > 0$ or Pannini) was used for the viewport rendering. For example, as shown in Figure 3.6, for the image *Buildings 1*, the FoVs of 75° and 90° obtained a similar scores for $pr_1(d = 0.25)$, but different scores (the score for 75° is much higher than 90°) for $pr_4(d = 1)$.

3.4 Final Remarks

The first objective of this chapter was to subjectively evaluate the perceptual impact of the geometric distortions that result from the viewport rendering. To achieve this objective, a subjective evaluation test campaign of GPP viewports was conducted, showing that the projection type (d value), the considered FoV, and the image content characteristics, are the three main factors that influence the distortion strength, and its perception.

The second objective of this chapter was to subjectively evaluate the FoV impact on the perceived geometric distortion, to determine the FoV that presents the best trade-off between user's immersive experience and the perceived degradations due to geometric distortions (when the rectilinear projection is used), and also to evaluate its dependency on the image content. This objective was also achieved through a subjective evaluation test campaign. The experimental results show that the best trade-off between user immersive experience and geometric distortions perception is achieved for a FoV close to 110°, regardless of the image content.

All the subjective tests were conducted using 2D displays. If HMDs were used, other factors such as lens distortions, the content magnification, and the effects of peripheral vision, may condition the perceptibility of the geometric distortions [67][116]; therefore, the main conclusions of this section should be validated (and eventually updated) for HMDs, by conducting similar subjective assessment tests with these devices.

The work conducted for the first objective of this chapter was included on three published conferences and one journal paper, referred in the first four rows of Table 3.3. As mentioned earlier, the GPP subjective test was conducted in several sessions, progressively over time. Therefore, the last column of Table 3.3 presents the omnidirectional images considered in each paper. The work conducted for the second objective of this chapter was included in a published conference paper, referred in the last row of Table 3.3.

Table 3.3. Publications related to this chapter.

Paper	**Type**	**Used omnidirectional images**
F. Jabar, J. Ascenso, and M.P. Queluz, "Perceptual Analysis of Perspective Projection for Viewport Rendering in 360° Images," Proc. of the IEEE International Symposium on Multimedia, Taichung, Taiwan, Dec. 2017.	Conference	Images *Conference*, *Buildings 1*, *Photography shop* and *Pole vault*, of Figure 3.1
F. Jabar, M.P. Queluz, and J. Ascenso, "Objective Assessment of Line Distortions in Viewport Rendering of 360° Images," Proc. of the IEEE International Conference on Artificial Intelligence and Virtual Reality, Taichung, Taiwan, Dec. 2018.	Conference	Images *Conference*, *Buildings 1*, *Photography shop*, *Pole vault*, *Museum*, and *Buildings 2*, of Figure 3.1
F. Jabar, J. Ascenso, and M.P. Queluz, "Content-Aware Perspective Projection Optimization for Viewport Rendering of 360° Images," Proc. of the IEEE International Conference on Multimedia and Expo, Shanghai, China, Jul. 2019.	Conference	All the images of Figure 3.1
F. Jabar, J. Ascenso, and M.P. Queluz, "Objective Assessment of Perceived Geometric Distortions in Viewport Rendering of 360° Images," IEEE J. Sel. Top. Signal Process., vol. 14, no. 1, pp. 49–63, Jan. 2020.	Journal	All the images of Figure 3.1
F. Jabar, J. Ascenso, and M.P. Queluz, "Field-of-View Effect on the Perceived Quality of Omnidirectional Images," Proc. of the IEEE International Conference on Multimedia & Expo Workshops, Athlone, Ireland, Jul. 2020.	Conference	All the images of Figure 3.1

Chapter 4

Objective Assessment and Optimization of the General Perspective Projection

4.1 Introduction

As mentioned in Chapter 2, any sphere to a plane projection results in stretching and/or bending distortions on the resulting planar image, whose strength and subjective impact depends on the image content. Accordingly, selecting a proper projection and its parameters may have an important role on the user's QoE. In the general perspective projection (GPP), the amount and type of geometric distortions can be controlled by changing its only projection parameter – the projection center – allowing to obtain the often used rectilinear and stereographic projections, and other intermediate projections. However, this should be done with an automatic procedure where the projection center is optimized (in a perceptual sense) based on the viewport content.

Optimizing the projection center, d, based on the viewport content requires the availability of a content-aware objective quality metric, that automatically estimates the perceived geometric distortion, after rendering. Therefore, the main objectives of this chapter are:

- Develop content-dependent geometric distortion metrics to characterize and measure the two main types of geometric distortion that occur during the viewport rendering of omnidirectional images, notably stretching of image regions and bending of straight lines.
- Develop a content-aware objective quality metric that automatically assesses the geometric distortions in the viewport image and predicts its subjective quality, when the GPP is used for the rendering of omnidirectional images.
- Develop procedures to optimize the GPP when used in the viewport rendering process, aiming to minimize the perceived geometric distortions, by adapting the projection center to the viewport content.

The rest of this chapter is organized as follows. Sections 4.2 and 4.3 propose several stretching and bending distortions metrics, respectively. Section 4.4 presents a novel content-aware objective quality metric, that integrates some of the individual distortion metrics previously described; it allows to assess the perceptual impact of the geometric distortions, when the GPP is used for viewport rendering. Section 4.5 describes a useful application of the proposed quality metric - the optimization of the GPP parameter (d), according to the viewport content - allowing to obtained viewports with enhanced visual quality. Section 4.6 finalizes this chapter by presenting some final remarks.

4.2 Stretching Distortion Metrics

In this section, a set of potential stretching distortion metrics, based on the Tissot ellipse of distortion (or indicatrix) described in Chapter 2, are proposed. First, the details about the Tissot indicatrix parameters computation are presented; these parameters are then used to characterize the geometric distortions induced by the GPP on the rendered viewport. Afterwards, three viewports (or global) stretching measures, are suggested.

4.2.1 GPP Stretching Characterization

The semi-major, $\hat{a}$, and semi-minor, $\hat{b}$, axis of the Tissot ellipse centered at position (x_p, y_p) of the projection plane, for the forward projection, $(x_p, y_p) = Proj(\phi, \theta)$, where (θ, ϕ) are the spherical coordinates of a point on the sphere of unit radius, are given by [46]

$$\hat{a} = \frac{\acute{a} + \acute{b}}{2} \tag{4.1}$$

$$\hat{b} = \frac{\acute{a} - \acute{b}}{2} \tag{4.2}$$

where $\acute{a}$ and $\acute{b}$ are auxiliary terms given by

$$\acute{a} = \sqrt{h^2 + k^2 + 2hk \sin \acute{\theta}} \tag{4.3}$$

$$\acute{b} = \sqrt{h^2 + k^2 - 2hk \sin \acute{\theta}}. \tag{4.4}$$

In (4.3) and (4.4), h and k are, respectively, the vertical and horizontal stretching factors, and $\acute{\theta}$ is the angular deformation, representing the angle between the projected meridian and parallel at (x_p, y_p), given by

$$h = \sqrt{\left(\frac{\partial x_p}{\partial \theta}\right)^2 + \left(\frac{\partial y_p}{\partial \theta}\right)^2} \tag{4.5}$$

$$k = \frac{1}{\cos \theta} \sqrt{\left(\frac{\partial x_p}{\partial \phi}\right)^2 + \left(\frac{\partial y_p}{\partial \phi}\right)^2} \tag{4.6}$$

$$\sin \acute{\theta} = \frac{1}{hk \cos \theta}\left[\left(\frac{\partial y_p}{\partial \theta} \times \frac{\partial x_p}{\partial \phi}\right) - \left(\frac{\partial x_p}{\partial \theta} \times \frac{\partial y_p}{\partial \phi}\right)\right]. \tag{4.7}$$

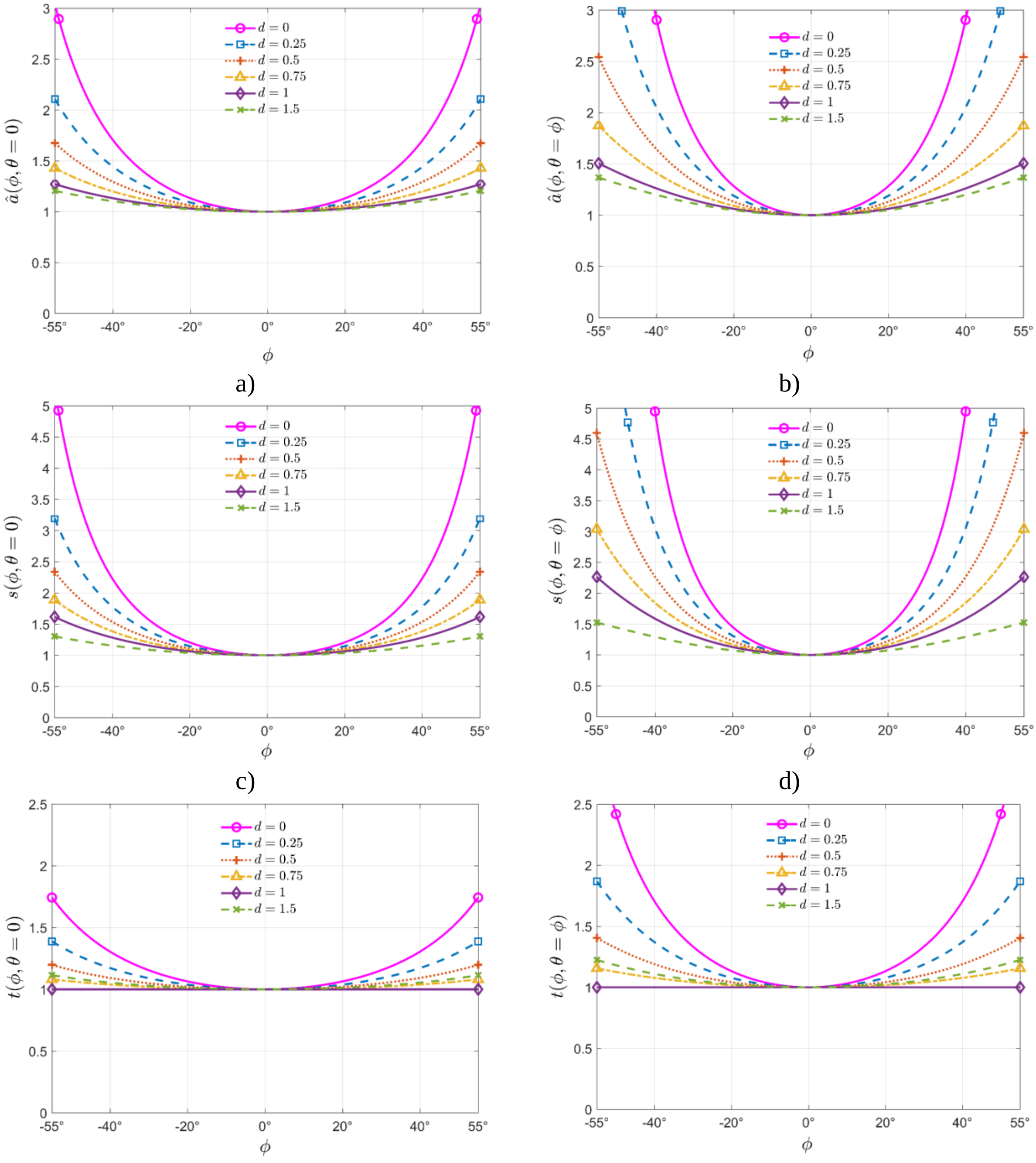
The local shape distortion, t, maximum angle deviation, ω, and amount of inflation or deflation in the area, s, can be computed by (2.49) to (2.51).

To obtain the Tissot indicatrix for the GPP, the partial derivatives required by (4.5) to (4.7) need to be obtained. Considering the GPP forward projection equations, $(x_p, y_p) = Proj(\phi, \theta, d)$, given by (2.24) and (2.25), results on the partial derivatives presented in Table 4.1. It is worthy to note that since the local Tissot parameters are content-independent, they can be computed for any sphere to plane projection having closed-form projection equations.

Figure 4.1 presents the resulting $\hat{a}$, the local area inflation, s, and the local shape distortion, t, as a function of the longitude (ϕ), for different values of d, and along a horizontal $(\theta = 0)$ and diagonal $(\theta = \phi)$ viewport direction.

Table 4.1. Partial derivatives for GPP forward projection equations.

$\dfrac{\partial x_p}{\partial \theta} = -\dfrac{d(1+d)\sin\theta\sin\phi}{(\cos\theta\cos\phi + d)^2}$	(4.8)	$\dfrac{\partial x_p}{\partial \phi} = \dfrac{(1+d)(\cos^2\theta + d\cos\theta\cos\phi)}{(\cos\theta\cos\phi + d)^2}$	(4.9)
$\dfrac{\partial y_p}{\partial \theta} = \dfrac{(1+d)(d\cos\theta + \cos\phi)}{(\cos\theta\cos\phi + d)^2}$	(4.10)	$\dfrac{\partial y_p}{\partial \phi} = \dfrac{(1+d)\sin\theta\cos\theta\sin\phi}{(\cos\theta\cos\phi + d)^2}$	(4.11)



a)

b)

c)

d)

e)

f)

Figure 4.1. Plot of $\hat{a}$ (ellipse semi-major axis), s (area factor), and t (shape distortion) for: horizontal direction, in a), c), e); diagonal direction, in b), d), f).

In Figure 4.1a), and for any considered projection, it can be observed an exponential growth in the value of $\hat{a}$ with ϕ; this stretching increases significantly along the diagonal direction (*cf.* Figure 4.1b). Similar behaviour is shown by s and t, although less remarkably by t; in particular, for $d = 1$ results that $t = 1$; for $d > 1$, the value of t increases again.

Accordingly, the following main conclusions can be drawn:

- **Semi-major ellipsis axis, $\hat{a}$** - for $d \in [0,1.5]$, the image is stretched towards the viewport borders and this stretching decreases as d increases. This can be observed in Figure 4.1a)-b), where $\hat{a}$ increases significantly towards the image borders for $d = 0$ (rectilinear projection) while this effect is weaker for larger values of d.
- **Area factor, s** - for $d \in [0,1.5]$, relative areas are not preserved. As can be seen in Figure 4.1c)-d), for $d = 0$ the area distortion increases significantly towards the image borders and decreases when d increases.
- **Shape distortion, t** - for $d \in [0,1.5]$, the shape and angle distortions decrease when d increases towards 1, and increase again when $d > 1$. For $d = 1$ (stereographic projection) the projection is conformal, so shapes are locally preserved.

Figure 4.2 depicts the resulting t as a function of the longitude (ϕ), using $d = 0$ and for different values of HFoV, and along a horizontal ($\theta = 0$) direction. When the HFoV increases from 110° to 150°, the value of t increases exponentially. This shows that the stretching of image regions increases significantly towards the image borders, justifying why a large FoV was not selected as the optimum value in the subjective test of FoV impact, presented in Chapter 3 (Section 3.3).

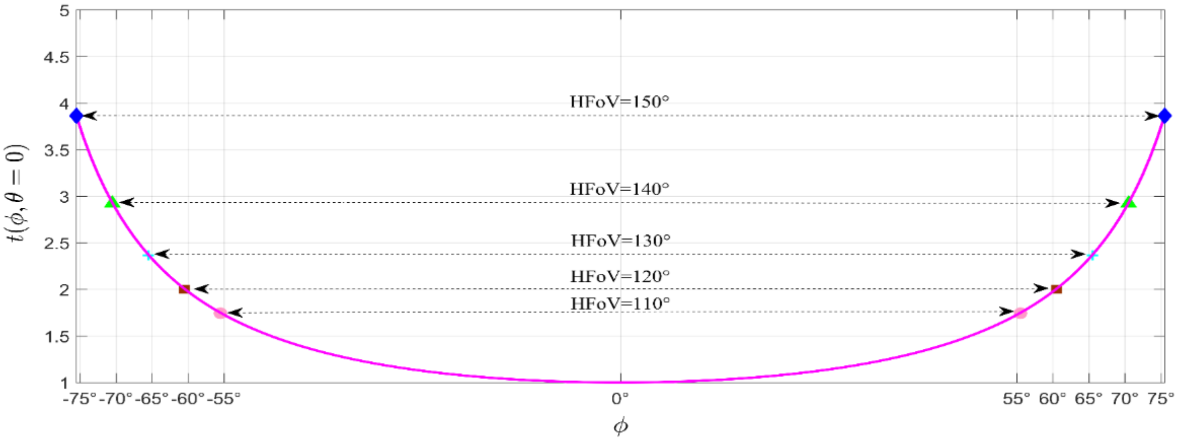


Figure 4.2. Plot of t (shape distortion) for $\theta = 0$ and different HFoV values.

4.2.2 Stretching Metric Computation

After obtaining the Tissot indicatrix parameters, $\hat{a}$ and $\hat{b}$, these parameters are then used to compute three measures of local stretching, namely, angle, scale, and area distortions. Finally, the local measures are aggregated to obtain three global stretching measures, that represent the whole viewport distortion. Each one of these steps is detailed herein:

- **Local Tissot Parameters** - The specified FoV for the viewport rendering defines a region on the sphere corresponding to $\phi \in [-F_h/2, F_h/2]$ and $\theta \in [-F_v/2, F_v/2]$, where F_h and F_v are, respectively, the horizontal and vertical FoVs. This region is then uniformly sampled, with intervals $\Delta\phi, \Delta\theta$ (set to 0.05 degrees in this Thesis). For the sampled point i, with spherical coordinates (ϕ_i, θ_i), the corresponding Tissot parameters, $\hat{a}_i$ and $\hat{b}_i$ are then obtained as described in Section 4.2.1.

- **Local Stretching Measures** - These measures account for the local angle, scale, and area distortions – denoted as $dangle$, $dscale$, and $darea$ – and are formally described by (4.12) to (4.14), respectively, as suggested in [117]; $\cos\theta_i$ reflects the decrease in the area comprised by $\Delta\phi, \Delta\theta$, as θ varies from 0 to ± 90 degrees.

$$dangle_i = 2\sin^{-1}\left(\frac{|\hat{a}_i - \hat{b}_i|}{\hat{a}_i + \hat{b}_i}\right)\cos\theta_i \tag{4.12}$$

$$dscale_i = \left(\frac{\hat{a}_i + \hat{b}_i}{2} - 1\right)\cos\theta_i \tag{4.13}$$

$$darea_i = \left(\hat{a}_i \times \hat{b}_i - 1\right)\cos\theta_i \ . \tag{4.14}$$

  Since the local Tissot parameters and stretching measures are content-independent, it can be computed beforehand for any considered FoV and d.

- **Global Stretching Measures** - These measures represent the viewport angle, scale, and area distortions - denoted by G_{dangle}, G_{dscale}, G_{darea}, respectively - and are formally described by

$$G_{dangle} = \frac{1}{\grave{s}}\sum_i dangle_i \tag{4.15}$$

$$G_{dscale} = \frac{1}{\grave{s}}\sum_i dscale_i \tag{4.16}$$

$$G_{darea} = \frac{1}{\grave{s}}\sum_i darea_i \tag{4.17}$$

  where $\grave{s}$ is given by

$$\grave{s} = \sum_i \cos\theta_i \ . \tag{4.18}$$

The aforementioned stretching measures are content-independent, and thus are not enough to explain the dependency of the geometric distortion visibility, on the image content (as seen on the subjective tests). To make these measures content-dependent, they are combined with viewport saliency weights, which gives more importance to the viewport regions that attract more the user attention (e.g., objects, human faces) and thus are more sensitive to geometric distortions. This step will be detailed in Section 4.4.1.B. Also, the evaluation of the proposed metrics, and the selection of the most useful ones - from the point of view of viewport quality estimation - is conducted in Section 4.4.2.

Although the Tissot indicatrix, and the proposed measures based on it, may explain some of the distortions that are visible on the rendered viewport, they are not able to explain, nor to measure, the bending of straight lines. However, this distortion plays a significant role on the user's QoE; for the particular case of the GPP, lines bending becomes quite visible and annoying as the projection center, d, approaches 1. This issue is considered on the next section, where a set of potential bending distortion metrics are proposed.

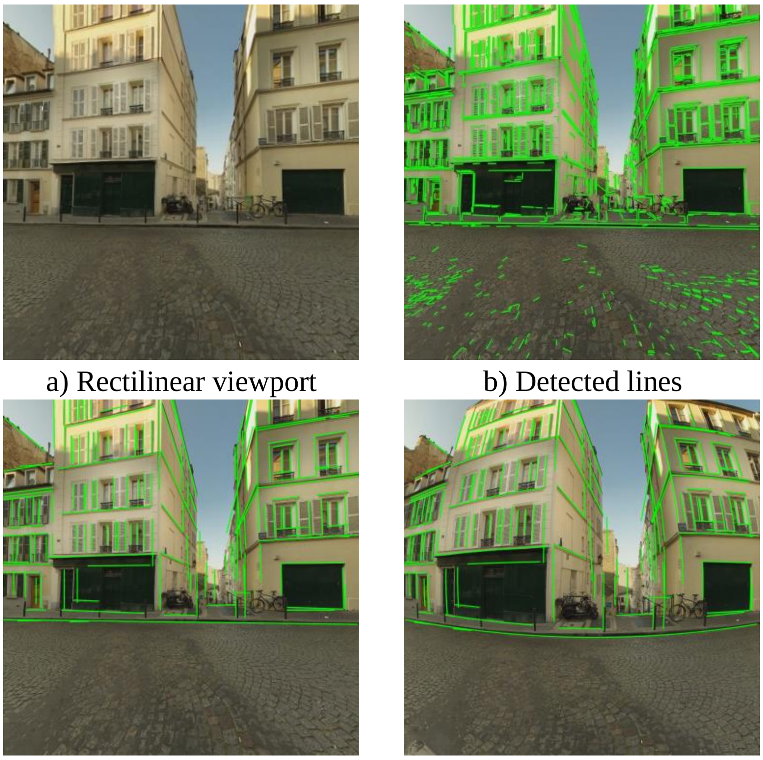
4.3 Bending Distortion Metrics

This section proposes a set of potential line distortion metrics, for viewport rendering of omnidirectional images. First, straight lines are detected and merged, and short lines are filtered out. After, for each remaining line, its distortion is computed. Finally, the individual distortion values are aggregated in a single value, that represents the viewport (or global) line distortion.

4.3.1 Line Detection and Merging

To compute the bending measures, straight lines are detected in the viewport obtained with the rectilinear projection, since this projection keeps the straightness of the lines. Each detected line is represented as a set of pixels. In general, the bending of short lines has a lower perceptual impact; therefore, lines that are close enough, thus likely belonging to the same image contour, are merged with some criteria; also, short lines that remain after the merging procedure, are removed. Each step is described next:

- **Straight Line Detection** - Straight lines are detected using a line segment detector (EDLines) proposed in [118]. EDLines is an algorithm that aims to extract straight lines in the image based on the image gradient, with a validation step that allows to reduce the number of false detections. It is based on edge segment chains, runs faster than other line detection algorithms, and does not require any parameter tuning. Figure 4.3b) illustrates an example of the detected lines, for the viewport represented in Figure 4.3a); as can be seen, several broken lines may result from the line detection step.
- **Line Merging** - After line detection, neighbour lines are connected, based on their orientations and locations. The following steps are applied:
    1) **Angular clustering** - Lines with a similar orientation (parallel or quasi parallel lines) are clustered together. A cluster represents a group of lines and is defined by some angular interval. According to their orientation, all lines are assigned to clusters which are precisely defined by their angular interval limits. The size of the uniform intervals is controlled by A_c, in degrees.
    2) **Collinear clustering** - For each cluster of parallel lines obtained in the previous step, lines are aggregated in sub-clusters of collinear lines (lines having their end points on the same straight line), according to their y-axis (or x-axis) intercept value difference. This process is controlled by the parameter C_c, in length units, which corresponds to the maximum difference between intercept values for the lines to be considered as collinear.
    3) **Merging** - Lines considered as collinear are merged if they are close to each other, i.e., if the start or end location of a line is close to the start or end location of another line. This means that the distance between lines is defined by their extremities. This process is controlled by the parameter L_m, in length units.
- **Line Filtering** - Since short lines do not have a relevant perceptual impact (as its bending is barely perceived), only the lines longer than a pre-defined threshold in length units, L_f, are kept. By increasing the value of L_f, more lines will be filtered out.



a) Rectilinear viewport b) Detected lines

c) Merged lines d) Projected lines

Figure 4.3. a) Rectilinear viewport with a square FoV of $\mathbf{110°}$; b) Detected lines using EDLines; c) Merged lines with $\boldsymbol{A_c = 3}$, $\boldsymbol{C_c = 10}$, $\boldsymbol{L_m = 25}$ and $\boldsymbol{L_f = 45}$; d) Projected lines with $\boldsymbol{d = 1}$.

The line merging parameters were determined by visual inspection of the results obtained for several viewports, rendered from different omnidirectional images. Following this procedure, A_c, C_c, L_m and L_f were set, respectively, to 3°, 0.033, 0.083, 0.150. Figure 4.3c) shows the resulting lines after applying the described procedure to the lines of Figure 4.3b).

To get the lines on the viewport resulting from a given projection, $(x_p, y_p) = Proj(\phi, \theta)$, every pixel position of the resulting lines is first backward projected to the spherical representation, using the rectilinear projection, and then forward projected to the viewport plane, using the intended projection, $(x_p, y_p) = Proj(\phi, \theta)$. As an example, Figure 4.3d) shows the resulting lines for GPP with $d = 1$, after applying the backward (with $d = 0$) and the forward (with $d = 1$) projections, to the lines of Figure 4.3c).

4.3.2 Bending Metric Computation

The bending metrics are computed by measuring two characteristics of each projected straight line - the line curvature and the line inclination - for the projection under consideration. Then, a pooling procedure is performed to fuse all measures into a single metric value, representing the whole (i.e., global) viewport line bending. The following steps are performed:

- **Line Distortion Computation** - Depending on the used projection, the projected lines can be distorted in two different ways: *i)* bending, and *ii)* direction change (or inclination). As an example, Figure 4.3d) shows that the lines of the windows located on the right part are unnaturally deformed and have odd angles. Consider that $L = \{\ell_1, \ell_2, \ell_3, \dots, \ell_T\}$ is the set of projected lines for a given viewport, where ℓ_i is a projected line indexed by i, and T is the total number of projected lines. In this context, two line distortion measures are proposed:

1) **Line curvature measure** - Consider a projected line, ℓ_i (represented in green in Figure 4.4), and the straight line joining the line endpoints, s_i (represented in blue in Figure 4.4), whose length is $\hat{L}_i$. Two measures are then defined: the line curvature LC_i, defined as the maximum distance between ℓ_i and s_i (along the perpendicular direction to s_i), and the normalized line curvature NLC_i, defined by (4.19)

$$NLC_i = \frac{\hat{L}_i}{LC_i + \hat{L}_i}. \tag{4.19}$$

which is rather similar to a metric proposed in [36]. As the projected line tends to a straight line, LC_i tends to 0, and NLC_i tends to 1; otherwise, LC_i has a value greater than 0, and NLC_i has a value lower than 1.

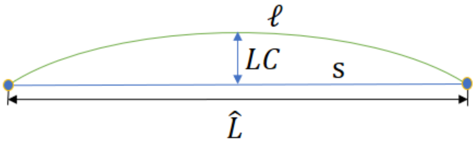


Figure 4.4. Illustration of the line curvature measure.

2) **Line inclination measure** - Now, consider a new line, o_i (shown in red in Figure 4.5) with the same angle as the straight line obtained with the rectilinear projection ($d = 0$), and starting at one endpoint of the corresponding projected line. Again, two measures are defined: the line inclination LI_i, defined as the maximum distance between o_i and s_i (and along the perpendicular direction to s_i), and the normalized line inclination NLI_i, defined by (4.20)

$$NLI_i = \frac{\hat{L}_i}{LI_i + \hat{L}_i}. \tag{4.20}$$

As the projected line inclination tends to the original straight line inclination, LI_i tends to 0, and NLI_i tends to 1; otherwise, LI_i has a value greater than 0, and NLI_i has a value lower than 1.

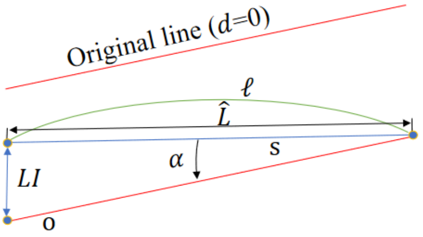


Figure 4.5. Illustration of the line inclination measure.

3) **Line distortion combination** - In this case, the previously defined line curvature and inclination measures are combined in a single distortion measure, according to

$$LMC_i = LC_i + LI_i \tag{4.21}$$

$$NLMC_i = \frac{\hat{L}_i}{LMC_i + \hat{L}_i}. \tag{4.22}$$

Figure 4.6 shows five straight lines (red color) obtained with the rectilinear $(d = 0)$ projection, the corresponding projected lines (green color) obtained with GPP projection $(d = 1)$ and all the measure values for each line. All line distortion measures vary as expected from the perceived distortions, having higher distinctive values for the LC and LI metrics.

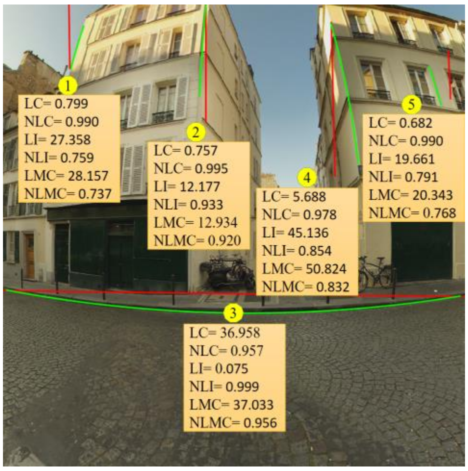


Figure 4.6. Example of line distortion values for a few projected lines (scaled by a scale factor of 300).

- **Line Pooling** - To fuse, in one single value, the line distortion values computed for all projected lines, several pooling functions (P_s^l) were considered, which are listed in Table 4.2. In this table, K is a vector containing one of the distortion measures - *LC*, *NLC*, *LI* or *NLI* - for all projected lines; K_p is a vector containing all the elements of K higher or equal to ρ, where ρ corresponds to the p percentile of K (i.e., p% of the elements of K are lower than ρ); $\overline{K_p}$ is a vector containing all the elements of K lower or equal to $\bar{\rho}$, where $\bar{\rho}$ corresponds to the $(100 - p)$ percentile of K. Poolings P_1^l and P_5^l assume that the subjective impact of the line distortion increases with the number of lines, while pooling P_2^l, P_6^l, and P_7^l, consider that the impact varies with the average line distortion; poolings P_3^l and P_4^l presume that the perceptual impact is mainly influenced by the most distorted lines. The reason for the percentile (p%) is to exclude the lines with low distortion values (e.g., the distortion for lines at the viewport center is low and may not be visible). The best value for p was 90%, obtained as described in Annex A.

Table 4.2. Line pooling strategies.

P_s^l	Distortion Measure
$P_1^l = \text{Sum}(K)$	*LC, LI, LMC*
$P_2^l = \text{Average}(K)$	*LC, NLC, LI, NLI, LMC, NLMC*
$P_3^l = \text{Max}(K)$	*LC, LI, LMC*
$P_4^l = \text{Min}(K)$	*NLC, NLI, NLMC*
$P_5^l = \text{Sum}(K_p)$	*LC, LI, LMC*
$P_6^l = \text{Average}(K_p)$	*LC, LI, LMC*
$P_7^l = \text{Average}(\overline{K_p})$	*NLC, NLI, NLMC*

It is worthy to mention that the normalized measures, NLC and NLI, vary inversely with the line distortion values, resulting that pooling P_1^l, P_3^l, P_5^l and P_6^l cannot be applied to them. In fact, these pooling functions sum the vector elements or consider the highest values of the vector (which correspond to the lowest distortions for these measures). Pooling P_4^l and P_7^l were specifically designed for these cases.

Considering the possible combinations of pooling strategies with distortion measures, a total of 24 potential line distortion metrics are obtained.

In [119], and besides the proposal of the line bending metrics and pooling functions presented above (with exception for *LMC* and *NLMC*, which were introduced later), a preliminary SVM-based model for objectively assessing the GPP was also proposed (which is included in Annex A). This model uses only the described line bending metrics and pooling functions and is able to classify a viewport obtained with a given GPP projection in one of the three quality groups referred to in Chapter 3, Section 3.2.4 (i.e., G1, G2 or G3), with an accuracy close to 91%. Accordingly, and besides validating the proposed line bending metrics, it allows to decide if it is worth to use a projection other than the conventional rectilinear one, for the considered viewport. However, for a reliable estimation of the viewport quality scores, a more complete model, that also incorporates an objective measure of the stretching distortion - besides some of the described line distortion measures - is required, as described in the next section.

4.4 Content-Aware Objective Quality Metric

This section describes the proposed content-aware objective metric, for assessing the perceived geometric distortions in viewport rendering of omnidirectional images, under GPP projection. The metric takes into account the image content, namely straight lines and salient regions, to compute two sets of geometric distortion measures (or features) described previously, that quantify the dominant distortion types - stretching of the objects and bending of straight lines - and predicts the CMOS value of a viewport rendered with the GPP.

4.4.1 Methodology

Figure 4.7 depicts the block diagram of the proposed metric. For a given input equirectangular image (*ERI*) image, viewport horizontal FoV, F_h, spatial resolution (W_{vp}, H_{vp}), and viewing direction (ϕ_{VD}, θ_{VD}), two viewports are obtained with the GPP: the reference one, rendered with the rectilinear projection, and the viewport under evaluation, rendered with $d = d_q$. After computing the viewport saliency map, stretching and bending features are extracted, which are then fed to a Support Vector Regression (SVR) model that outputs the predicted CMOS value for $d = d_q$.

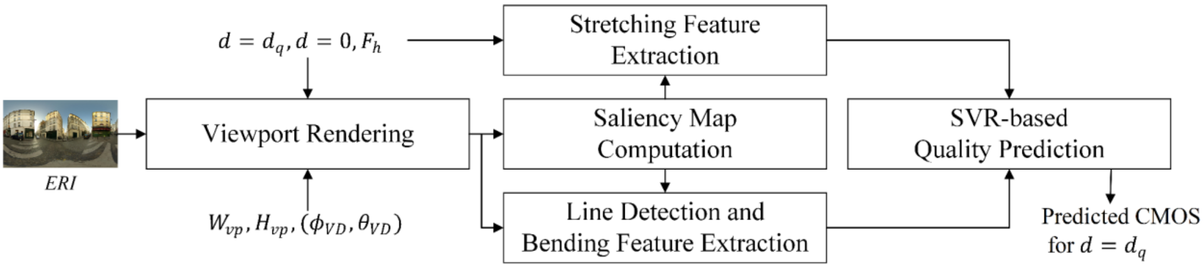

$d = d_q, d = 0, F_h$
Stretching Feature Extraction
ERI
Viewport Rendering
Saliency Map Computation
SVR-based Quality Prediction
$W_{vp}, H_{vp}, (\phi_{VD}, \theta_{VD})$
Line Detection and Bending Feature Extraction
Predicted CMOS for $d = d_q$

Figure 4.7. The block diagram of the proposed quality assessment metric.

The following sections detail the main steps of the metric.

A. *Saliency Map Computation*

Saliency detection models aim at identifying the image regions where the human attention is more focused on. To detect the salient regions in the viewport, the ML-Net saliency detection method [120] was used, since it is computationally efficient and has a good performance. In this method, the saliency map is computed based on features extracted at different levels of a Deep Convolutional Neural Network (DCNN). As in many saliency detection models, the resulting saliency map has the highest values at the center of salient regions, which progressively decrease towards the region borders (as shown in Figure 4.8b). However, the straight lines are typically located on the regions borders or objects contours. Thus, to increase the saliency values over these important structures, a power-law transformation was applied to the whole saliency map, according to

$$D(m,n) = \lambda \times S^{\gamma}(m,n) \tag{4.23}$$

where $S(m,n)$ and $D(m,n)$ are, respectively, the input and output saliency values (in the range [0,255]) at the viewport pixel position (m,n), and λ and γ are positive values that condition the transformation behavior.

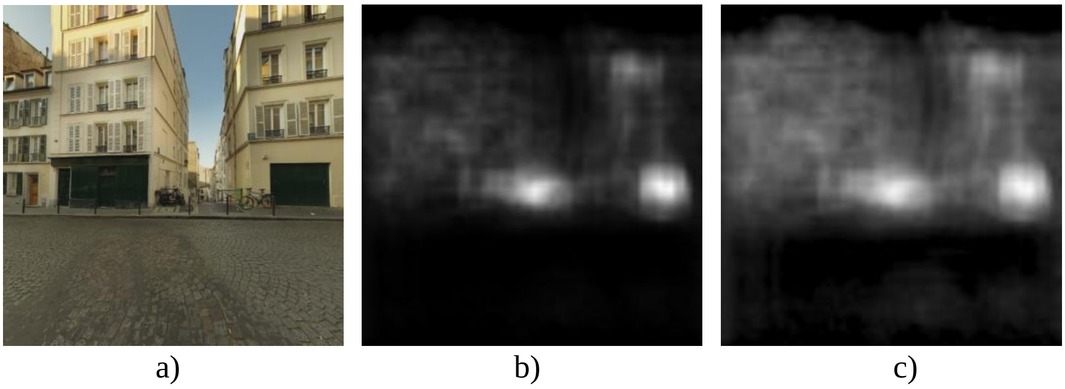


Figure 4.8. a) Viewport obtained with GPP, $d = 0$ and a square FoV of $110°$; b) Saliency map obtained with the ML-Net method; c) Saliency map after applying the power-law transformation, with $\lambda = 1$ and $\gamma = 0.8$.

For $\gamma < 1$, a narrow range of the lowest input values is mapped into a wider range of output values. For the results presented in this Thesis, $\gamma = 0.8$ and $\lambda = 1$, which slowly expands the saliency from the regions center, towards their borders. These values were determined by visual inspection of the resulting viewport saliency maps, and by their impact in the final metric results. After applying the power-law transformation, the resulting saliency maps are normalized to the range [0,1]; each pixel value is then considered as a saliency weight. Figure 4.8b) shows the saliency map resulting from the ML-Net method, for the viewport depicted in Figure 4.8a); Figure 4.8c) shows the obtained map after power-law transformation.

B. *Stretching Feature Extraction*

The stretching features are based on the global Tissot distortion measures described in Section 4.2.2, namely angle, scale, and area distortion. However, to make it content-dependent, these measures are weighted by the viewport saliency scores, according to

$$G_{dangle}^{w} = \frac{1}{s^{w}} \sum_{i} dangle_i \times w_i \tag{4.24}$$

$$G_{dscale}^{w} = \frac{1}{s^{w}} \sum_{i} dscale_i \times w_i \tag{4.25}$$

$$G_{darea}^{w} = \frac{1}{s^{w}} \sum_{i} darea_i \times w_i \, , \tag{4.26}$$

where $dangle_i$, $dscale_i$, $darea_i$ are, respectively, the local angle, scale and area distortion at (ϕ_i, θ_i), given by (4.12) to (4.14), w_i is the saliency weight at (ϕ_i, θ_i), and s^{w} is given by

$$s^{w} = \sum_{i} \cos\theta_i \times w_i \, . \tag{4.27}$$

Since the saliency was obtained on the viewport plane, with pixel coordinates $(m$,$n)$, to get the saliency weigh at (ϕ_i, θ_i) these coordinates are first projected on the viewport plane, using (2.24) and (2.25), and the corresponding viewport position is then obtained by (2.39) to (2.42). This final step integrates the saliency maps into the local stretching measures, thus resulting in a global measure that considers the content characteristics; this brings an added value compared to the content-independent Tissot measures.

Figure 4.9c) presents the global stretching measures with respect to different projection centers, resulting for the two viewport images depicted in Figure 4.9a) and Figure 4.9b). As shown, all measures have the highest value for $d = 0$ (pr_0) and decrease when the projection center tends to 1 (pr_4). But more importantly, Figure 4.9c) shows that the proposed measures, with saliency, allows to discriminate both images according to the perceived stretching distortion; in fact, the distortion values represented in solid lines (corresponding to Figure 4.9a), which have a higher amount of perceived stretching, as in the boy on the left, and on the table) are higher than the ones represented in dashed lines, corresponding to Figure 4.9b). Without the use of the saliency, the stretching measures would be the same for both images.

Since the CMOS values obtained experimentally have, as a reference, the viewport rendered with d=0, the three stretching features for a query projection center, d_q, are defined as

$$SF_{dangle} = G_{dangle}^{w}(0) - G_{dangle}^{w}(d_q) \tag{4.28}$$

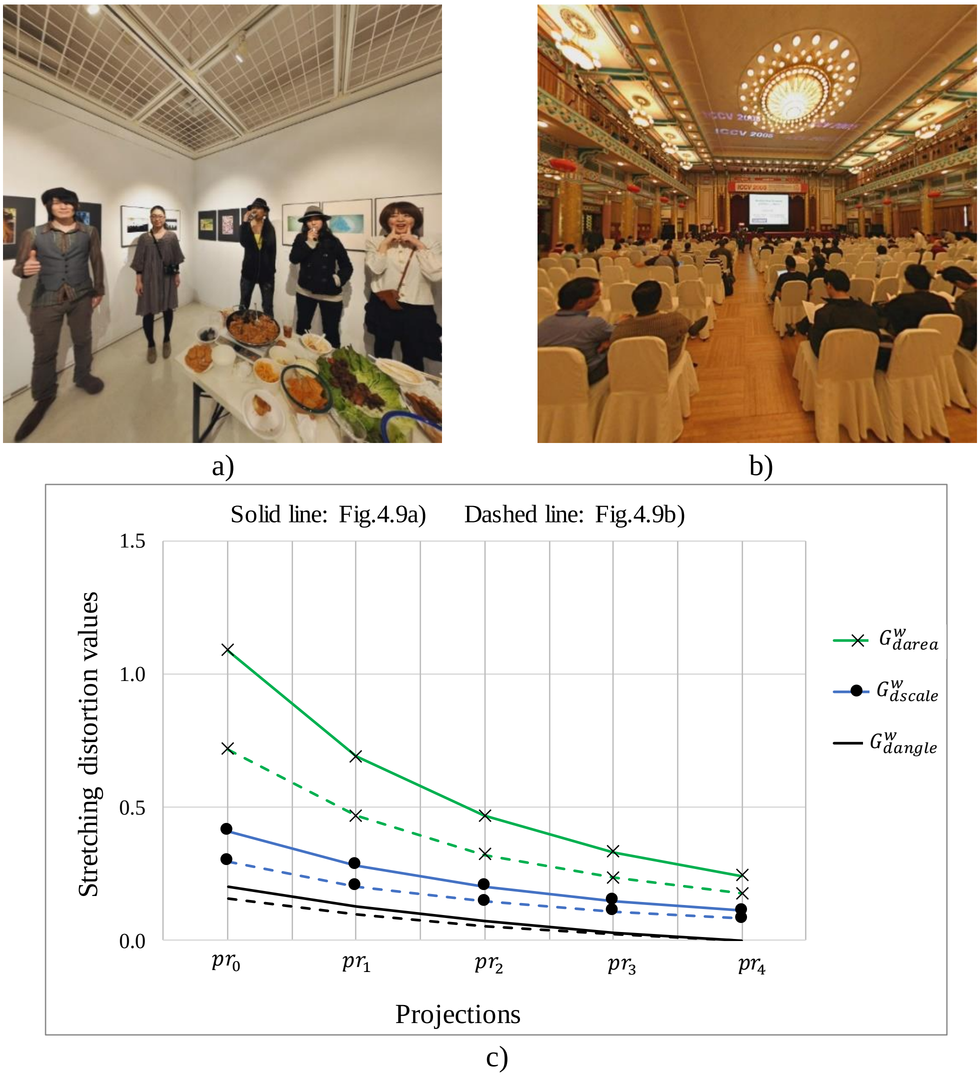
$$SF_{dscale} = G_{dscale}^{w}(0) - G_{dscale}^{w}(d_q) \tag{4.29}$$

$$SF_{darea} = G_{darea}^{w}(0) - G_{darea}^{w}(d_q) \, . \tag{4.30}$$

where SF_{dangle}, SF_{dscale}, SF_{darea}, obtained by (4.28) to (4.30), represent relative measures of global angle, scale, and area distortion, respectively.

C. Line Detection and Bending Feature Extraction

To compute the line bending features for $d = d_q$, straight lines are first detected in the viewport rendered with $d = 0$, according to the procedure described in Section 4.3.1. The resulting lines are then transformed to spherical coordinates, using the backward-projection with $d = 0$, and re-projected on the viewport plane using the forward-projection, with $d = d_q$. Finally, bending features are computed based on the line distortion measures, and pooling functions, proposed in Section 4.3.2. However, since line distortions occurring in salient regions of the viewport are likely to have a higher perceptual impact, these measures are weighted by saliency scores,



a)

b)

c)

Figure 4.9. a)-b) Two rendered viewports with a square FoV of 110° and with a high and low amount of perceived stretching distortion, respectively; c) Plot of global stretching distortion values with respect to different projection centers, d, and computed forFigure 4.9a) and b).

similarly to what was done for the stretching features. Thus, six new line distortion measures are obtained, which are presented in Table 4.3.

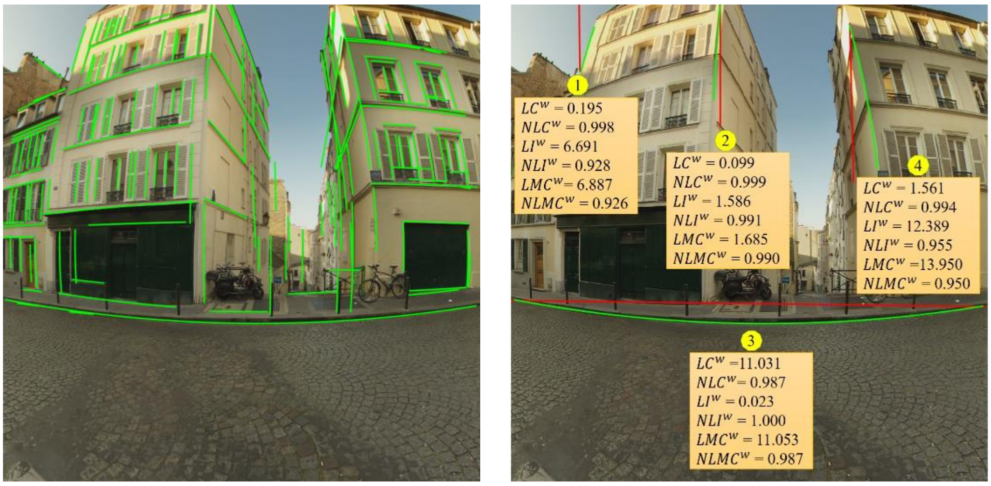
In Table 4.3, LC^w, LI^w, and LMC^w are the weighted line distortion measures and their corresponding normalized measures NLC^w, NLI^w, and $NLMC^w$, respectively; i is the line index and $\hat{L}$ is the line length; w^l is the line saliency weight, computed by averaging all the saliency values coincident with the projected line.

Figure 4.10b) depicts the line distortion measures for a few straight lines, projected with $d = 1$. As shown, all line distortion measures vary as expected from the perceived distortions.

To aggregate, in a single global value, the line distortion values computed for all projected lines, the proposed line pooling functions presented in Table 4.2 are applied, with a p value that was fixed at 90%. This value was found according to the procedure explained in Annex A. Considering the possible combinations of pooling strategies with distortion measures, a total of 24 potential line bending features are obtained.

Table 4.3. Weighted line distortion measures.

$LC_i^w = w_i^l \times LC_i$	(4.31)	$NLC_i^w = \dfrac{\hat{L}_i}{LC_i^w + \hat{L}_i}$	(4.32)
$LI_i^w = w_i^l \times LI_i$	(4.33)	$NLI_i^w = \dfrac{\hat{L}_i}{LI_i^w + \hat{L}_i}$	(4.34)
$LMC_i^w = LC_i^w + LI_i^w$	(4.35)	$NLMC_i^w = \dfrac{\hat{L}_i}{LMC_i^w + \hat{L}_i}$	(4.36)



a) b)

Figure 4.10. a) Projected lines with a square FoV of 110° and $d = 1$; b) A few projected lines and the respective line distortion measure values (scaled by 300); corresponding straight lines are depicted in red.

The bending features for a query projection center, d_q, are not subtracted from the reference, as in stretching features, since the values of these features for the reference viewport are always zero (no bending occurs for the rectilinear image).

D. SVR-based Quality Prediction

To predict a quality score for the viewport obtained with $d = d_q$, the selected stretching and bending features are computed, and are combined using Support Vector Regression (SVR). The choice for the SVR is justified by its proved efficiency in existing solutions for image and video quality assessment, e.g. [85][121][122].

To find a regression function that accurately predicts the quality scores from the input features, an SVR model has to be computed with some training data. This step was accomplished using a part of the GPP viewport dataset described in Chapter 3 (Section 3.2) and associated CMOS values, that are considered the target ground truth quality values. A linear SVR was used with a margin of tolerance ε (influences the number of support vectors used for prediction) and penalty factor C, also referred as cost or regularization constant. The ε and C parameters were optimized by grid-search: a finite set of reasonable values was evaluated, and the best values found for ε and C were 1.2 and 1.4, respectively. The Matlab SVR implementation was used.

4.4.2 Experimental Procedure and Performance Evaluation

In this section, the proposed stretching and bending features are first evaluated, with the goal of selecting a subset of potentially relevant features to be used in the final quality prediction model. After describing the SVR training and testing procedures, the performance of the trained model is then assessed and compared to benchmark solutions.

A. *Feature Selection*

In Sections 4.4.1.B and 4.4.1.C, several stretching and bending features have been defined. However, it is very likely that some of them are more relevant than others, to the perceived viewport quality prediction, or are highly correlated with others, and may have less impact on the SVR performance. Furthermore, training the model with a too high number of features may lead to overfitting. Since it will be very time consuming to train and test the model for all possible subset of features, the following procedure was adopted to select a subset of potentially relevant features (as suggested in [123]):

- **Feature correlation with perceptual scores** - a good feature is expected to be well correlated with the CMOS values obtained in the subjective evaluation. In this step, the Pearson Linear Correlation Coefficient (PLCC) between each feature and the CMOS values is computed.
- **Correlation between features** - highly correlated features convey similar information and thus are potentially redundant, if both are kept. In this step, the PLCC between features (inter-features) is computed.
- **Decision step** - If the PLCC between two features (or a group of features), is higher than 0.9, then the selected feature is the one having the highest PLCC value, relative to CMOS values, and the other feature is (are) removed.

For the stretching feature selection, the proposed stretching features were extracted only for the viewports where the bending distortion was not dominant; from the analysis of Figure 3.4 in the previous chapter, these correspond to the images contained in the group "pr_i better than pr_0". Since images *Photography shop* and *Museum* are in this group for both FoVs, the 48 viewports from these images were used. Table 4.4 presents the results of the first two steps. Since the inter-features PLCC values are higher than 0.9, after step 3 only the SF_{darea} feature was selected.

Table 4.4. Stretching feature correlation (PLCC) between feature and CMOS values and between features.

Feature correlation with perceptual scores		Correlation between features			
			SF_{dangle}	SF_{dscale}	SF_{darea}
SF_{dangle}	0.44	SF_{dangle}	-	-	-
SF_{dscale}	0.51	SF_{dscale}	0.97	-	-
SF_{darea}	0.54	SF_{darea}	0.93	0.99	-

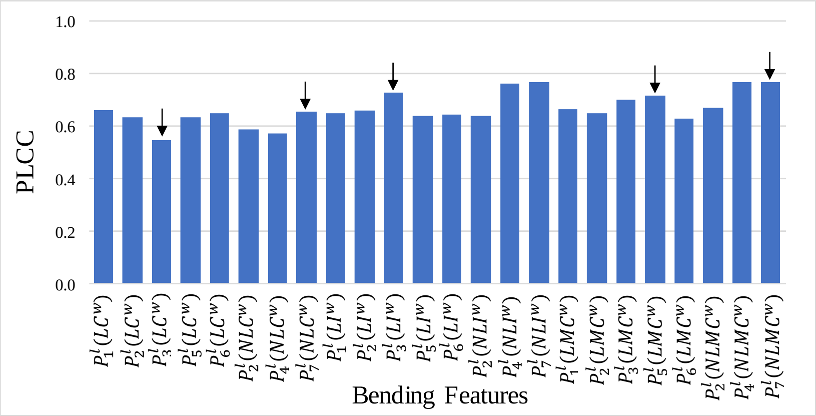


Figure 4.11. Correlation between bending features and CMOS values; the black arrows signalize the selected features.

Table 4.5. Correlation between selected bending features.

	$P_3^l(LC^w)$	$P_7^l(NLC^w)$	$P_3^l(LI^w)$	$P_5^l(LMC^w)$	$P_7^l(NLMC^w)$
$P_3^l(LC^w)$	-	-	-	-	-
$P_7^l(NLC^w)$	0.82	-	-	-	-
$P_3^l(LI^w)$	0.73	0.88	-	-	-
$P_5^l(LMC^w)$	0.69	0.86	0.85	-	-
$P_7^l(NLMC^w)$	0.67	0.84	0.82	0.76	-

For the bending feature selection, the proposed bending features were extracted only for the viewports where the stretching distortion was not dominant; this happens for images *Buildings 1* and *Buildings 2* contained in the group "pr_i worse than pr_0" of Figure 3.4 in the previous chapter; accordingly, the 48 viewports from these images were used. Figure 4.11 depicts the resulting PLCC values between each feature and CMOS values; the black arrows signalize the features remaining after step 3; Table 4.5 presents the correlation (PLCC) between the selected features.

As shown by Table 4.4 and Figure 4.11, while the selected bending metric showed to be well correlated with the perceptual scores (CMOS), this correlation is not so good for the stretching measures. Thus, additional experiments were performed (which are included in Annex B), considering the viewport depth map. The goal was to see if the performance of the stretching features could be improved if they were weighted by depth scores instead of saliency scores. This is based on the fact that geometric distortions have a higher perceptual impact for objects closer to the camera. However, as shown in Annex B, the performance of the stretching feature was not improved much. Furthermore, computing the depth map was more computationally demanding than the saliency map, thus the depth map was no further considered.

Concluding, from the initial set of features, only six are included in the subset of potentially relevant ones: SF_{darea}, $P_3^l(LC^w)$, $P_7^l(NLC^w)$, $P_3^l(LI^w)$, $P_5^l(LMC^w)$ and $P_7^l(NLMC^w)$. After having evaluated the proposed metric with this reduced subset, the impact of the complete set of features will be also assessed, to understand the contribution of the remaining features.

B. *SVR Training and Testing*

After feature selection, the SVR training and testing steps were performed using the Cross-

Validation (CV) procedure, which was applied 200 times. In each complete CV run, the GPP viewport dataset, described in the previous chapter (Section 3.2), was first randomly split into ten subsets (or folds) of equal size. Then, nine of the ten folders, together with the corresponding CMOS values, were used to train the model, and the remaining fold was used as testing (or validation) set. After ten CV iterations, each fold was used exactly once as testing data. Thus, in each complete run of the CV procedure, all viewports were used for training and testing of the SVR model.

C. CMOS Prediction Evaluation

The metric performance was evaluated using the PLCC, the Spearman rank-order correlation coefficient (SROCC), and the root-mean-square error (RMSE), between ground truth and predicted CMOS values. Figure 4.12 depicts the resulting PLCC value for each CV run (averaged over the ten folds), using the six selected features on the prediction model, showing that the metric performance is quite stable along the different runs.

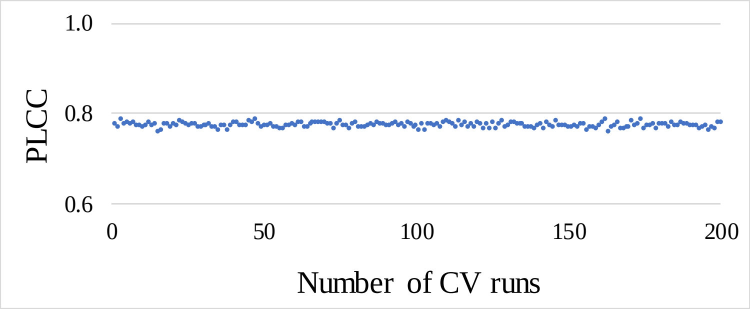


Figure 4.12. PLCC values for each run of the CV procedure.

The resulting PLCC, SROCC, and RMSE (averaged over the 200 CV runs) are presented in Table 4.6 for the proposed metric, using the selected features and also all features. Moreover, two additional benchmark metrics were included; their description and results analysis are performed in the next section.

To evaluate the power-law transformation impact on the final results, the six selected features were extracted without the use of this transformation; the SVR model train and test was repeated using these features, resulting in PLCC, SROCC, and RMSE values of, respectively, 0.68, 0.76 and 1.39, which are lower than those resulting from the use of this transformation.

Table 4.6. Quality prediction performance for proposed and benchmark metrics.

Proposed metric, with selected features		**Proposed metric, with all features**	
PLCC	0.78	PLCC	0.82
SROCC	0.79	SROCC	0.83
RMSE	1.19	RMSE	1.08
Benchmark metric from [36]		**Benchmark metric from “Scenario 3” described in Annex A**	
PLCC	0.65	PLCC	0.51
SROCC	0.68	SROCC	0.59
RMSE	1.42	RMSE	1.62

To evaluate the impact, on the metric performance, of each feature (including those considered as not relevant in the feature selection procedure) the SVR model was built with the stretching feature, SF_{area}, and with the bending feature having the highest individual PLCC (shown in Figure 4.11), $P_7^l(NLMC^w)$. The remaining features were ordered by their individual PLCC values and each feature was added to the model one by one (iteratively). At each iteration, the SVR model was trained and tested using the CV procedure described before. Figure 4.13 depicts the evolution of the resulting PLCC value. As can be observed, the model performance is marginally improved when the number of features increases above the six previously selected, showing that, from a performance perspective, it is worthy to keep only those features.

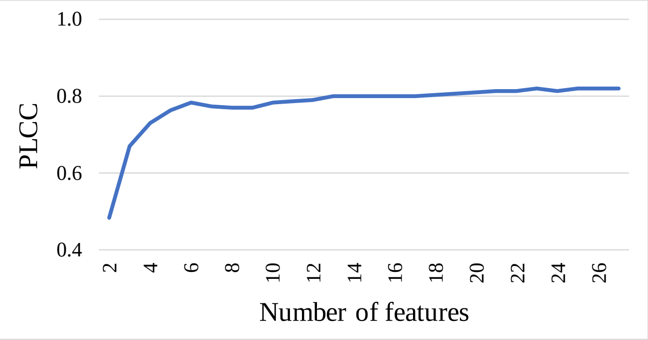


Figure 4.13. PLCC evolution for the proposed metric, when the features are added one by one.

Figure 4.14a) and Figure 4.14b) show the scatter plots of the ground truth versus predicted CMOS values considering all viewports on the dataset, for the proposed metric with the six selected features, and using all the features.

D. Comparison with Related Work

For comparison purposes, a benchmark metric with the conformality and bending measures proposed in [36], was defined. The bending measure is similar to (4.19) with poolings $P_2^l(.)$ and $P_4^l(.)$, and the conformality measurement, denoted as CM, is described by (2.55), in Chapter 2. In this work, both measures have been weighted additionally by the saliency. Since the CMOS scores represent the perceived quality relatively to the reference viewport (with d=0), a conformality feature was defined as

$$SF_{CM} = CM(0) - CM(d_q) \tag{4.37}$$

where $CM(d)$ is the conformality obtained for the viewport projected with d. The bending and conformality features were then used for SVR-based quality prediction (which was not used in [36]) using the GPP viewport dataset and the cross-validation procedure, previously described. The resulting PLCC, SROCC and RMSE are presented in Table 4.6, and the scatter plot of ground-truth versus predicted CMOS is depicted in Figure 4.14c). To assess the impact of using a stretching feature and the saliency weighing of the features, the SVR model was also trained and tested with the line bending features proposed in “Scenario 3” of [119], and described in Annex A. The resulting PLCC, SROCC and RMSE are included in Table 4.6, and the scatter plot of ground-truth versus predicted CMOS is depicted in Figure 4.14d). Both Table 4.6 and Figure 4.14 show that, with the proposed set of features, the predicted CMOS values are better correlated with the values resulting from subjective assessment.

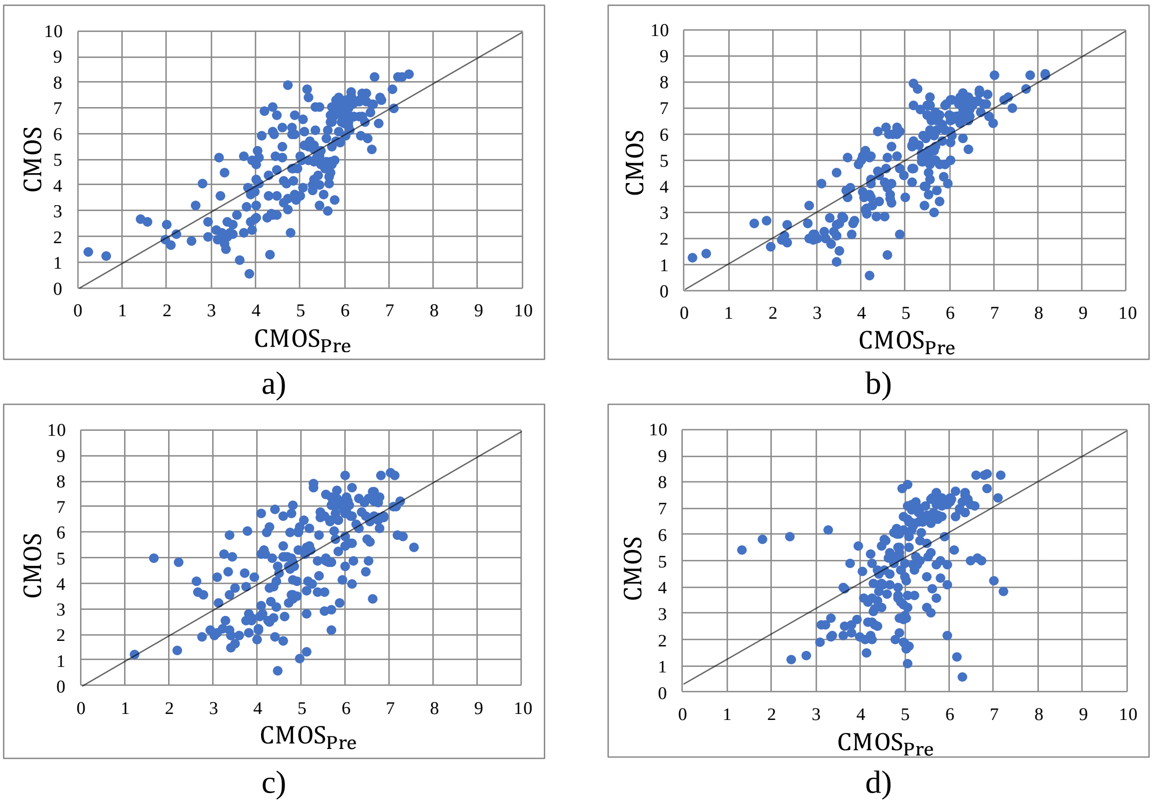


Figure 4.14. Ground truth CMOS versus predicted CMOS for: a) Proposed metric, with selected features; b) Proposed metric, with all features; c) Benchmark metric from [36]; d) Benchmark metric from "Scenario 3" described in Annex A.

4.5 GPP Optimization

This section describes a useful application of the proposed content-aware objective quality metric proposed in Section 4.4, which is the computation of the optimum value for the GPP parameter (d), to be used on the viewport rendering. This allows to minimize the perceived geometric distortions, by adapting the projection center, globally, to the viewport content, resulting in a content-aware general perspective projection (CA-GPP). The optimum value for parameter d can be obtained by iteratively running the SVR model for several d values and selecting the one that results in the highest CMOS value. Yet, an alternative procedure was also implemented and evaluated, where the viewport geometric distortion is modeled by a simple cost function, defined by one bending and one stretching metric; the optimum d is then obtained by iteratively minimizing this function, over a pre-defined set of d values. Both approaches are next presented.

4.5.1 SVR-based CA-GPP

The GPP projection parameter is optimized based on the predicted perceptual score obtained from SVR, as depicted in Figure 4.15. To automatically select the "best" GPP projection center, viewports are iteratively rendered for d values in the interval [0,1], with a step-size of $\Delta d = 0.25$; d values higher than 1 were not considered, since the fisheye effect becomes too much visible. For each rendered viewport, the six selected features described in Section 4.4.2.A, are extracted and fed to the SVR model that was built in Section 4.4.1.D, which predicts the corresponding quality score, CMOS$_{\text{Pre}}$. The selected projection center (d_{est}) is the one resulting

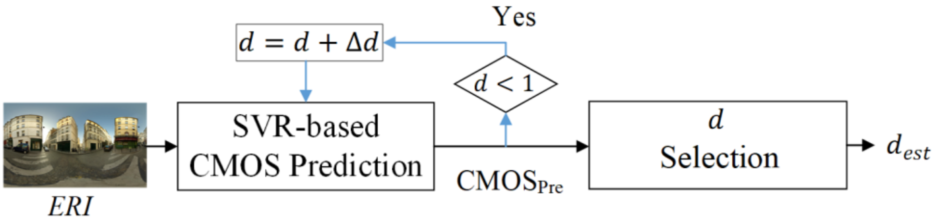


Figure 4.15. Proposed CA-GPP framework based on the SVR-based quality scores prediction.

in the highest CMOSPre value. Since the predicted quality scores are relative to a reference viewport (obtained with $d = 0$) if the highest CMOSPre value is less than 5, then the selected projection center is $d_{est} = 0$, which is the best one in this case (*cf.* Figure 3.4).

To perform a quantitative evaluation of the automatic GPP parameter selection, a suitable metric must be first defined. Consider d_{est}^{v} the estimated optimum projection center for viewport v, obtained with the proposed CA-GPP, and d_{opt}^{v} the optimum projection center for the same viewport, obtained from the GPP subjective assessment (the d value corresponding to the highest CMOS score). Note that using $\Delta d = 0.25$ on the CA-GPP, the evaluated d values are the same as those used on the GPP subjective assessment tests described in Chapter 3, i.e., $d = 0, 0.25, 0.5, 0.75, 1$. A simple metric could be the absolute difference between d_{est}^{v} and d_{opt}^{v}. However, a high difference in d_{est}^{v} and d_{opt}^{v} is not meaningful if the corresponding subjective scores are similar, meaning that the perceived quality of the corresponding viewports are also similar; on the contrary, a small difference in d_{est}^{v} and d_{opt}^{v} should be penalized if the perceived quality of the corresponding viewports is rather different. Thus, the proposed metric uses the corresponding CMOS values to make the evaluation and considering the perceived quality of different values for the projection center parameter. The prediction error for viewport v (PPE^{v}), is then defined as the absolute difference in the CMOS scores resulting for v when rendered with d_{est}^{v} and d_{opt}^{v}

$$PPE^{v} = \left|\mathrm{CMOS}_{d_{est}}^{v} - \mathrm{CMOS}_{d_{opt}}^{v}\right|. \tag{4.38}$$

Table 4.7 presents the average PPE per quality groupe (G1 - pr_i better than pr_0, G2 - pr_i similar to pr_0, G3 - pr_i worse than pr_0, as defined in Chapter 3 (Section 3.2.4), and shown in Figure 3.4), and the global maximum and average values (considering all the viewports evaluated on the subjective assessment) for the rectilinear, stereographic, and the proposed CA-GPP. The optimized projection achieves the lowest average PPE (per group) for groups G1 and G3, and also the lowest global "max PPE" and "average PPE" by a significant margin, which validates the proposed CA- GPP projection. As expected, the rectilinear projection achieves the minimum PPE average value for G3, which is mainly composed by viewports with linear structures; this also justifies why the stereographic projection has the highest average PPE (per class) for group G3.

Table 4.7. Quantitative evaluation of compared projections.

	Average *PPE* per group			Max	Average
Projection	**G1**	**G2**	**G3**	***PPE***	***PPE***
Rectilinear	2.17	**0.41**	**0.00**	3.33	1.08
Sterographic	0.48	1.04	3.14	4.42	1.32
CA-GPP	**0.42**	**0.42**	**0.00**	**2.42**	**0.32**

4.5.2 Cost Function-based CA-GPP

The proposed SVR-based procedure to optimize the GPP could be also applied to globally optimize other sphere to plane projections. However, it requires subjective tests to obtain the ground truth quality scores, to be used for SVR training. For a projection with more than one parameter (e.g., Pannini, which has two parameters, d and vc) it would be also more time consuming to perform those tests. Thus, a simpler procedure was also conceived, where the projection parameter (d) is optimized based on a simple cost function, that includes just two geometric distortions measures, one for stretching and another for bending. Although in [10][56] a similar procedure was used, it was not validated with respect to the perceived distortion, nor the cost function parameters, that determine how much weight should be applied to each type of geometric distortion, were obtained perceptually.

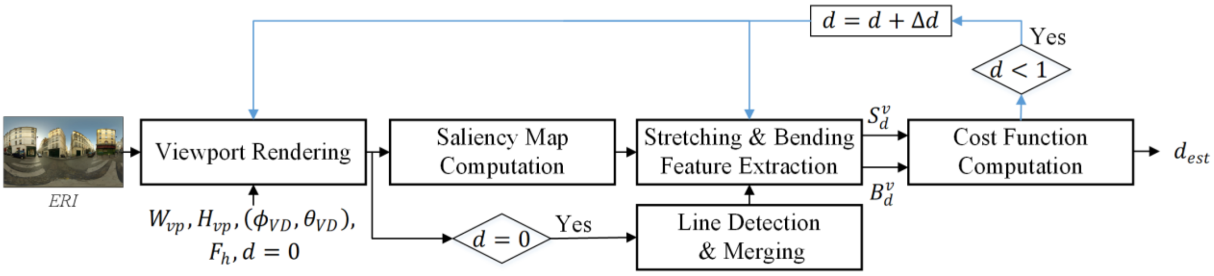


Figure 4.16. Cost function-based CA-GPP framework.

Figure 4.16 depicts the architecture of the proposed cost function-based CA-GPP, which has several common points with the SVR-based CA-GPP (*cf.* Figure 4.7 and Figure 4.15); the differentiating ones are the used distortion features and cost function, which are detailed below:

- **Stretching and Bending Feature Extraction** - The best stretching feature, G^{w}_{darea}, proposed in Section 4.4.1.B, and one of the best bending features - Line Measure Combination, LMC^{w}, with line pooling function, P_5^l proposed in Section 4.4.1.C, were used. Both measures are weighted by saliency scores, obtained from the viewport saliency map as described in Section 4.4.1.A. The line detection and merging, required before bending feature extraction, is accomplished as presented in Section 4.3.1.

- **Cost Function Computation** - To estimate the optimum projection center (d_{est}), that minimizes the perceived geometric distortions in the viewport, the stretching and bending features are combined in a distortion cost function, defined as

$$D(v, d) = \alpha S_d^v + \beta B_d^v \,, \tag{4.39}$$

where S_d^v and B_d^v are, respectively, the stretching and bending features of viewport v, when rendered with projection center d; α and β are parameters that determine how much weight should be applied to each type of geometric distortion. The optimum projection center for rendering the viewport v, is then estimated by

$$d_{est}^v = \min_d (D(v, d)) \,. \tag{4.40}$$

In this Thesis, (4.40) was solved by evaluating the cost function $D(v, d)$ for every d in the interval [0,1], with a step-size $\Delta d = 0.1$. Parameters α and β, that must be obtained beforehand, have a critical importance in the context of the minimization of (4.40) and thus on the predicted projection center value. Related works [10][36][55] have established these parameters heuristically, with no defined method. In this work, they are derived from the subjective tests results, to guarantee that the importance of each factor reflects the human sensibility to each type of geometric distortion. Accordingly, it was assumed a linear relationship between the CMOS scores (before normalization) obtained from the GPP subjective tests and the difference on geometric distortion for the compared viewports. This relationship can be defined as

$$\mathrm{CMOS}_d^v = k(D_0^v - D_d^v) = \alpha'(S_0^v - S_d^v) + \beta'(-B_d^v) \tag{4.41}$$

where k is a positive constant and $(\alpha', \beta') = k \times (\alpha, \beta)$; D_0^v and S_0^v are, respectively, the distortion cost function and the stretching feature for the reference viewport ($d = 0$), for which $B_0^v = 0$; CMOS_d^v is the comparative MOS value for viewport v when rendered with projection center d. For the optimum d solution, (4.41) becomes

$$\mathrm{CMOS}_{d_{opt}}^v = \alpha'\left(S_0^v - S_{d_{opt}}^v\right) + \beta'\left(-B_{d_{opt}}^v\right). \tag{4.42}$$

and subtracting (4.42) to (4.41), results in

$$\mathrm{CMOS}_d^v = \alpha'\left(S_{d_{opt}}^v - S_d^v\right) + \beta'\left(B_{d_{opt}}^v - B_d^v\right) + \mathrm{CMOS}_{d_{opt}}^v\,. \tag{4.43}$$

For K viewports, whose d_{opt} and CMOS values are obtained from the GPP subjective assessment tests, (4.43) results in $4 \times K$ equations (as there are four d values, besides the optimum one), that can be solved for α' and β' using a linear least squares method. Since α', β' are related with α, β by the same multiplicative constant, k, they can also be used in the minimization problem defined by (4.40). The best values for α' and β' were found at 0.67 and 0.99, respectively.

4.5.3 Comparative Results

Table 4.8 presents the average PPE per groupe of viewport qualities (G1, G2 and G3), and the global maximum and average PPE values, for the SVR-based (CA-GPP) and for the cost function-based (CA-GPP*), GPP optimization; the PPE values were computed according to the procedure explained in the previous section.

As shown by Table 4.8, the CA-GPP has the best performance for G1 and G2, and the same performance as CA-GPP* for G3. Both CA-GPP and CA-GPP* achieved the same PPE max value, but the PPE average value for CA-GPP is lower than for CA-GPP*. Also, both outperform the rectilinear and stereographic projections.

Table 4.8. Quantitative evaluation of compared projections.

	Average *PPE* per group			**Max *PPE***	**Average *PPE***
Projection	**G1**	**G2**	**G3**		
Rectilinear	2.17	**0.41**	**0.00**	3.33	1.08
Sterographic	0.48	1.04	3.14	4.42	1.32
CA-GPP*	0.46	0.75	**0.00**	**2.42**	0.44
CA-GPP	**0.42**	**0.42**	**0.00**	**2.42**	**0.32**

It is worthy to note that the cost function-based approach also requires subjective tests to obtain the ground truth quality scores, that are used to establish the cost function parameters; however, in this case only a small number of viewports and associated quality scores is needed, so the subjective tests can be implemented in a fast way. On the contrary, the SVR-based approach requires a large number of viewports and associated quality scores to train the model, involving

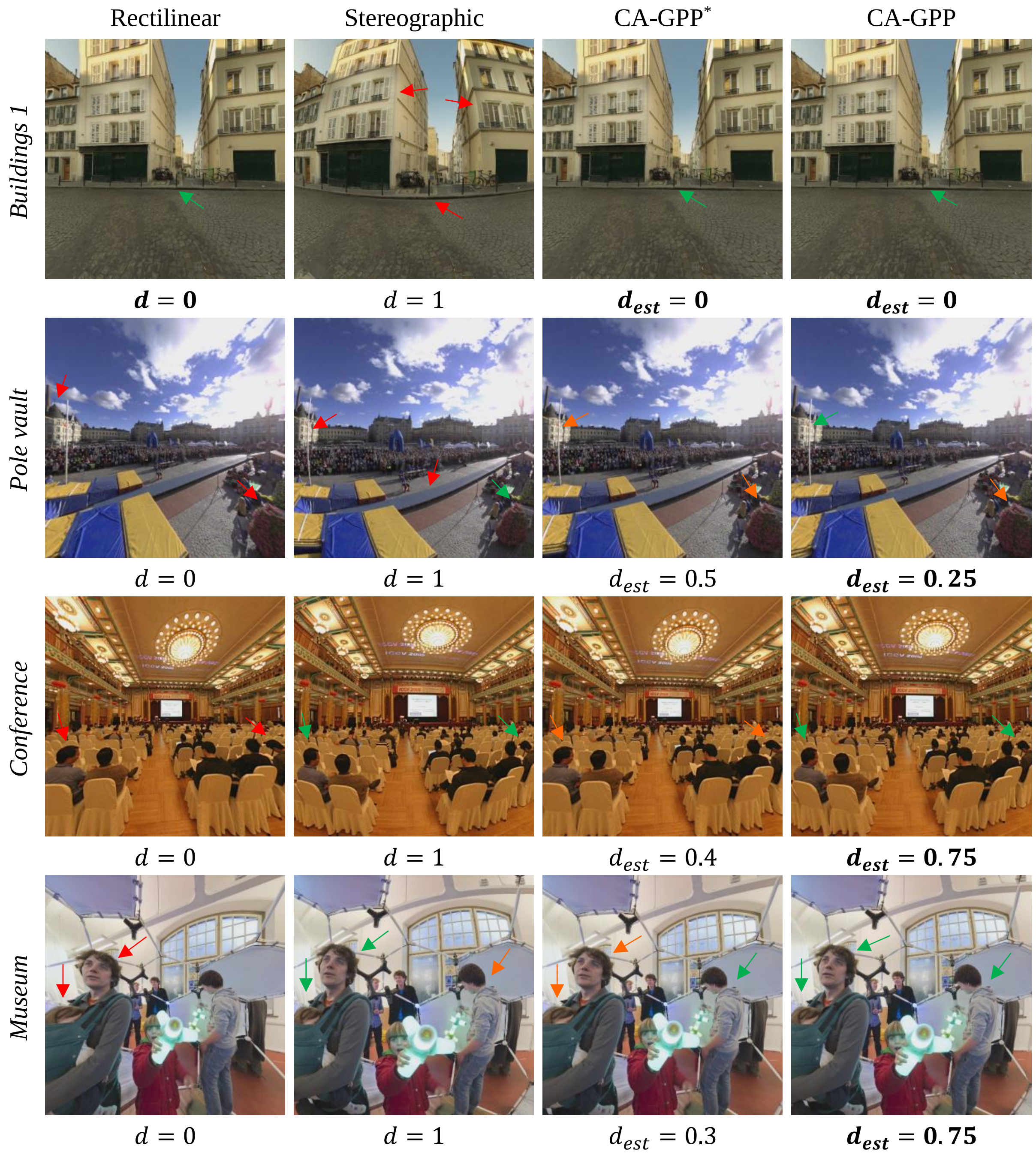


Figure 4.17. Example of viewports rendered with different projections. The red, orange, and green arrows indicate, respectively, the objects/regions with high, medium, and low geometric distortions.

very time-consuming subjective tests, notably for those projection with more than one parameter (e.g., the Pannini projection).

Figure 4.17 shows a few viewports obtained with rectilinear, stereographic, and CA-GPP projections, using a squared FoV of 110°. As can be figured out, the viewports obtained with the content-aware projections result in less geometric distortions. For *Buildings 1*, which has predominant straight lines, both CA-GPP and CA-GPP* coincide with the rectilinear. For *Pole vault*, both CA-GPP and CA-GPP* achieved a better balance between bending and stretching than rectilinear and stereographic; however, the pole on the left side is less deformed for

CA-GPP than for CA-GPP*. For *Conference* and *Museum*, which have salient objects located at the viewport borders and close to the camera, both CA-GPP and CA-GPP* preserve the object shapes better than rectilinear. For these two images, the CA-GPP results in viewports with a visual quality close to the ones obtained with stereographic, and thus with better object conformality than with CA-GPP*.

4.6 Final Remarks

This chapter proposed a content-aware objective quality metric to assess the perceived geometric distortions of viewport images, obtained by rendering omnidirectional images with the general perspective projection (GPP). The proposed metric is based on a set of features, extracted from the viewport images, that account for the two main perceived geometric distortions resulting from the sphere to the plane projection: objects deformation due to stretching, and straight line distortion due to bending. A selected subset of features, and the viewport comparative MOS (CMOS) scores, obtained using the rectilinear projection as reference, were used to build a quality prediction model based on SVR. The experimental results show that the proposed quality prediction model allows to predict the viewport CMOS score with a Pearson correlation coefficient close to 0.8, and when the GPP projection center varies between 0 (rectilinear projection) and 1 (stereographic projection).

Moreover, two procedures were developed to automatically optimize the GPP projection parameter, d, in a perceptual sense, resulting in content-aware general perspective projections, CA-GPP and CA-GPP*. In CA-GPP, d is obtained based on the proposed SVR-based quality prediction model. In CA-GPP*, d is obtained by minimizing a simple cost function that models the resulting geometric distortions through a linear combination of a bending and a stretching metric. Both CA-GPP and CA-GPP* showed significant performance improvement when compared to the popular rectilinear and stereographic projections.

The work achieved in this chapter was included in three published conferences and one journal paper, presented in Table 3.3.

Table 4.9. Publications related to this chapter.

Paper	Type
F. Jabar, J. Ascenso, and M.P. Queluz, “Perceptual Analysis of Perspective Projection for Viewport Rendering in 360° Images,” Proc. of IEEE International Symposium on Multimedia, Taichung, Taiwan, Dec. 2017.	Conference
F. Jabar, M.P. Queluz, and J. Ascenso, “Objective Assessment of Line Distortions in Viewport Rendering of 360° Images,” Proc. of the IEEE International Conference on Artificial Intelligence and Virtual Reality, Taichung, Taiwan, Dec. 2018.	Conference
F. Jabar, J. Ascenso, and M.P. Queluz, “Content-Aware Perspective Projection Optimization for Viewport Rendering of 360° Images,” Proc. of IEEE International Conference on Multimedia and Expo, Shanghai, China, Jul. 2019.	Conference
F. Jabar, J. Ascenso, and M.P. Queluz, “Objective Assessment of Perceived Geometric Distortions in Viewport Rendering of 360° Images,” IEEE J. Sel. Top. Signal Process., vol. 14, no. 1, pp. 49–63, Jan. 2020.	Journal

Chapter 5

Object-based Geometric Distortion Metric

5.1 Introduction

The stretching distortions metrics proposed in the previous chapter were based on the Tissot indicatrices, which are content independent, and cannot capture the perceptual impact of the stretching distortion with high accuracy, even when weighed by saliency scores. The work developed in this chapter seeks to overcome this limitation, by measuring the stretching distortion of semantic objects, and thus achieving a higher correlation with the perceived distortions. This approach was inspired by the fact that human perception is more sensitive to the stretching distortion of objects with semantic meaning, e.g., the human body.

The subjective tests reported on Chapter 3 did not discriminate between stretching and bending, i.e., both distortion types were simultaneously visible in several viewports. Accordingly, to better evaluate the perceptual impact of the stretching distortion (thus, without the influence of the bending) and collect the ground truth subjective scores required for the design of object-based stretching distortion metrics, an additional set of subjective tests was conducted, that are described in this chapter. Moreover, the Pannini projection was used in these tests, since it allows to have different levels of object stretching (by changing the projection center, d) while keeping the vertical lines straight.

In this context, this chapter addresses the following objectives:

- Subjectively assess the stretching distortion impact on the perceived quality of the viewport image, using the Pannini projection for viewport rendering.
- Develop an object-based distortion metric that automatically assesses the stretching distortions in the viewport image, after rendering.

The rest of this chapter is organized as follows. Section 5.2 describes the subjective test campaign to assess the perceptual impact of stretching distortion and the corresponding analysis. Section 5.3 details the proposed object-based stretching distortion metrics. Section 5.4 presents, and analyses, the metrics performance evaluation results. Finally, Section 5.5 summarizes and concludes this chapter.

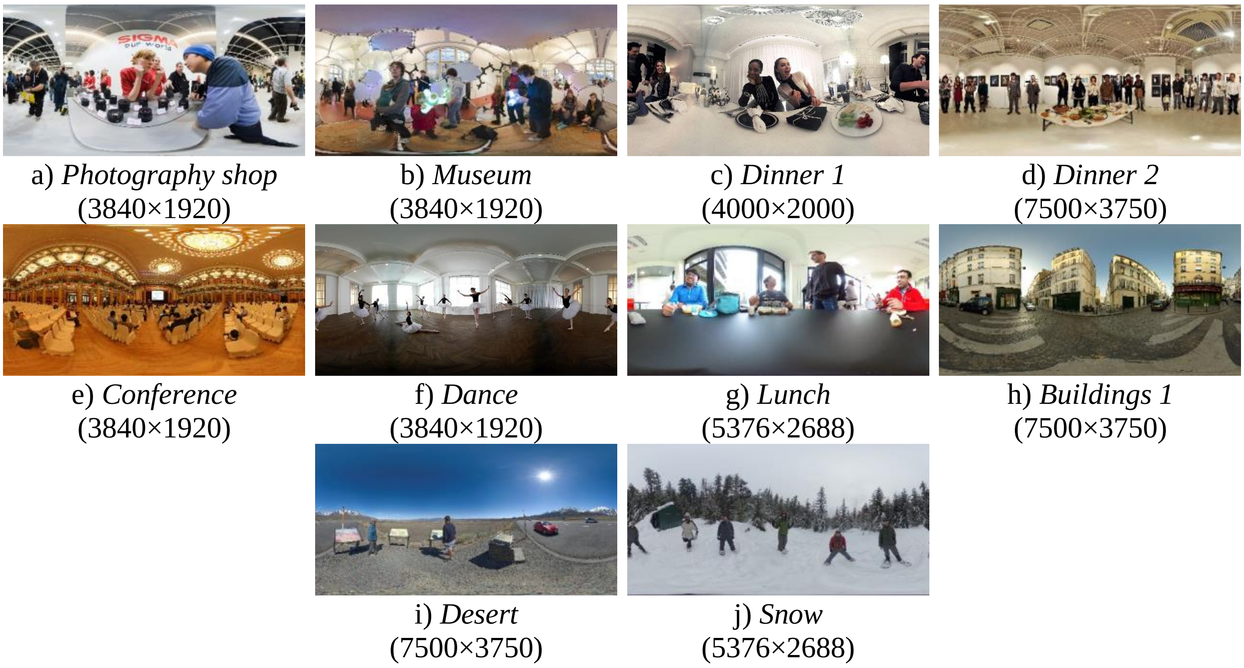
5.2 Subjective Assessment of the Stretching Distortion Effect

This section describes a crowdsourcing-based subjective evaluation of viewport images, aiming to assess the perceptual impact of the stretching distortion; the viewport images were rendered using the Pannini projection which, for a particular choice of its projection parameters, results in a pure rectilinear projection, for which the stretching effect is most evident. After describing the considered omnidirectional images dataset and the subjective evaluation methodology, the final subjective test results and its analysis are then successively presented.

5.2.1 Dataset

Ten omnidirectional images in equirectangular format (*ERI*), extracted from the datasets available in [10] and [52], were used in the subjective assessment. The images, and their spatial resolutions, are depicted in Figure 5.1. This set of images includes six images that were already used in the previous subjective tests, described in Chapter 3, and four new images which are *Dinner 1*, *Dance*, *Lunch*, and *Snow*. These images were selected carefully to have perceptual relevant objects (such as people near and far away from the camera) where the stretching has a high perceptual impact.

For each image, three viewports were rendered, corresponding to three different viewing directions, with 70% overlapping FoV between successive directions. This allows to compare different levels of stretching distortion of the same objects, when these objects appear in different positions on the viewports. In the subjective test described in Chapter3 (Section 3.2), the viewports correspond to the front view, 45° to the right, and 45° to the left. In this work, to guarantee that the viewports are rendered from the part of omnidirectional image that includes the objects or regions where the users attention is often attracted for, the saliency maps available for the images taken from [52], and the attention-related model proposed in [124] for the images taken from [10], were used. As an example, Figure 5.2 depicts the *Museum* image and its saliency map (available in [52]) with the three identified viewport regions, confirming that the viewports used for the subjective assessment contain image regions which attract a significant amount of attention.



a) *Photography shop* (3840×1920)

b) *Museum* (3840×1920)

c) *Dinner 1* (4000×2000)

d) *Dinner 2* (7500×3750)

e) *Conference* (3840×1920)

f) *Dance* (3840×1920)

g) *Lunch* (5376×2688)

h) *Buildings 1* (7500×3750)

i) *Desert* (7500×3750)

j) *Snow* (5376×2688)

Figure 5.1. Omnidirectional images used in the subjective tests, and their spatial resolution.

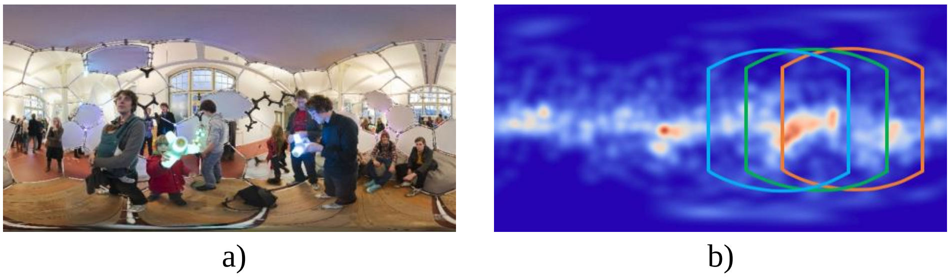


Figure 5.2. a) ***Museum*** **image and b) its saliency map available in Salient360! [52], and the viewport regions with 70% overlapping between successive directions.**

For each viewing direction, two viewports were rendered with the Pannini projection (PP), using different parameter values: PP_1 $(d = 0, vc = 0)$; PP_2 $(d = 0.5, vc = 0)$. Thus, for each image, six viewports were produced, denoted as VP_i, $i = 1, 2, \dots, 6$, where VP_1,VP_2,VP_3 correspond to PP_1, and VP_4, VP_5,VP_6 are correspond to PP_2 (*cf.* Figure 5.3). PP_1, which is a rectilinear projection, was selected since it is often used for viewport rendering of omnidirectional images and results on strong objects stretching. PP_2 was included to get, for the same viewing directions, different levels of stretching distortion of the objects. The viewports were rendered with a F_h of either 110° or 115° (presented for each *ERI* on the bottom of Table 5.1), and with a spatial resolution of 856×856 pixels ($AR = 1$); as already mention in Chapter 3, this resolution was recommended in [53] for subjective tests, and allows the simultaneously display of two viewports, side-by-side, in typical monitors. The F_h of 110° was selected based on the study described in Chapter 3 (Section 3.3). As shown in Table 5.1, for some of the images a F_h of 115° was used to guarantee that the main objects were not cut by the image borders.

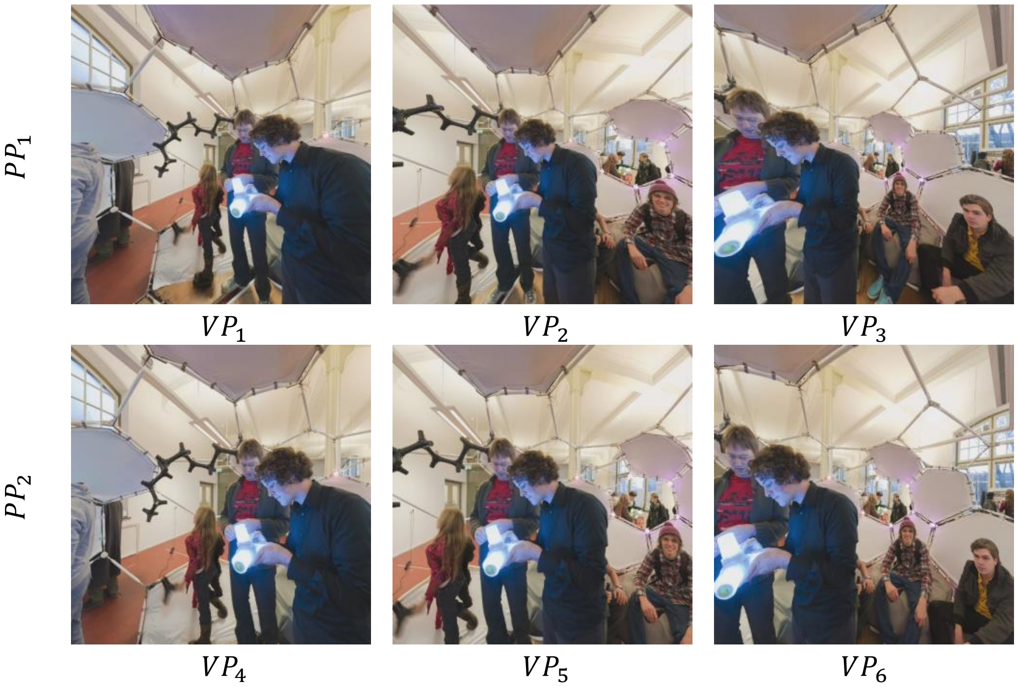


Figure 5.3. Viewport examples rendered from ***Museum*** **image with** PP_1 **and** PP_2**.**

Table 5.1. Selected pairs (indicted in ✓), and F_h for each omnidirectional images.

Pair	*Photography shop*	*Museum*	*Dinner 1*	*Dinner 2*	*Conference*	*Dance*	*Lunch*	*Buildings 1*	*Desert*	*Snow*
(VP_1, VP_2)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(VP_2, VP_3)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(VP_1, VP_3)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(VP_1, VP_4)	✓	✓	×	×	×	✓	×	×	✓	✓
(VP_2, VP_5)	✓	✓	×	×	×	×	✓	×	✓	✓
(VP_3, VP_6)	×	✓	✓	×	×	×	✓	×	✓	✓
F_h	110°	110°	115°	110°	115°	115°	115°	115°	110°	115°

5.2.2 Subjective Evaluation Method

As in Chapter 3 (Section 3.3.2), where the FoV impact was subjectively assessed, the pairwise comparison (PC) method was also chosen for the subjective evaluation of the stretching distortion. However, in this case, the viewer is asked to observe a pair of rendered viewports that are shown side by side, and to either select the one that has higher quality in his opinion, or to consider that both have the same quality.

As depicted in Figure 5.4, six comparisons were made per omnidirectional image: a complete set of comparisons between the viewports rendered with PP_1, and three additional comparisons between the viewports rendered with PP_1 and PP_2, and having the same viewing direction. The comparisons between viewports rendered with PP_2 were not considered to limit the test duration; furthermore, these viewports have a similar level of stretching distortion. Also, since for some of the omnidirectional images and viewing directions the bending distortion was visible for viewports rendered with PP_2, these were excluded from the subjective test. Table 5.1 presents the selected pairs and horizontal field of view, F_h, for each omnidirectional image.

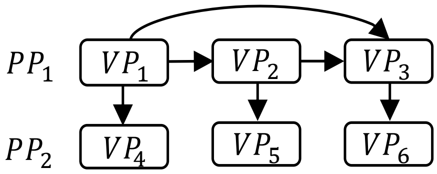


Figure 5.4. Considered comparisons per omnidirectional image.

In total, 45 viewport pairs were considered. Due to the COVID-19 pandemic, the subjective tests were no longer performed on IST premises, as the previous ones; instead, a web-based crowdsourcing interface was designed to perform the visualization of the stimuli and to collect the subjective scores. This interface, similar to the one described in Section 3.3.2 (except for the grading scale and viewport visualization), presents two viewport images, ‘A’ and ‘B’, side by side as shown in Figure 5.5, and requires a monitor with a minimum resolution of 1920×1080 pixels, and with a minimum diagonal size of 13-inch. To participate in the test, an invitation email with detailed instructions was sent to several observers; the observers were asked to not perform the test if they do not have a monitor with the aforementioned characteristics. Before starting the subjective test, the observers were asked to: *i)* open the subjective test interface in their web browser and put it on full-screen mode; *ii)* type their name and age in the top of the page; *iii)* select their monitor size in inches. The instructions about the subjective test procedure were shown on the same page. Subsequently, to familiarize the observer with the stretching

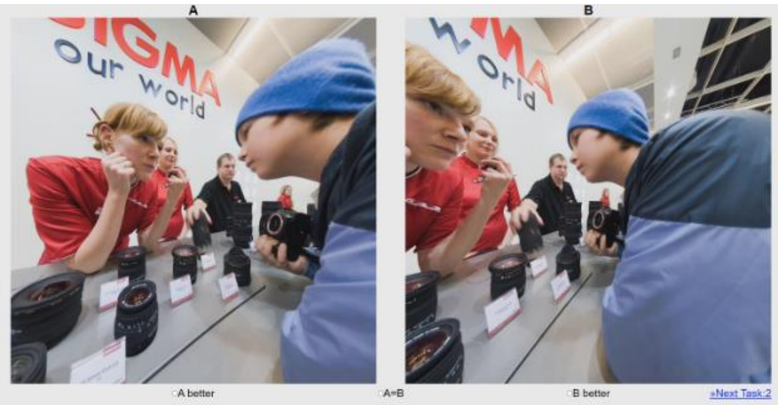


Figure 5.5. Web-based crowdsourcing interface.

distortion's characteristics and the evaluation interface, a short training video was shown. The viewports used in the training video were not used for the actual test. During the test, the viewport pairs were shown in random order and position, and the observers were asked to judge which viewport image ('A' or 'B') had the best quality. To avoid random preference selections, the option 'A=B' was also included. A total of 32 subjects, aged between 21 and 58 years, from Instituto Superior Técnico (IST), performed the online subjective evaluation. The omnidirectional images, the rendered viewports, and the resulting subjective scores were made publicly available in [125].

5.2.3 Subjective Tests Results and Analysis

Outliers were first detected by computing the transitivity satisfaction rate, Ɍ, as described in Chapter 3 (Section 3.3.2.C). Four outliers were detected, and their subjective scores were not further considered. Next, for each compared viewports pair, $\left(VP_i, VP_j\right)$, the winning frequency, w_{ij}, which represents the number of times VP_i was preferred over VP_j, was computed. To solve the tie cases, a score of 0.5 was given to each viewport whenever the observer had chosen the option 'A=B'. Note that $w_{ij} + w_{ji} = O$, where O is the number of observers, and $w_{ii} = 0$. The probability of selecting VP_i against VP_j, is given by

$$P_{ij} = P\left(VP_i > VP_j\right) = \frac{w_{ij}}{O}. \tag{5.1}$$

To determine whether the difference on the number of times VP_i was preferred over VP_j (and vice-versa) is statistically significant, a statistical hypothesis test was performed according to the procedure suggested in [126]. After solving the tie cases, the subjective scores roughly follow a Bernoulli process $B(O, p)$, where O is the number of subjects and p is the probability of success in a Bernoulli trial. Figure 5.6 depicts the cumulative distribution function (CDF) for a Binomial distribution with $O = 28$ (the final number of observers, after outliers removal) and p=0.5, as suggested in [126], meaning that when comparing VP_i and VP_j both have the same chance of being selected. The CDF of the Bernoulli distribution can be expressed as [127]

$$CDF(k; O, p) = \sum_{i=1}^{\lfloor k \rfloor} \binom{O}{i} p^i \ (1-p)^{(O-i)} \tag{5.2}$$

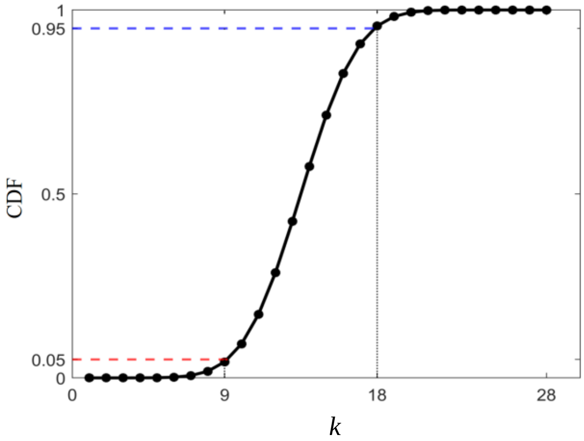


Figure 5.6. CDF with $O = 28$ and $p = 0.5$.

where k is the number of times that VP_i was selected over VP_j, p is the probability of selecting VP_i over the VP_j in the Bernoulli trial, and $\lfloor . \rfloor$ is the floor operator.

The resulting CDF value is the probability that the observers select, k times, VP_i over VP_j. The critical region for the statistical test is obtained from the CDF. To find out whether the number of times VP_i was preferred over VP_j is statistically significant, thus allowing to conclude that " VP_i is better than VP_j", a one-tailed binomial test was performed with a significant level of 0.05, with the following hypothesizes: H0 (VP_i is equal or worse than VP_j); H1 (VP_i is better than VP_j). In Figure 5.6, the CDF has values above the probability of 0.95 for $O \geq 18$ ($F(18{,}28{,}0.5) = 0.9564$). Therefore, if $k \geq 18$, the null hypothesis (H0) can be rejected. A similar statistical test was applied to find out if the number of times that VP_j was preferred over VP_i is statistically significant, thus allowing to conclude that "VP_i is worse than VP_j", with the following hypothesizes: H0 (VP_i is equal or better than VP_j); H1 (VP_i is worse than VP_j). In Figure 5.6, the CDF has values below the probability of 0.05 for $O \leq 9$ ($F(9{,}28{,}0.5) = 0.0436$). Therefore, if $k \leq 9$, the null hypothesis (H0) can be rejected. Note that the Bernoulli process is defined only for integer values, and non-integer values need to be rounded; the floor function is used for this purpose.

Figure 5.7 presents the probability of selecting VP_i over VP_j, computed by (5.1), for each compared pair, and for all considered omnidirectional images. The horizontal blue dashed line corresponds to the case where the vote count for VP_i is equal to, or greater than, 18, i.e., $P(VP_i > VP_j) = 18/28 = 0.643$. The horizontal red dashed line corresponds to the case where the vote count for VP_i is equal or less than 9, i.e. $P(VP_i > VP_j) = 9/28 = 0.321$. Values on or above the horizontal blue dashed line, and on or below the horizontal red dashed line, correspond to the cases where the difference in the votes between VP_i and VP_j is statistically significant; in the first case, VP_i has a higher perceived quality than VP_j; on the second case, VP_i has a lower perceived quality than VP_j. The values between the two horizontal dashed lines correspond to the cases where the difference in the votes between VP_i and VP_j is not statistically significant.

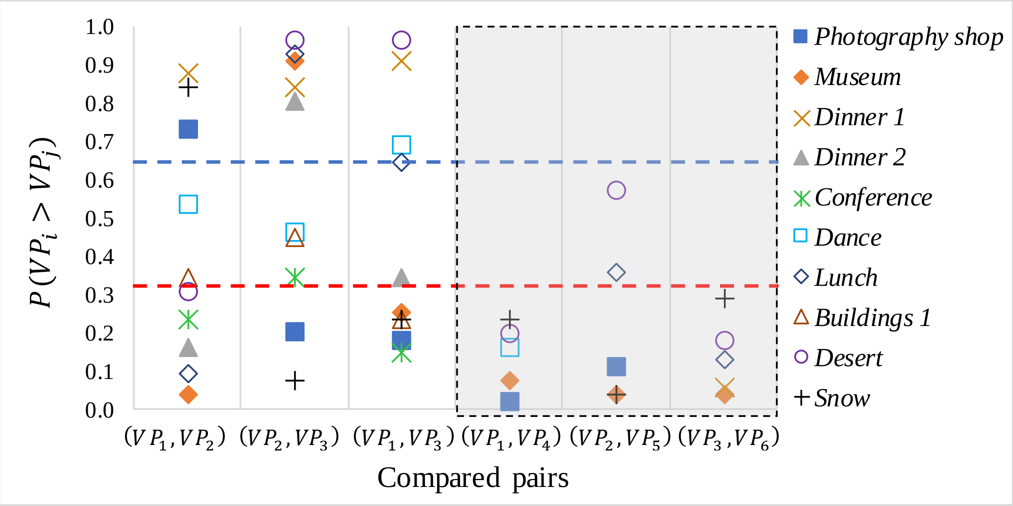


Figure 5.7. Preference probability of selecting VP_i over VP_j for compared pairs.

From the experimental results, the following conclusions can be obtained:

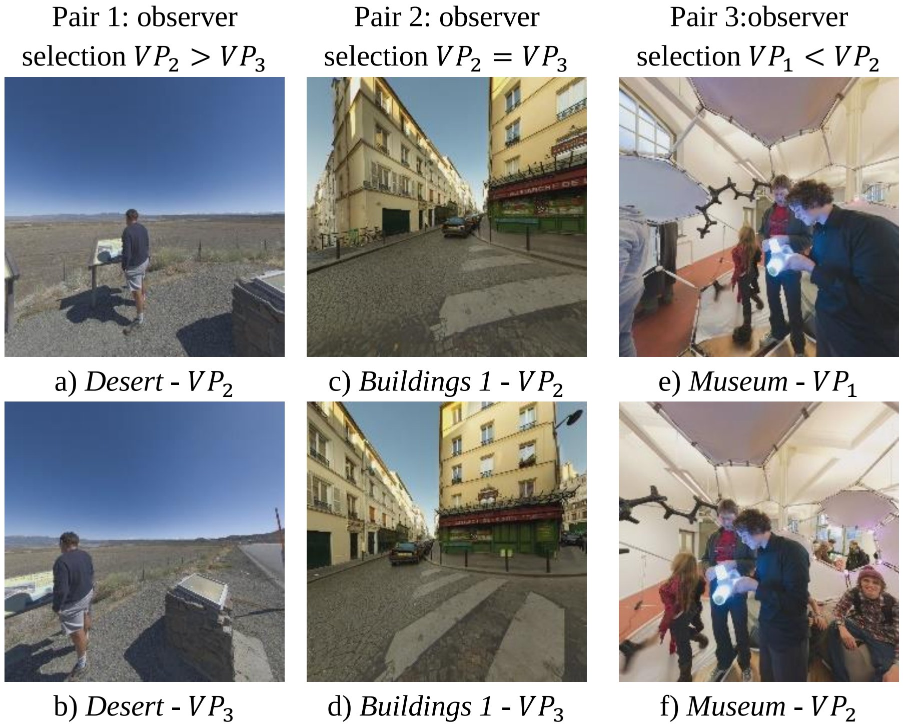
- As can be observed in Figure 5.7, the viewports rendered with PP_2 $(d = 0.5, vc = 0)$ were selected over those rendered with PP_1 $(d = 0, vc = 0)$, for most of the considered images (values inside the shaded area in Figure 5.7). This was expected since the rectilinear projection has a strong stretching effect, and this stretching decreases as the value of d increases.

- For the viewports rendered with PP_1, the preferred ones (by the subjects) are strongly dependent on the position of the main objects, because the stretching distortion has a higher perceptual impact when the objects are close to the image borders, and/or close to the camera. As an example, Figure 5.8 shows three pairs of viewports rendered with PP_1, together with the subject selections; due to the difference on viewing direction, the same objects are rendered in different positions of a viewport pair, having different perspective distortion.

- The human perception is very sensitive to the stretching distortion of the human body, which may justify why, as shown in Figure 5.8, most of the observers have selected VP_2 over VP_3 for *Desert*, and VP_2 over VP_1 for *Museum*, while for *Buildings 1* there is no clear choice between VP_2 and VP_3 .

- The stretching distortion of the background, e.g., sky, floor, building walls, has a lower perceptual impact than the stretching of foreground objects.

Finally, and since the subjects have used different monitor sizes, the impact of it on the subjective results was assessed. For that, and based on the used monitor size, the observes were divided into two groups: O_{g1}, containing the observers that have a monitor size in the range [13,16] inch, and O_{g2}, contains the observers that have a monitor size in the range [22,27] inch. Then, a paired sample T-test with a significant level of 0.05 was applied, to compare the preference probability, $P(VP_i > VP_j)$, between groups. This procedure is illustrated in Figure 5.9. The T-test results indicate that the null-hypothesis, i.e., that the two groups have similar means, cannot be rejected since the resulting p-value, 0.57, is much higher than the significant level of 0.05; this confirms that the difference on the subjective results from the two groups is not statistically significant.


Pair 1: observer selection $VP_2 > VP_3$

Pair 2: observer selection $VP_2 = VP_3$

Pair 3:observer selection $VP_1 < VP_2$

a) *Desert* - VP_2 c) *Buildings 1* - VP_2 e) *Museum* - VP_1

b) *Desert* - VP_3 d) *Buildings 1* - VP_3 f) *Museum* - VP_2

Figure 5.8. Rectilinear viewports, in pairs, evaluated by observers.

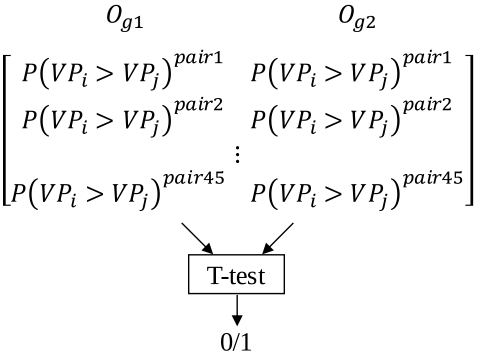


Figure 5.9. T-test procedure for evaluating if there is a significant impact of screen size on the preference probability.

5.3 Object-Based Stretching Distortion Measurement

This section describes two new approaches for measuring the object shape distortion in viewport rendering of omnidirectional images. The first one directly computes and compares object shape measures on the sphere and on the viewport, thus before and after rendering, while the second is based on the Tissot indicatrices [96], which are computed for individual objects in the rendered viewport. As depicted in Figure 5.10, the process starts with the semantic segmentation of the omnidirectional image, in equirectangular format (*ERI*), producing a segmentation map denoted as ERI_{seg}. Afterwards, for a required viewport horizontal field-of-view, F_h, spatial resolution (W_{vp}, H_{vp}) and viewing direction (ϕ_{VD}, θ_{VD}), the viewport rendering process is applied to ERI_{seg}, resulting in the viewport segmentation map. For the rendering, the Pannini projection, with parameters (d, vc), is used. The objects distortion is then computed, using the two approaches aforementioned.

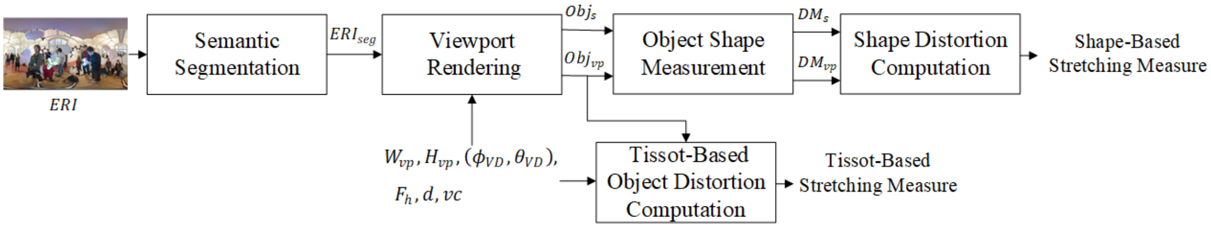
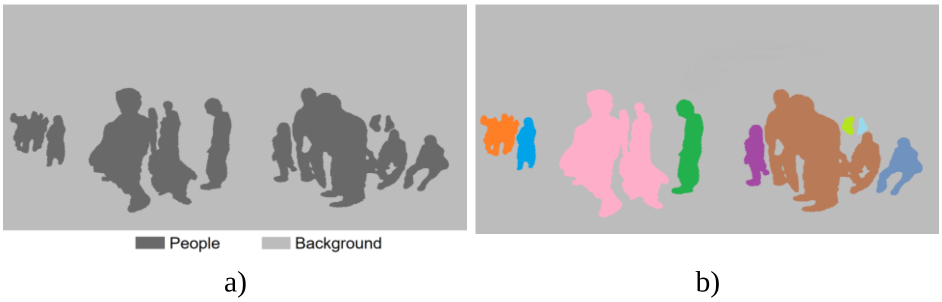

ERI
Semantic Segmentation
ERI_{seg}
Viewport Rendering
Obj_s
Obj_{vp}
Object Shape Measurement
DM_s
DM_{vp}
Shape Distortion Computation
Shape-Based Stretching Measure
$W_{vp}, H_{vp}, (\phi_{VD}, \theta_{VD}),$
F_h, d, vc
Tissot-Based Object Distortion Computation
Tissot-Based Stretching Measure

Figure 5.10. The main architecture of the proposed object-based stretching metrics.

The main steps are described in the following sections.

5.3.1 Semantic Segmentation

Semantic segmentation is a process of assigning a label (e.g., person, car, bicycle, and so on) to objects in the image; in this process, multiple objects of the same class have the same label; it has been used in many computer vision tasks, using 2D images. Although some semantic segmentation models have been developed for omnidirectional images (e.g., [128]–[131]), they were designed for the purpose of autonomous driving, with outdoor images. In this Thesis, to obtain the semantic segmentation of both indoor and outdoor omnidirectional images, the input equirectangular image (*ERI*) is transformed to cubic format, which results in six 2D, rectilinear projected images (the cube faces), with horizontal and vertical FoVs of 90°. The DeepLab semantic segmentation model, proposed in [132], is then applied to each cube face. This model is a deep learning-based approach designed for semantic segmentation of 2D images, that was trained, validated, and tested on several datasets that include indoor and outdoor scenes, and has high accuracy. In this case, it was used the Auto-DeepLab with multi-scale inference and the network backbone *Xception-65*, pretrained on ImageNet [133] and on MS-COCO [134] datasets. The training was performed on the PASCAL VOC 2012 dataset [135], which contains 20 foreground object classes and one background class. As described in [132], the training was performed with a polynomial learning rate with an initial value of 0.05, and a crop size of 513 × 513 pixels. Batch normalization parameters were fine-tuned during training. After obtaining the semantic segmentation for all six cube face images, it is transformed back to equirectangular format. As an example, Figure 5.11a) depicts the semantic segmentation of *Museum* image, using DeepLab. As already mentioned, multiple objects of the same class have the same label. To obtain different labels for disconnected objects, the connected component analysis (CCA) [136], with 4-connectivity, is applied to the segmented *ERI*. Figure 5.11b) depicts the resulting *ERI* segmentation map after CCA, where each disconnected object is represented with a different color.



a)

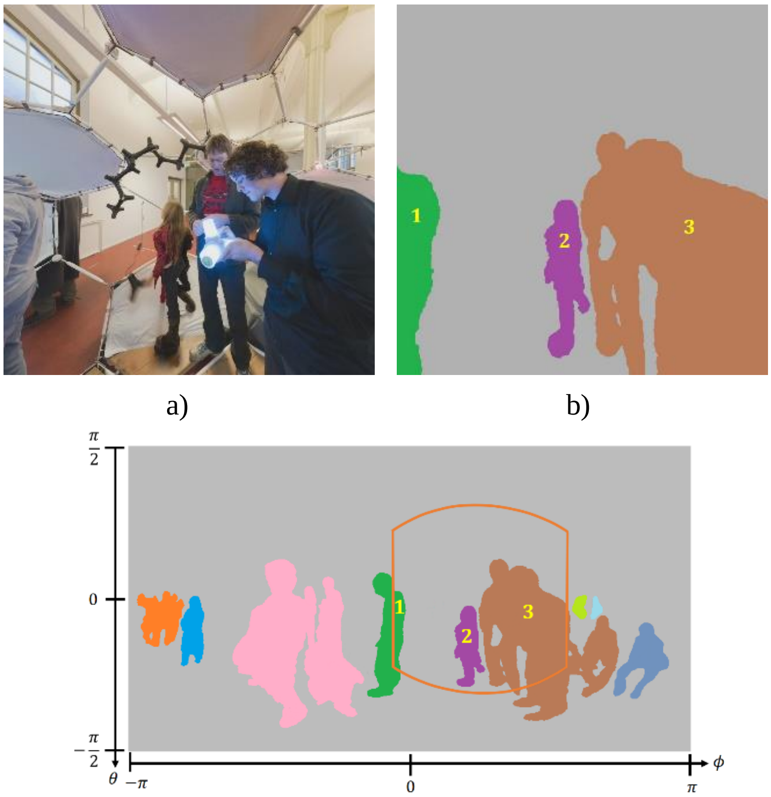
b)

Figure 5.11. a) Semantic segmentation of *Museum*; b) Disconnected objects.

5.3.2 Object Shape Measurement

The object shape distortion measure can be obtained by relating the object shape on the viewing sphere and on the viewport. Several object shape measures have been proposed on the literature [137][138]. Since the sphere to plan projections typically alter the area of the objects, or objects are stretched in the horizontal and/or vertical directions towards the viewport borders (*cf*. Figure 5.5 and Figure 5.8), three shape measures were considered: area, average width and average height. In cartography, these measures showed good performance when used to characterize the distortion of continents and countries for different map projections [139]; this justifies why they were chosen for the study described in this Thesis.

After semantic segmentation of the *ERI* image, it is possible to obtain the semantic segmentation map for any viewport by projecting ERI_{seg}; this allows to obtain the objects in the viewport, Obj_{vp}, linked to the same objects on the viewing sphere, Obj_s. Figure 5.12a) depicts an example of a viewport from *Museum* image and its segmentation map, with three objects (Figure 5.12b), obtained by projecting ERI_{seg}; the same objects are also identified in ERI_{seg} (Figure 5.12c).



a)

b)

c)

Figure 5.12. a) A viewport of *Museum* image; b) Three identified objects in the viewport; c) Corresponding objects in the *ERI*.

The following object shape measures were considered:

- **Object area** - On the sphere, the object area can be computed by summing up the area covered by parallel lines (defined as a sequence of pixels) within the object. At latitude θ, the parallel line area, $PLA_s(\theta)$, contained in an object is given by

$$PLA_s(\theta) = PA_s(\theta) \times N_{ERI}^{PL}(\theta) \tag{5.3}$$

where $PA_s(\theta)$ is the area covered by a pixel at latitude θ, and $N_{ERI}^{PL}(\theta)$ is the total number of pixels within the object at latitude θ. $PA_s(\theta)$ can be approximated by

$$PA_s(\theta) = \Delta\phi \times \Delta\theta \times \cos(\theta) = \frac{2\pi}{W_{ERI}} \times \frac{\pi}{H_{ERI}} \times \cos(\theta) \tag{5.4}$$

where W_{ERI} and H_{ERI} are, respectively, the width and height of the ERI image, in pixels, $\Delta\phi\times\Delta\theta$ is the area covered by a pixel in the ERI image, and $\cos(\theta)$ reflects the decrease in the area (on the sphere) comprised by $\Delta\phi, \Delta\theta$, as θ varies from 0 to ± 90 degrees.

The object area, on the sphere, is computed by

$$OA_s = \sum_{k=1}^{K_{ERI}^{PL}} PLA_s\,(\theta_k) \tag{5.5}$$

where K_{ERI}^{PL} is the total number of parallel lines covered by the object, $k = 1 \ldots K_{ERI}^{PL}$ is the index of those lines, and θ_k is the latitude of the *k*-th parallel line.

The area covered by a pixel on the viewport, PA_{vp}, is given by

$$PA_{vp} = \frac{V_{hs}}{W_{vp}} \times \frac{V_{vs}}{H_{vp}} \tag{5.6}$$

where V_{hs} and V_{vs} are, respectively, the viewport width and height, in length unit. For the Pannini projection, they are given by

$$V_{hs} = 2\ (d+1)\frac{\sin\left(\frac{F_h}{2}\right)}{d + \cos\left(\frac{F_h}{2}\right)} \tag{5.7}$$

$$V_{vs} = 2\ \tan\left(\frac{F_v}{2}\right) \tag{5.8}$$

The object area in the viewport is computed by

$$OA_{vp} = PA_{vp} \times N_{vp}^{obj} \tag{5.9}$$

where N_{vp}^{obj} is the total number of pixels within the object.

- **Object average width** - On the sphere, and at latitude θ, the width of the object, $OW_s(\theta)$, is the length of the parallel line at θ, covered by the object. Since in discrete domain, each parallel corresponds to a line on the ERI image, $OW_s(\theta)$ can be computed as

$$OW_s(\theta) = \frac{2\pi}{W_{ERI}} \times N_{ERI}^{PL}(\theta) \times \cos(\theta) \tag{5.10}$$

where $N_{ERI}^{PL}(\theta)$ is the total number of pixels within the object at latitude θ, and $2\pi/W_{ERI}$ is the width covered by a pixel in the ERI image. The object average width, on the sphere, is computed by

$$OW_s = \frac{1}{K_{ERI}^{PL}} \sum_{k=1}^{K_{ERI}^{PL}} OW_s\,(\theta_k)\,. \tag{5.11}$$

On the viewport, the width of the object at line i can be computed as

$$OW_{vp}(i) = \frac{V_{hs}}{W_{vp}} \times N_{vp}^{l}(i) \tag{5.12}$$

where $N_{vp}^{l}(i)$ is the total number of pixels covered by the object at line i, and V_{hs}/W_{vp} is the width covered by a pixel in the viewport image. The object average width, on the viewport, is given by

$$OW_{vp} = \frac{1}{K_{vp}^{l}} \sum_{i \in \mathrm{Obj}} OW_{vp}(i) \tag{5.13}$$

with the summation applied to the viewport lines covered by the object, and K_{vp}^{l} being the total number of those lines.

- **Object average height** - On the sphere, at longitude ϕ, the object height is the length of the meridian line (ML) at ϕ - which corresponds to a column of the ERI image - covered by the object

$$OH_{s}(\phi) = \frac{\pi}{H_{ERI}} \times N_{ERI}^{ML}(\phi) \tag{5.14}$$

where $N_{ERI}^{ML}(\phi)$ is the total number of pixels within the object at longitude ϕ, and π/H_{ERI} is the height covered by a pixel in the ERI image. The object average height, on the sphere, is given by

$$OH_{s} = \frac{1}{K_{ERI}^{ML}} \sum_{k=1}^{K_{ERI}^{ML}} OH_{s}\,(\phi_{k}) \tag{5.15}$$

where K_{ERI}^{ML} is the total number of meridian lines covered by the object, $k = 1 \ldots K_{ERI}^{ML}$ is the index of those lines, and ϕ_{k} is the longitude of the k-th meridian line.

On the viewport, the height of the object at viewport column j, can be computed as

$$OH_{vp}(j) = \frac{V_{vs}}{H_{vp}} \times N_{vp}^{c}\,(j) \tag{5.16}$$

where $N_{vp}^{c}(j)$ is the total number of pixels covered by the object at column j, and V_{vs}/H_{vp} is the height covered by a pixel in the viewport image. The object average height, on the viewport, is given by

$$OH_{vp} = \frac{1}{K_{vp}^{c}} \sum_{j \in \mathrm{Obj}} OH_{vp}(j) \tag{5.17}$$

with the summation applied to the viewport columns covered by the object, and K_{vp}^{c} being the total number of those lines.

It is important to note that all the shape measurements are in length units and are obtained only for the objects (or parts of the objects) that are rendered on the viewport. As an example, only the parts of objects 1 and 3 that can be seen in Figure 5.12b), were used for the shape measures. Table 5.2 presents the resulting $OA, OW,$ and OH values, on the sphere and on the viewport, for the three objects of Figure 5.12b). All the measures increase after projection, especially for the objects closer to the viewport borders.

Table 5.2. Resulting OA, OW, OH values for three objects on the sphere, before projection, and in the viewport, after projection.

Object / Measure	1	2	3
OA_s	0.067	0.132	0.807
OA_{vp}	0.336	0.220	1.990
OW_s	0.073	0.161	0.605
OW_{vp}	0.215	0.183	1.066
OH_s	0.564	0.470	0.913
OH_{vp}	1.084	0.652	1.386

5.3.3 Shape Distortion Computation

Based on the object shape measures previously presented, the following object shape distortion metrics are defined:

- **Area distortion** - For each object in the viewport, the area distortion is expressed by

$$OAD = |OA_s - OA_{vp}| \tag{5.18}$$

where the OA_s and OA_{vp} are computed by (5.5) and (5.9), respectively.

- **Width distortion** - The object width distortion is given by

$$OWD = |OW_s - OW_{vp}| \tag{5.19}$$

where OW_s and OW_{vp} are given by (5.11) and (5.13), respectively. This measure characterizes the horizontal stretching of the object.

- **Height distortion** - The object height distortion is computed by

$$OHD = |OH_s - OH_{vp}| \tag{5.20}$$

where OH_s and OH_{vp} are computed by (5.15) and (5.17), respectively. This measure characterizes the vertical stretching of the object.

- **Total length distortion** - The total length distortion of an object is defined as

$$OTD = OWD + OHD \tag{5.21}$$

where OWD and OHD are computed by (5.19) and (5.20), respectively.

It is important to mention that, besides the absolute difference expressed by (5.18) to (5.20), the relative difference was also considered, but did not improve the performance of the metric, since it gives more important to the distortion of small objects than to the large ones.

To obtain a global viewport stretching distortion measure, several pooling functions were considered to aggregate the shape distortion measure of all detected objects in the viewport. The considered pooling functions are listed in Table 5.3, where, $\boldsymbol{D}$ is a vector containing one of the distortion measures for all objects in the viewport, and $\boldsymbol{D_p}$ is a vector containing the p% highest elements of $\boldsymbol{D}$; $\boldsymbol{OA_{vp}}$ is a vector containing the object area on the viewport, and $\odot$ denotes element-wise product. Poolings P_1^o and P_2^o assume that the subjective impact of the distortion increases with the number of objects, while pooling P_3^o and P_4^o consider that the impact varies with the average objects distortion; pooling P_5^o presume that the perceptual impact is mainly influenced by the most distorted object, while pooling P_6^o considers the object area in

Table 5.3. Object distortion pooling functions.

$P_1^o = \mathrm{Sum}(\boldsymbol{D})$	(5.22)	$P_2^o = \mathrm{Sum}(\boldsymbol{D_p})$	(5.23)
$P_3^o = \mathrm{Average}(\boldsymbol{D})$	(5.24)	$P_4^o = \mathrm{Average}(\boldsymbol{D_p})$	(5.25)
$P_5^o = \mathrm{Max}(\boldsymbol{D})$	(5.26)	$P_6^o = \frac{\mathrm{Sum}(\boldsymbol{OA_{vp}} \odot \boldsymbol{D})}{\mathrm{Sum}(\mathbf{OA_{vp}})}$	(5.27)

the viewport, giving more emphasis to the distortion of large objects. The reason for the percentile ($p\%$) is to exclude the objects with low distortion values (e.g., the distortion for objects at the viewport center is low and may not be visible); as $p\%$ approaches 100%, P_2^o and P_4^o will be closer to P_5^o; if $p\%$ approaches 0%, P_2^o will be closer to P_1^o and P_4^o will be closer to P_3^o. In summary, considering four shape distortion measures with six pooling functions, results in 24 potential shape-based stretching measures.

5.3.4 Tissot-Based Object Distortion Computation

In the previous chapter, three global viewport Tissot distortion measures - namely, area distortion (G_{darea}^{w}), scale distortion (G_{dscale}^{w}), and angle distortion (G_{dangle}^{w}) - were defined to measure the stretching distortion in the viewport rendering of omnidirectional images; to make these measures content dependent, saliency weights were used. In this Thesis, object based Tissot distortion measures are proposed, and obtained according to the two following steps:

1) **Compute local Tissot distortion metrics** - For a given horizontal and vertical field of view, F_h and F_v, the viewing area on the sphere is defined by $\phi \in [-F_h/2, F_h/2]$ and $\theta \in [-F_v/2, F_v/2]$; this region is then uniformly sampled with a fixed interval $\Delta\phi, \Delta\theta$ (set to 0.05 degrees). For each sampled point, indexed by i, with spherical coordinates (ϕ_i, θ_i), the corresponding Tissot scale factors h_i and k_i , semi-major, $\hat{a}_i$, and semi-minor, $\hat{b}_i$, axis of the Tissot ellipse are obtained. The details about the computation of these parameters were presented in Chapter 4 (Section 4.2.1). To compute these parameters for the PP, the partial derivatives of (x_p, y_p) with respect to (ϕ, θ) need to be computed. From (2.29) and (2.30), it follows:

$$\frac{\partial x_p}{\partial \phi} = \frac{(d+1) \times (d \times \cos(\phi) + 1)}{(d + \cos(\phi))^2} \tag{5.28}$$

$$\frac{\partial x_p}{\partial \theta} = 0 \tag{5.29}$$

$$\frac{\partial y_p}{\partial \phi} = \frac{(1 - vc) \times (d + 1\,) \times \tan(\theta) \times \sin(\phi)}{(d + \cos(\phi))^2} + \frac{vc \times \tan(\theta) \times \sin(\phi)}{\cos^2(\phi)} \tag{5.30}$$

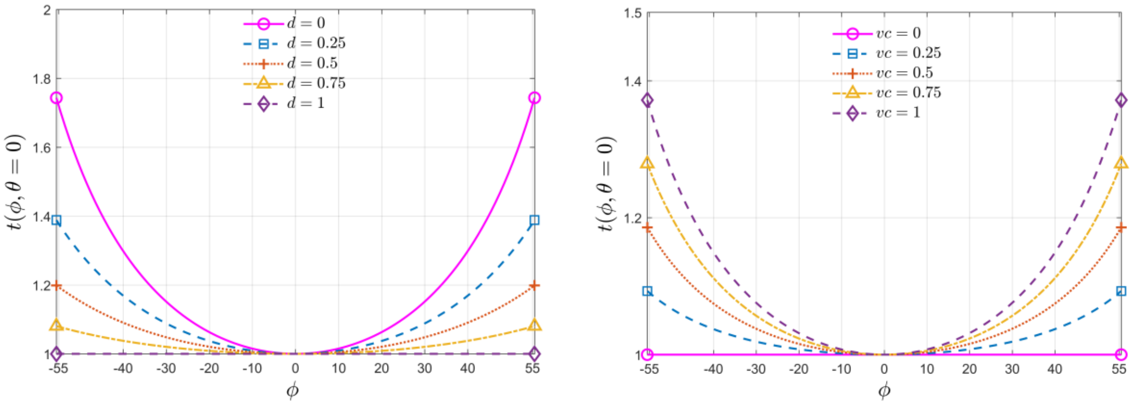
$$\frac{\partial y_p}{\partial \theta} = \frac{1}{\cos^2(\theta)} \times \left[(1 - vc) \times \frac{d+1}{d + \cos(\phi)} + \frac{vc}{\cos(\phi)}\right]. \tag{5.31}$$

Afterwards, the local area distortion, s_i, and local shape distortion, t_i, are computed as

$$s_i = \left(\hat{a}_i \times \hat{b}_i - 1\right) \times \cos\theta_i \tag{5.32}$$

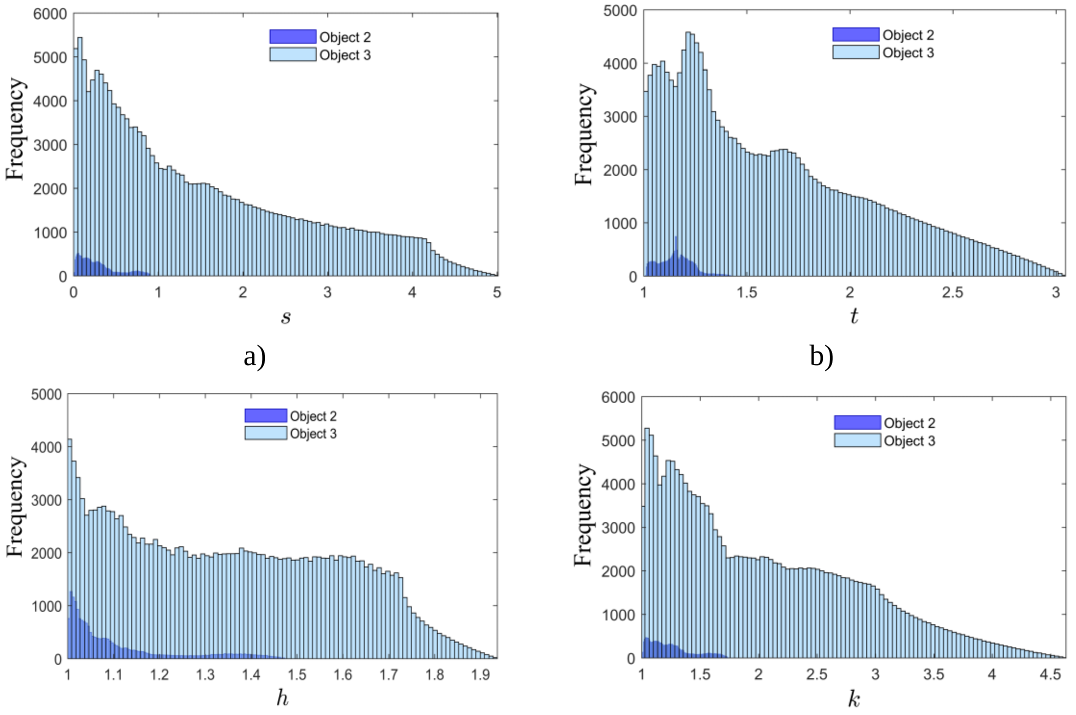
$$t_i = \frac{\hat{a}_i}{\hat{b}_i}\ . \tag{5.33}$$

Although the local angle distortion was also initially considered, it did not improve the results, and was not retained for further assessment.



a) b)

Figure 5.13. The plot of t along the equatorial line $(\theta = 0)$ under the PP for: a) varying d and $vc = 0$; b) varying vc and $d = 1$.



a) b) c) d)

Figure 5.14. Histogram plots of s, t, h, k for two identified viewport objects, Object 2 and Object 3 of Figure 5.12b): a) Local area distortion, s; b) Local shape distortion, t; c) Scale factor, h; d) Scale factors, k.

Figure 5.13 depicts the local shape distortion, t, along the equatorial line $(\theta = 0)$ and $\phi \in [-55^o, 55^o]$, for PP with varying parameters d and vc, one at a time. As can be seen in Figure 5.13a), local shape distortion is maximum for the rectilinear projection $(d = 0)$. On the other hand, the stereographic PP $(d = 1, vc = 0)$, is locally conformal $(t = 1)$, although horizontal lines are bended. In the PP, the bending of horizontal lines can be corrected by applying vc, however shape distortion is introduced, as can be concluded from Figure 5.13b).

Figure 5.14 presents histogram plots of s, t, h, k for two objects of Figure 5.12b), namely *Object* 2 (close to the viewport center and with low distortion), and *Object* 3 (close to the border and with high distortion). In this figure, the frequency represents the number of occurrences of the values in the x-axis. As can be seen, s, t, h, k have a wider range of values (and with higher variance) for *Object* 3, than for *Object* 2.

2) **Compute Tissot-based object distortion metrics** - For each object in the viewport, the following object-based Tissot distortion metrics are obtained:

$$OAD^{TO} = Variance(\boldsymbol{s}) \tag{5.34}$$

$$OSHD^{TO} = Variance(\boldsymbol{t}) \tag{5.35}$$

$$OSD^{TO} = \mathrm{Max}\big(Variance(\boldsymbol{h}), Variance(\boldsymbol{k})\big)\,, \tag{5.36}$$

where OAD^{TO}, $OSHD^{TO}$ and OSD^{TO} are, respectively, the object based Tissot area, shape, and scale distortion metrics. The superscript TO denotes object-based Tissot measure; $\boldsymbol{s}$ and $\boldsymbol{t}$ are vectors containing, respectively, the local area and shape distortions for all points within an object; $\boldsymbol{h}$ and $\boldsymbol{k}$ are vectors containing the scale factors for all points within an object. To obtain a single measure per object, the *Variance* and *Average* functions were considered in (5.34) to (5.36); however, the *Variance* function was selected as it showed the best performance.

To obtain a global viewport stretching distortion measure, the pooling functions presented in Table 5.3 were used. In this case, using three Tissot based object distortion measures, with six pooling functions, results in 18 potential Tissot based stretching distortion metrics.

Figure 5.15 presents the resulting stretching distortion values for a sub-set of the proposed object based stretching measures with pooling function P_6^o, and the three stretching measures - G_{darea}^{w}, G_{dscale}^{w}, and G_{dangle}^{w} - proposed in Chapter 4 (Section 4.4.1.B), computed for a pair of viewports, *Museum-*VP_1 and *Museum-*VP_2 (depicted in Figure 5.8e) and Figure 5.8f); the blue and orange bars correspond, respectively, to *Museum-*VP_1 and to *Museum-*VP_2. As can be figured out, the proposed object based stretching measures allow a higher discrimination between the quality of the two viewport images, than the metrics proposed in Chapter 4, since for the former the difference between blue and orange bars are much more evident.

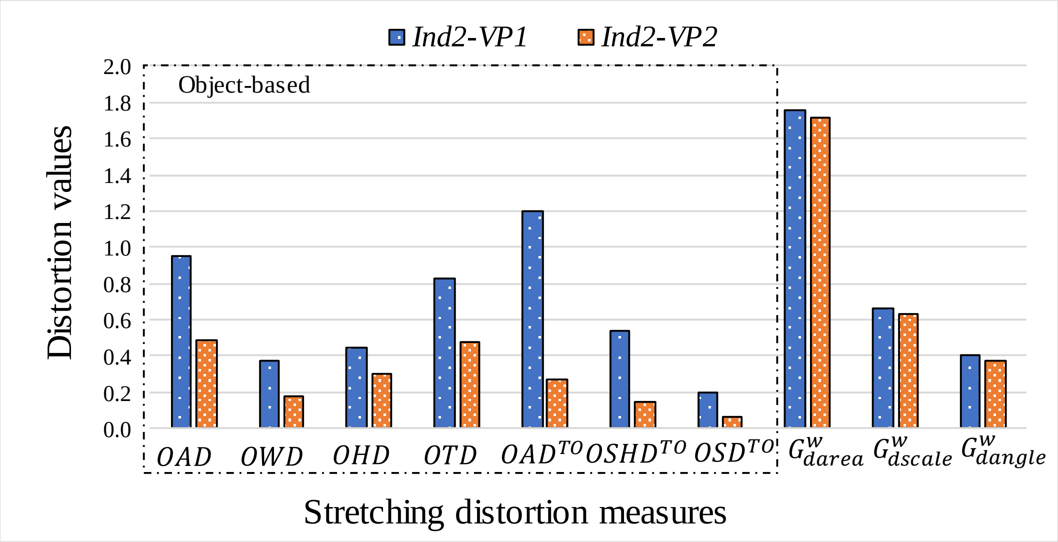


Figure 5.15. Stretching distortion values for the proposed object-based measures, and for the stretching measures proposed in Chapter 4 (Section 4.4.1.B), computed for a pair of viewports, *Museum-*VP_1 and *Museum-*VP_2, presented respectively in Figure 5.8e) and Figure 5.8f).

5.4 Metrics Performance Evaluation

In this section, the proposed object-based stretching distortion metrics are evaluated and compared to benchmark solutions. The usual measure to evaluate an objective quality metric is the correlation (Pearson and/or Spearman) between objective scores and ground truth opinion scores (typically, MOS or DMOS). However, since in this work a pairwise comparison (PC) method was used on the subjective tests, it is not possible to obtain MOS or DMOS values for each individual stimulus. Accordingly, the proposed metrics are assessed versus the subjective scores using the classification errors approach, as suggested in Rec. ITU-T J.149 [72], and applied in related literature [126][140].

5.4.1 Classification Errors

According to Rec. ITU-T J.149 [72], a classification error (CE) occurs when the objective and subjective scores lead to different conclusions about the relative quality of a pair of stimuli, VP_i and VP_j. Three types of errors may happen:

- **False Tie (FT)** - when the subjective score indicates that VP_i and VP_j are different, but the objective score indicates that they are similar.
- **False Differentiation (FD)** - when the subjective score indicates that VP_i and VP_j are similar, but the objective score indicates that they are different.
- **False Ranking (FR)** - when the subjective score indicates that VP_i (VP_j) is better than VP_j (VP_i), but the objective score indicates the opposite.

Let ΔOM represent the minimum difference, between the objective quality scores of two stimuli, that defines when the two stimuli become perceptually distinguishable. As ΔOM increases, more stimuli pairs are considered similar, increasing the occurrence of FT, but the occurrences of FD and FR will decrease. On the contrary, as ΔOM decreases, the occurrence of FT also decreases, but the occurrence of FD and FR will increase. Following ITU-T J.149, the percentage of each error type and of correct decisions are obtained from the considered stimuli pairs as a function of ΔOM, for individual metrics; this allows to compare the metrics and determine the best one for the application under analysis. The best ΔOM value is the one that maximizes the correct decision percentage [72][140].

5.4.2 Experimental Results and Analysis

To evaluate the proposed distortion metrics, the viewport dataset described in Section 5.2.1, and the processed PC scores after outlier's removal, as described in Section 5.2.3, were used. Moreover, the performance of the metrics were compared to the following benchmark solutions: *area distortion* (G_{darea}^{w}), *scale distortion* (G_{dscale}^{w}), and *angle distortion* $\left(G_{dangle}^{w}\right)$, proposed in Chapter 4 (Section 4.4.1.B), the *conformality* measure (CM) proposed in [36], and the *content-dependent conformality* (CM^{sal}); the latter is a modified CM, by integrating the viewport saliency on it. For poolings P_2^o, P_4^o, several values of $p\%$ were considered, and the resulting classification errors and correct decision were obtained. The best performance was obtained for $p = 50\%$.

Table 5.4 reports the classification errors and correct decision values for each proposed distortion measure, using the pooling functions described in Section 5.3.3, and for the benchmark solutions; the ΔOM value that maximized the correct decision percentage was used. As can be figured out, there is a significant performance improvement for the object-based metrics, when compared with the benchmark solutions. Among the proposed metrics, the object-based Tissot metrics achieved the highest Correct Decision and the lowest False Tie percentages. Among the benchmark metrics, the CM has the worst performance. This metric is content independent, and the same metric value is obtained for any viewport image. When the conformality integrates the saliency, as in CM^{sal}, the performance increases, confirming that it brings some additional value to the metric. To find out the best solution among the proposed ones, the true positive rate (TPR), defined by (5.37), was computed

$$TPR = \frac{\mathrm{CD}}{\mathrm{CD} + \mathrm{FT} + \mathrm{FD} + \mathrm{FR}}. \tag{5.37}$$

Table 5.4. Classification errors and correct decision values, in percentage (%), for the proposed and benchmark metrics.

Correct Decision (%)

$\boldsymbol{P_i^o}$ **Metric**	P_1^o	P_2^o	P_3^o	P_4^o	P_5^o	P_6^o
OAD	80.0	84.4	82.2	82.2	84.4	82.2
OWD	64.4	71.1	82.2	75.6	73.3	80.0
OHD	71.1	68.9	82.2	80.0	71.1	77.8
OTD	68.9	68.9	84.4	86.7	71.1	82.2
OSD^{TO}	84.4	84.4	86.7	86.7	84.4	84.4
OAD^{TO}	84.4	82.2	86.7	86.7	86.7	88.9
$OSHD^{TO}$	82.2	84.4	84.4	84.4	84.4	82.2

False Tie (%)

$\boldsymbol{P_i^o}$ **Metric**	P_1^o	P_2^o	P_3^o	P_4^o	P_5^o	P_6^o
OAD	2.2	0.0	0.0	0.0	2.2	4.4
OWD	0.0	2.2	15.6	24.4	4.4	0.0
OHD	2.2	2.2	0.0	2.2	0.0	6.7
OTD	0.0	4.4	2.2	2.2	2.2	8.9
OSD^{TO}	0.0	0.0	0.0	2.2	0.0	2.2
OAD^{TO}	0.0	0.0	0.0	0.0	0.0	0.0
$OSHD^{TO}$	0.0	0.0	0.0	0.0	0.0	2.2

False Differentiation (%)

$\boldsymbol{P_i^o}$ **Metric**	P_1^o	P_2^o	P_3^o	P_4^o	P_5^o	P_6^o
OAD	11.1	8.9	13.3	15.6	8.9	8.9
OWD	17.8	15.6	2.2	0.0	13.3	15.6
OHD	13.3	13.3	13.3	11.1	13.3	6.7
OTD	17.8	13.3	8.9	6.7	13.3	6.7
OSD^{TO}	11.1	8.9	8.9	8.9	13.3	11.1
OAD^{TO}	13.3	13.3	13.3	13.3	13.3	11.1
$OSHD^{TO}$	11.1	11.1	8.9	11.1	11.1	11.1

False Ranking (%)

$\boldsymbol{P_i^o}$ **Metric**	P_1^o	P_2^o	P_3^o	P_4^o	P_5^o	P_6^o
OAD	6.7	6.7	4.4	2.2	4.4	4.4
OWD	17.8	11.1	0.0	0.0	8.9	4.4
OHD	13.3	15.6	4.4	6.7	15.6	8.9
OTD	13.3	13.3	4.4	4.4	13.3	2.2
OSD^{TO}	4.4	6.7	4.4	2.2	2.2	2.2
OAD^{TO}	2.2	4.4	0.0	0.0	0.0	0.0
$OSHD^{TO}$	6.7	4.4	6.7	4.4	4.4	4.4

Benchmark Metrics

Metric	Correct Decision	False Tie	False Differentiation	False Ranking
G_{darea}^{w}	68.9	0.0	17.8	13.3
G_{dscale}^{w}	68.9	0.0	15.6	15.6
G_{dangle}^{w}	66.7	17.8	6.7	8.9
CM^{sal}	66.7	0.0	15.5	17.8
CM	42.2	53.3	4.4	0.0

Table 5.5. Selected pooling function for each metric and corresponding TPR values.

Metric	OAD	OWD	OHD	OTD	OSD^{TO}	OAD^{TO}	$OSHD^{TO}$
Pooling	P_5^o	P_3^o	P_3^o	P_4^o	P_4^o	$\boldsymbol{P_6^o}$	P_5^o
$\boldsymbol{TPR}$	0.84	0.82	0.82	0.87	0.87	**0.89**	0.84

Table 5.6. Classification errors and correct decision values, in percentage (%), for the best proposed metric, $\boldsymbol{OAD^{TO}}$ with $\boldsymbol{P_6^o}$, considering, or not, the background distortion.

	Not Considering the background distortion	**Considering the background distortion**
Correct Decision	88.9	71.1
False Tie	0.0	2.2
False Differentiation	11.1	15.6
False Ranking	0.0	11.1

Table 5.5 presents the best pooling function for each metric and the resulting TPR values. Since the metric OAD^{TO} with P_6^o has the highest TPR value, it was the selected one. For the benchmark metrics, the TPR have values of 0.69, 0.69, 0.67, 0.67, and 0.42, for G_{darea}^w, G_{dscale}^w, G_{dangle}^w, CM^{sal}, and CM, respectively. These values are much lower than the TPR value of 0.89 obtained for OAD^{TO} with P_6^o, being also lower than the TPR values obtained for the other proposed metrics. Taking into account the evaluation results of the different metrics, the object-based Tissot area distortion (OAD^{TO}), with pooling function P_6^o, is then the proposed one to assess the subjective impact of the viewport stretching distortion, in omnidirectional image rendering.

Figure 5.16 depicts the plots of classification errors and correct decision for the selected metric, OAD^{TO} with P_6^o, and for the benchmark metric G_{darea}^w, where the dashed line indicates the ΔOM value that maximizes the correct decision. For OAD^{TO} with P_6^o, and with $\Delta\text{OM} = 0$ (all stimuli pairs are considered as perceptually different by the objective metric), the correct decision percentage is 82%, which agrees with the results of the subjective test, where the difference was statistically significant in 82% of the pairs (*cf.* Figure 5.7); on the other hand, the correct decision percentage is just 68.9% for G_{darea}^w. Also, for $\Delta\text{OM} = 0$, the false differentiation percentage is 18% and 17.8% for OAD^{TO} with P_6^o and G_{darea}^w respectively; however, the false ranking percentage is 0% for OAD^{TO} with P_6^o and 13.3% for G_{darea}^w.

To evaluate the impact, on the metric performance, of considering or not the background, the background distortion was computed using the selected metric (OAD^{TO} with P_6^o), and included in the metric as an additional measure; after, the classification errors and correct decision values with/without considering the background distortion were compared. Table 5.6 presents the resulting classification errors and correct decision values, showing that the metric performance decreases when the background distortion is included. This is consistent with fact that the stretching in the background is not as visible as the stretching of foreground objects, and shows the advantage of having an object-based stretching metric.

5.5 Final Remarks

In this chapter, a novel object-based quality metric to assess the subjective impact of the objects shape deformation in viewport images, rendered from omnidirectional images, was proposed.

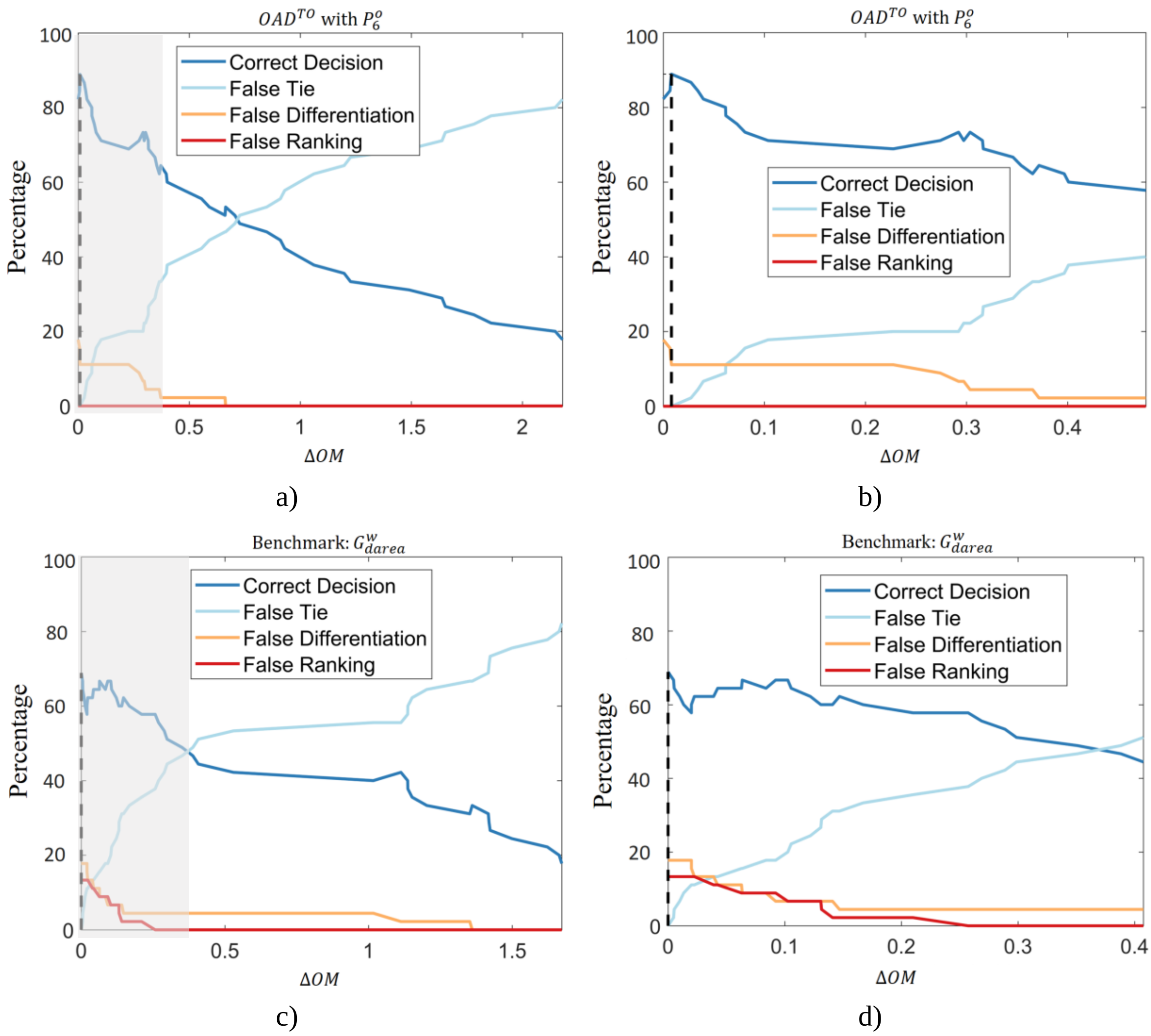


Figure 5.16. Plots of classification errors and correct decision for the best proposed metric and for the benchmark metric G^{w}_{darea}. For a better visualization, the shaded area on the left side plots are represented in the right-side plot, using a larger scale.

The metric uses semantic segmentation to identify the relevant objects in the viewport, where the stretching distortion has a higher perceptual impact, and computes the stretching distortion for each object. Two distinct approaches were exploited and evaluated: the first one, directly computes and compares object shape measures on the sphere and on the viewport; the second one is based on Tissot indicatrices, which are computed for individual objects in the viewport. The experimental results show that while the Tissot based method performs slightly better than the direct shape measurement, both approaches outperform benchmark solutions; furthermore, they were able to classify the viewport quality, with respect to quality scores obtained in a subjective crowdsourcing study, with a correct decision percentage close to 90%.

The next chapter describes a useful application for the proposed metric, where it is integrated in a procedure to globally optimize the Pannini projection parameters, according to the viewport content.

The work presented in this chapter has been included in the following journal paper:

- **F. Jabar**, J. Ascenso, and M.P. Queluz, “Object-Based Geometric Distortion Metric for Viewport Rendering of 360° Images”, *IEEE Access*, vol.10, no.1, 13827-13843, Jan. 2022.

Chapter 6

Pannini Projection Optimization

6.1 Introduction

As mentioned in Chapter 1, the most often used perspective projections for viewport rendering (i.e., rectilinear, and stereographic) are content-unaware. Therefore, in Chapter 4 a content-aware general perspective projection (CA-GPP) was proposed. Although the CA-GPP allows to obtain visually pleasant viewport images for FoVs (~ 110°), the geometric distortion becomes quite visible and annoying for higher FoVs; however, larger FoVs offer a higher user's sense of immersion and presence. Also, as seen in Chapter 2, current state-of-the-art content-aware projections [10][36][37][43][54]–[56] are not able to provide viewports with large FoVs (~150°), without noticeable distortions. Excluding [36][54][55], these projections are globally or regionally adapted to the content, but are not able to provide a good balance between bending and stretching, while others are not fully automatic and require user interaction [36][43][54], or are used to only to reduce the geometric distortions for human faces without considering general objects that may appear in the scene [55].

This chapter addresses the rendering of viewports with large FoVs (~150°), targeting new content adapted sphere to plan projections, with reduced geometric distortions comparatively to state-of-the-art methods. The solutions here proposed are built over the Pannini projection, due to its good performance compared to other content-unaware projections, notably its suitability for viewport rendering with high FoV values. In this context, this chapter has the following main objectives:

- To globally adapt the Pannini projection parameters (d and vc) to the viewport content, aiming to minimize the geometric distortions with a single set of parameters (similarly to what was done in CA-GPP).
- To further improve the conformality of semantically relevant objects, by locally optimizing the Pannini projection parameters resulting from the previous step.

The rest of this chapter is organized as follows. Section 6.2 describes the global optimization of the Pannini projection parameters. Section 6.3 proposes a two-step procedure where the Pannini parameters are firstly globally optimized, followed by a local conformality improvement of relevant viewport objects. Section 6.4 presents the evaluation of the proposed Pannini projection optimization. Section 6.5 finalizes this chapter with some final remarks.

6.2 Globally Adapted Pannini Projection

This section describes the global adaptation of the Pannini projection to the viewport content; it consists in obtaining the optimal - in the perceived quality sense - projection parameters for the viewport rendering of omnidirectional images, resulting in the globally adapted Pannini projection (GA-PP).

As shown in Chapters 1 and 2, the stretching of objects and the bending of straight lines are the two main artifacts that condition the perceived geometric distortion of the rendered viewports. Furthermore, they have an opposite evolution with the variation of the projection parameters, i.e., stretching decreases and bending increases when d varies from 0 to 1, and/or vc varies from 1 to 0. Thus, the procedure to find the optimal parameters, (d_{opt}, vc_{opt}), seeks the best compromise between these two types of artifacts.

6.2.1 Methodology

In Chapter 4, two solutions were proposed to optimize the general perspective projection, one based on SVR and another based on a simple cost function. In this work, the cost function-based solution was chosen to optimize the Pannini projection parameters. In fact, since this projection has two parameters, a large number of viewports and associated ground truth quality scores would be required for training the SVR model, which will be very time consuming

The proposed GA-PP framework is illustrated in Figure 6.1. For a given input equirectangular (*ERI*) image, viewing direction (ϕ_{VD}, θ_{VD}), and horizontal FoV, F_h, the resulting viewport stretching and bending metrics are iteratively computed for different combinations of d and vc values, varying d between d_{min} and d_{max}, with a step-size Δd, and vc between vc_{min} and vc_{max}, with a step-size Δvc. For the results presented in this Thesis, $d_{min} = 0.1, d_{max} = 1$, $vc_{min} = 0$ and $vc_{max} = 1$; Δd and Δvc were both set to 0.1, resulting in N=110 possible (d, vc) pairs. In Figure 6.1, $\boldsymbol{L}$ is the set of detected lines, after line merging and filtering; $\boldsymbol{K}$ is a 2D matrix with size $N \times 2$, which contains the (d, vc) pairs, indexed by i, $i = 1,2, \dots, N$. The projection with $(d = 0, vc = 0)$ - pure rectilinear projection - was not considered in the d, vc set, since it results in too much stretching in the viewport. The optimum parameters, (d_{est}, vc_{est}), are obtained by minimizing, over the considered (d, vc) pairs, a simple cost function similar to (4.39).

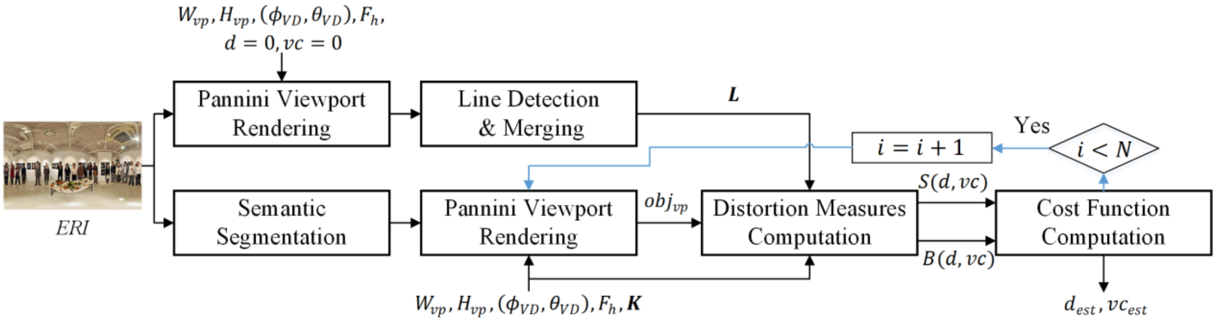

$W_{vp}, H_{vp}, (\phi_{VD}, \theta_{VD}), F_h,$
$d = 0, vc = 0$
Pannini Viewport Rendering
Line Detection & Merging
$\boldsymbol{L}$
Yes
$i = i + 1$
$i < N$
$S(d, vc)$
ERI
Semantic Segmentation
Pannini Viewport Rendering
obj_{vp}
Distortion Measures Computation
Cost Function Computation
$B(d, vc)$
$W_{vp}, H_{vp}, (\phi_{VD}, \theta_{VD}), F_h, \boldsymbol{K}$
d_{est}, vc_{est}

Figure 6.1. Proposed GA-PP framework.

Every procedure in Figure 6.1 has been already introduced previously, except the two last steps – Distortion Measure Computation and Cost Function Computation – which are detailed herein:

- **Distortion Measures Computation** - For the stretching distortion measure, the best proposed Tissot object-based stretching measure - OAD^{TO} with pooling function P_6^o - detailed in the previous chapter (Section 5.3.4) was considered. Note that the OAD^{TO} requires the omnidirectional image semantic segmentation, which justifies the need for the segmentation block in Figure 6.1. For the bending distortion measure, one of the best bending measures - Line Measure Combination LMC, with line pooling function P_5^l - proposed in Chapter 4 (Section 4.3) was selected.

- **Cost Function Computation** - The optimum parameters, (d_{est}, vc_{est}), to be used on the final viewport rendering, are obtained by minimizing, over the considered (d, vc) pairs, a simple cost function, described by:

$$(d_{est}, vc_{est}) = \min_{(d,vc)} \left(\alpha \left[\frac{S(d, vc) - S_{min}}{S_{max} - S_{min}} \right] + \left[\frac{B(d, vc) - B_{min}}{B_{max} - B_{min}} \right] \right) \tag{6.1}$$

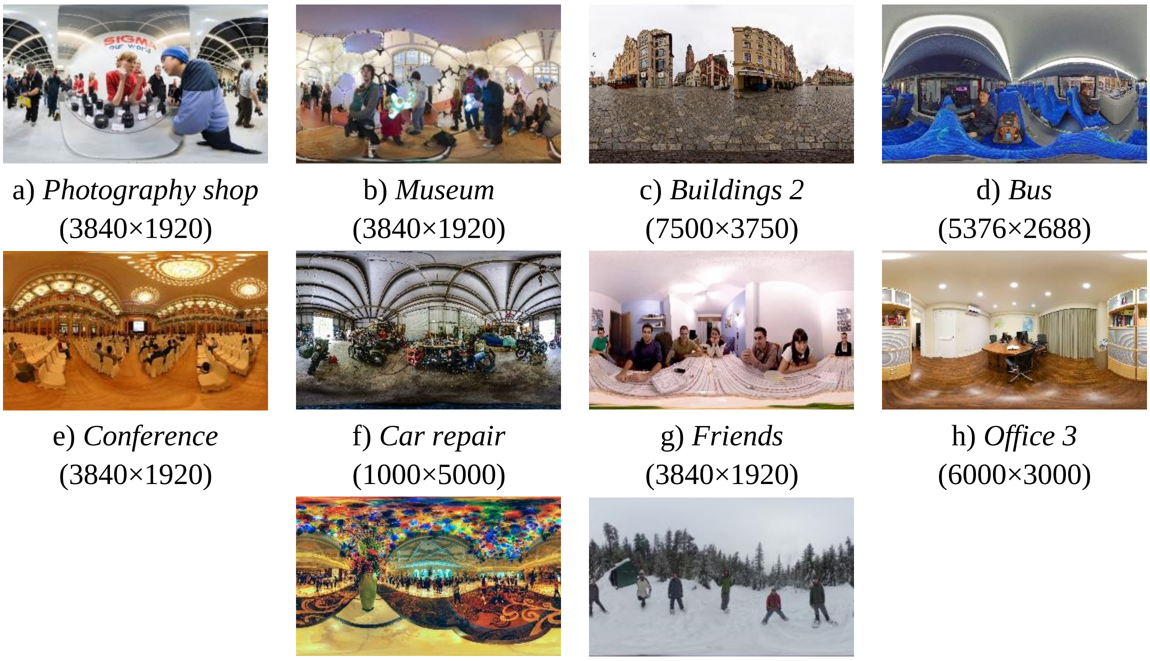
  where $S(d, vc)$ and $B(d, vc)$ are, respectively, the viewport stretching and bending measures for projection parameters (d, vc); α is the stretching to bending ratio; $S_{min}, S_{max}, B_{min}$, and B_{max} are normalizing constants guaranteeing that the metric values are on the interval [0,1]. The normalization constants correspond to the minimum and maximum S and B values that were found for a set of 2200 viewports, rendered from 20 omnidirectional images, and using (d, vc) values on the intervals previously specified.

In (6.1), parameter α seeks the best balance between stretching and bending subjective impact, and it was learned in a perceptual way using a small data set of Pannini viewports, not contained in the final evaluation dataset. The details about the procedure to obtain this parameter are provided in the next section.

6.2.2 Cost Function Parameter Selection

As mentioned above, parameter α used in (6.1) was obtained in a perceptual way. For that, a short subjective assessment session was conducted with just three observers (the author of this Thesis and his supervisors), to build a Pannini viewport dataset with associated ground-truth (GT) optimal projection parameters, (d_{gt}, vc_{gt}). The considered dataset, subjective test methodology, and resulting GT projection parameters, are detailed herein:

- **Dataset** - Ten omnidirectional images in equirectangular format (*ERI*), taken from [10][52], were used in the subjective assessment. The images and their resolutions are depicted in Figure 6.2. This set of images includes six images that were already used in previous subjective tests, and four new images, namely *Bus*, *Car repair*, *Office 3, Exhibition*. The new images were included to have contents with/without people, and different types of dominant distortion (from stretching to bending) when the Pannini projection is used with different values of d and vc. For each omnidirectional image, one viewing direction was considered, which was selected with the procedure described in the previous chapter (Section 5.2.1). For each viewing direction, 25 viewports were rendered, corresponding to the possible combinations of d and vc values, varying d between d_{min} and d_{max}, with a step-size Δd, and vc between vc_{min} and vc_{max}, with a step-size Δvc, where $d_{min} = 0.2$, $d_{max} = 1, vc_{min} = 0$ and $vc_{max} = 1$. In (6.1), Δd and Δvc were both set to 0.1; however, to reduce the number of comparisons and thus limit the test duration less than half an hour, in this test Δd was set to 0.2 and Δvc was set to 0.25. Moreover, since there were only three



a) *Photography shop* (3840×1920)

b) *Museum* (3840×1920)

c) *Buildings 2* (7500×3750)

d) *Bus* (5376×2688)

e) *Conference* (3840×1920)

f) *Car repair* (1000×5000)

g) *Friends* (3840×1920)

h) *Office 3* (6000×3000)

i) *Exhibition* (13320×6660)

j) *Snow* (5376×2688)

Figure 6.2. Omnidirectional images used in the subjective tests, and their spatial resolution.

observers, having a higher granularity for d and vc increases the chance of having similar choices between the observers. The rendered viewports had a spatial resolution of 960×540 pixels ($AR = 16/9$)) and a F_h of 150°. This resolution allows the simultaneously display of two viewports, side by side, in typical monitors.

- **Subjective Evaluation Method** - The objective of the test was to find out, for each omnidirectional image and considered viewing direction, the Pannini parameters, the projection parameters that resulted in the most pleasant viewport. As on the subjective test described in the previous chapter, the pairwise comparison (PC) method was selected for the subjective evaluation. The subjective assessment interface was similar to the one designed in the previous chapter, except that the option “A=B” was excluded from the grading scale. For each omnidirectional image, two viewports were shown side by side (each rendered with a different (d, vc) pair) and the observers were asked to select the best one, in his opinion. The selected viewport remained on the screen and the next viewport, from the same omnidirectional image, was then shown; this procedure was repeat for all (d, vc) pairs. The projection parameters of the last selected viewport were then considered as the best ones for that image and viewing direction. Each viewport was shown in random order and position. The subjective test was conducted with a 2D display, using a Full HD monitor, with a native resolution of 1920×1080 pixels.

- **GT Pannini Viewport** - For each omnidirectional image, a pair of GT parameters, (d_{gt}, vc_{gt}), was obtained. Since parameters selection was quite similar among the observers and there were at least two observers selected the same viewport, it was possible to obtain the (d_{gt}, vc_{gt}) for each omnidirectional image by counting the number of votes. The ten resulting GT parameters, (d_{gt}, vc_{gt}), and corresponding viewports, were made available in [125].

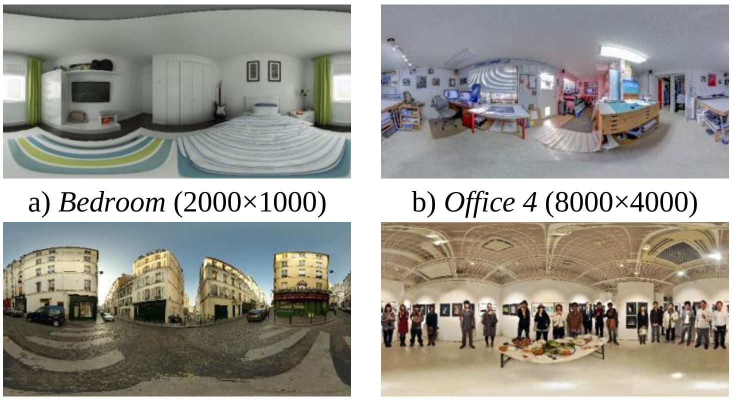
For each viewport in the resulting dataset, the optimum projection parameters were predicted using (6.1), and varying α between 0 and 10 with an increment of 0.01. For each α, the training error, $TE(\ \alpha)$, given by:

$$TE(\alpha) = \frac{1}{N_t}\sum_{i=1}^{N_t} \frac{\left|d_{est}^{i}(\alpha) - d_{gt}^{i}\right| + \left|vc_{est}^{i}(\alpha) - vc_{gt}^{i}\right|}{2}, \tag{6.2}$$

was computed and stored. In (6.2), i is the viewport index, N_t is the number of viewports in the dataset (in this case, $N_t = 10$), $(d_{est}^{i}(\alpha), vc_{est}^{i}(\alpha))$ are the predicted optimum parameters for viewport *i* when α is used in (6.1), and $(d_{gt}^{i}, vc_{gt}^{i})$ are the GT parameters for viewport i. The α value resulting in the lowest training error was 0.24 and was the selected value.

6.2.3 GA-PP Qualitative Evaluation

The proposed GA-PP projection was compared with several benchmark projections that include rectilinear, stereographic, two Pannini with fixed parameters - $(d = 0.5, vc = 0)$ and $(d = 1, vc = 0)$ - and the globally adapted Pannini (OP) projection proposed in [10]. For comparison purposes, four viewports were rendered from four different omnidirectional images available in the datasets of [10][52]. The equirectangular images and their spatial resolution are presented in Figure 6.3. The viewports were rendered with a horizontal FoV, F_h, of 150° and a spatial resolution of 960×540 pixels (aspect ratio, $AR = 16/9$), as in [10].



a) *Bedroom* (2000×1000) b) *Office 4* (8000×4000)

c) *Buildings 1* (7500×3750) d) *Dinner 2* (7500×3750)

Figure 6.3. Omnidirectional images and their spatial resolution used for producing viewports with different projections.

Figure 6.4 depicts the viewports obtained with the proposed GA-PP and with the considered benchmark projections; the OP viewports were obtained from the authors of [10]. As can be figured out, the GA-PP viewports are generally more pleasant, providing a good compromise between bending and stretching distortions; in particular, the following qualitative comparisons can be made:

- **GA-PP *vs* rectilinear and stereographic -** The viewports resulting from GA-PP are clearly more pleasant than those resulting from rectilinear and stereographic projections. While the lines are straight in the rectilinear viewports, the perspective effect is very strong and annoying, and the object shapes are too much stretched, notably for the *Office 4*, *Buildings 1*, and *Dinner 2* viewports. Although the objects shape is preserved in the stereographic viewports, the lines are severely bent (fisheye effect).

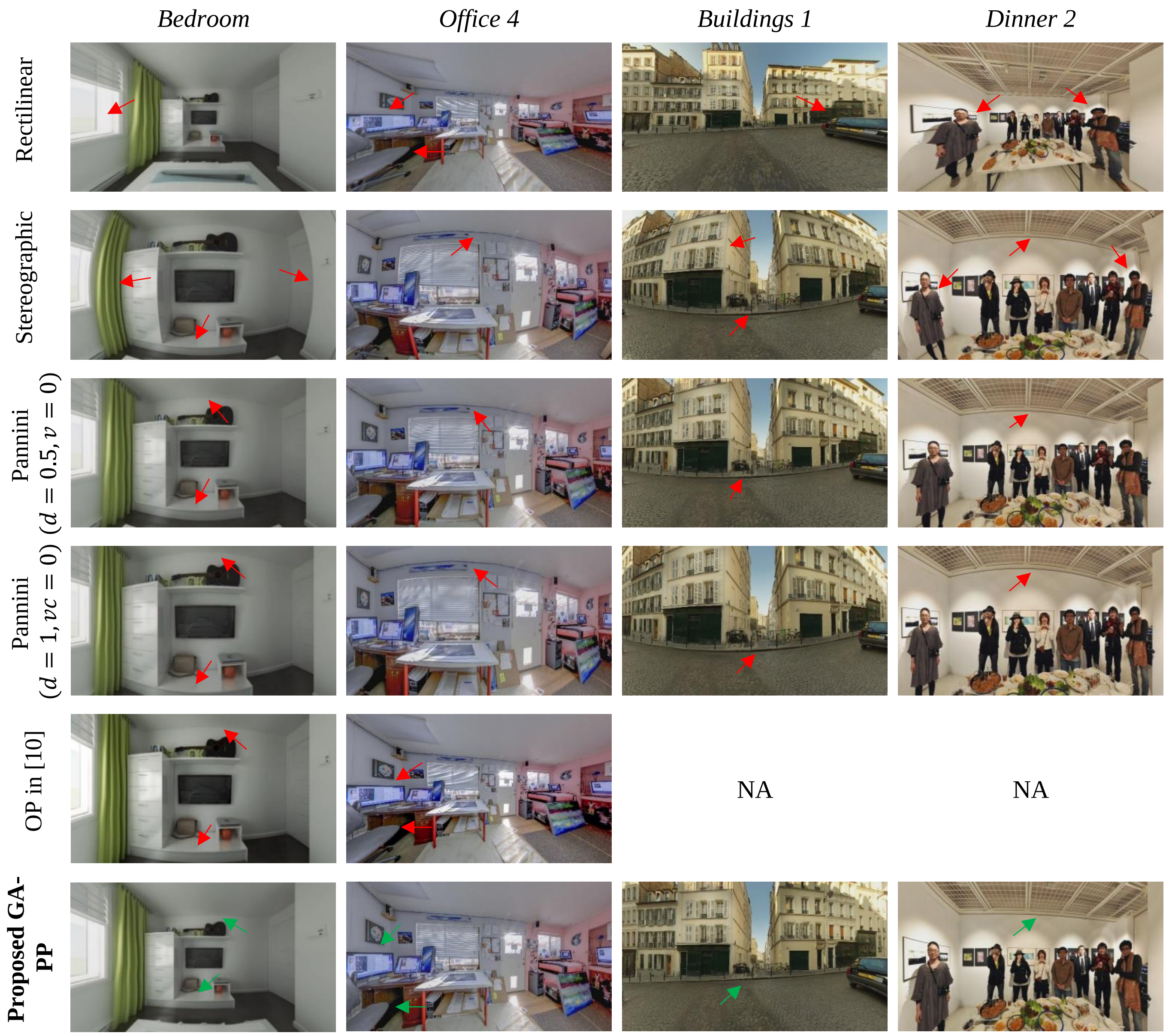


Figure 6.4. Example of viewports rendered with different projections and using a horizontal FoV of 150°. NA corresponds to Not Available. The red and green arrows indicate, respectively, objects/regions with high and low geometric distortions.

- **GA-PP *vs* Pannini with fixed parameters -** The proposed GA-PP generates viewports with a good balance between the stretching of objects and bending of lines. This cannot be achieved for Pannini with fixed parameters, as for $vc = 0$ the horizontal lines are rather bent, particularly for $d = 1$.

- **GA-PP *vs* OP -** The viewports obtained for GA-PP have less global geometric distortion than the viewports resulting from OP. In particular, for the *Bedroom* viewport, the horizontal lines on the ceiling and on the floor are straighter for GA-PP. In the *Office 4* viewport, the GA-PP kept the horizontal lines as straight as OP, but the objects shape (e.g., the chair and monitor on the left side) is more conformal for GA-PP.

An additional evaluation of the GA-PP, within a crowdsourcing subjective assessment test, is provided in Section 6.4.

The previous analysis showed that the GA-PP results in viewports with a more pleasant visual quality than the considered benchmark solutions. However, since this projection is globally adapted to the viewport content (i.e., d and vc have the same values for the whole viewport), stretching and/or bending may be still visible for some image regions and structures; as an example, in Figure 6.4 the lady on the left side of the *Dinner 2* viewport resulting from GA-PP is stretched in the vertical direction. If the projection parameters are allowed to vary locally, the geometric distortions could be further reduced. This possibility is exploited in the next section, where a procedure to globally and locally optimize the Pannini projection is proposed.

6.3 Globally and Locally Adapted Pannini Projection

In this section, a globally and locally adapted Pannini projection (GLA-PP) is proposed. In this projection, a two-step procedure was conceived to minimize the geometric distortions, first globally, based on the optimization of the Pannini projection parameters (as in GA-PP), and then locally for some regions using a content-aware mesh optimization, to improve the conformality of perceptually relevant viewport objects.

6.3.1 Methodology

Figure 6.5 depicts the GLA-PP framework. To minimize the viewport distortions when high FoVs are used, this projection is globally and regionally adapted to the viewport content, according to two optimization steps:

1) **Global optimization** - The Pannini projection is optimized considering the whole viewport, resulting in the projection parameters (d_b,vc_b) that present the best compromise between stretching and bending. Due to the higher visual impact of lines bending, the optimization procedure gives more importance to this distortion.

2) **Local optimization** - The projection resulting from the previous step is further improved for relevant objects. This is obtained by defining two meshes, M_b and M_f, on the viewport plane, that are iteratively combined in one optimized mesh, M_o, as suggested in [55]. While M_b corresponds to the globally optimized projection, M_f corresponds to a conformal (or quasi conformal) projection. The goal is to increase the conformality of the foreground objects, using M_f, while assuring a seamless transition to M_b, which is mainly applied over the background.

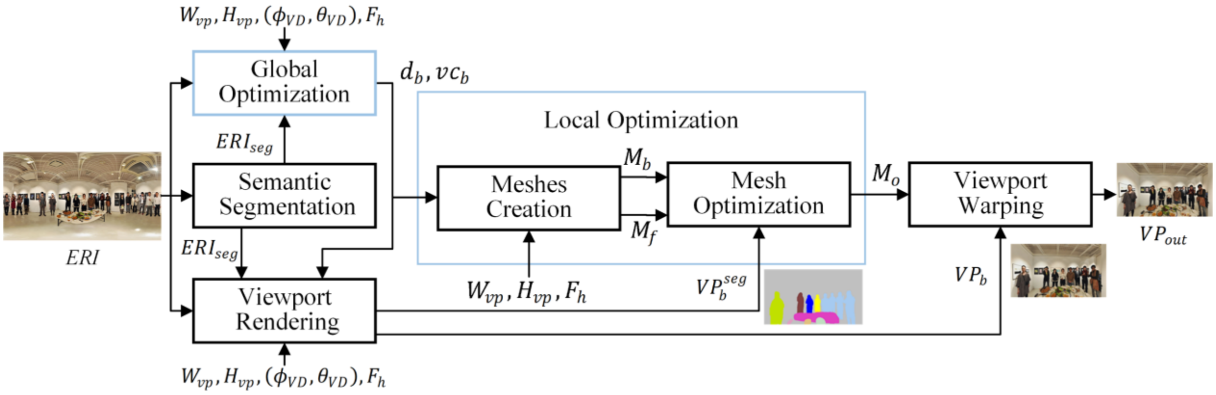
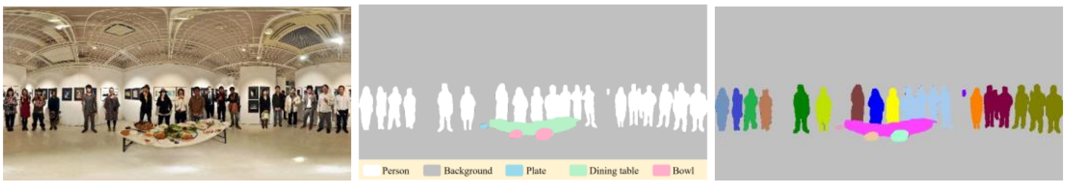


Figure 6.5. Globally and locally adapted Pannini (GLA-PP) projection framework.



a) b) c)

Figure 6.6. a) Example of an ERI image; b) Its semantic segmentation; c) Its final segmentation map, ERI_{seg}.

Both optimization procedures require the detection of relevant objects, which is accomplished by the semantic segmentation block, producing a segmentation map, ERI_{seg}, of the input image. The semantic segmentation (including the connected component analysis (CCA)), is obtained according to the procedure described in the previous chapter (Section 5.3.1). Figure 6.6 depicts an example of an equirectangular image ERI, its semantic segmentation, and the resulting ERI segmentation map (ERI_{seg}) after CCA, where disconnected objects are represented with different colors.

The Pannini projection with the globally optimized parameters, (d_b, vc_b), is applied to the input image and to ERI_{seg} producing, respectively, a viewport image denoted as VP_b, and its corresponding segmentation map, denoted as VP_b^{seg}, which is used by the mesh optimization procedure. Finally, VP_b is warped according to the optimized mesh, to obtain the final output viewport, VP_{out}. The main steps involved in the GLA-PP projection are described in the following sections.

A. *Global Optimization*

The global optimization aims to find out the Pannini projection parameters, (d_b, vc_b), that result in the least perceived global geometric distortion, for a viewport rendered according to the user viewing direction, (ϕ_{VD}, θ_{VD}), and with a predefined horizontal field of view, F_h, and spatial resolution (W_{vp}, H_{vp}). This is obtained by applying the procedure described in Section 6.2, and where the projection parameters, (d_b, vc_b), are obtained by minimizing a simple cost function defined by (6.1).

In (6.1), parameter α - stretching to bending ratio - seeks the best balance between bending and stretching distortions, and was set to 0.24 in Section 6.2.2. In GLA-PP, to better preserve the straightness of the lines, more importance was given to the line bending than to the stretching of the objects, by reducing α to 0.17. This value was obtained by varying α in the range [0.05, 0.24] with a step size of 0.01, and retaining the value that leads to viewports (rendered from several omnidirectional images) with straighter background lines, at the expense of a slightly decrease of the objects conformality (and since the local optimization will only improve the latter).

B. *Meshes Creation*

Two meshes, M_b and M_f, are generated on the viewport plane, as depicted in Figure 6.7, using the Pannini backward and forward projections. This allows to obtain, for a given position in the viewport, the corresponding positions in both M_b and M_f:

- ***M_b* mesh creation** - A uniform grid mesh, $M_b = \{\boldsymbol{b}_i\}$, is defined over VP_b, consisting of a vertex set $\{\boldsymbol{b}_i\}$, where $\boldsymbol{b}_i$ refers to the i-th vertex Cartesian coordinates, (x^b, y^b), in length units, with origin at the center of the viewport plane. For a given integer position of VP_b, (n, m), with a coordinate system centered on the top-left corner of the viewport plane, the corresponding Cartesian coordinates, (x^b_{nm}, y^b_{nm}), can be computed by (see Section 2.4.8.A)

$$x^b_{nm} = 2\ (d_b + 1)\frac{\sin\left(\frac{F_h}{2}\right)}{d_b + \cos\left(\frac{F_h}{2}\right)}\left(\frac{m + 0.5}{W_m} - \frac{1}{2}\right), 0 \leq m < W_m \tag{6.3}$$

$$y^b_{nm} = 2\ \tan\left(\frac{F_v}{2}\right)\left(\frac{1}{2} - \frac{n + 0.5}{H_m}\right), 0 \leq n < H_m \tag{6.4}$$

where F_v is the vertical FoV, obtained by (*cf.* (5.7) and (5.8))

$$F_v = 2\ \tan^{-1}\left(\frac{(d_b + 1)\ \sin\left(\frac{F_h}{2}\right)}{AR\ \left(d_b + \cos\left(\frac{F_h}{2}\right)\right)}\right), \tag{6.5}$$

and W_m and H_m are the horizontal and vertical mesh resolution, respectively, which were set to $W_m = W_{vp}/c$ and $H_m = W_{vp}/c$, being c a constant; AR is the viewport aspect ratio.

- ***M_f* mesh creation** - The M_b mesh coordinates, (x^b_{nm}, y^b_{nm}), are projected back to the sphere, using the Pannini backward projection with parameters (d_b, vc_b), resulting in the corresponding spherical coordinates (ϕ_{nm}, θ_{nm}). These are then projected to the plane using the Pannini forward projection with parameters (d_f, vc_f), resulting in the corresponding M_f mesh coordinates, (x^f_{nm}, y^f_{nm}), of a vertex set $\{\boldsymbol{f}_i\}$; thus, M_f represents the initial viewport reprojected according to (d_f, vc_f), that should preserve the objects conformality (e.g., stereographic Pannini $(d_f = 1, vc_f = 0)$). The selection of these parameters is detailed in Section 6.3.2.

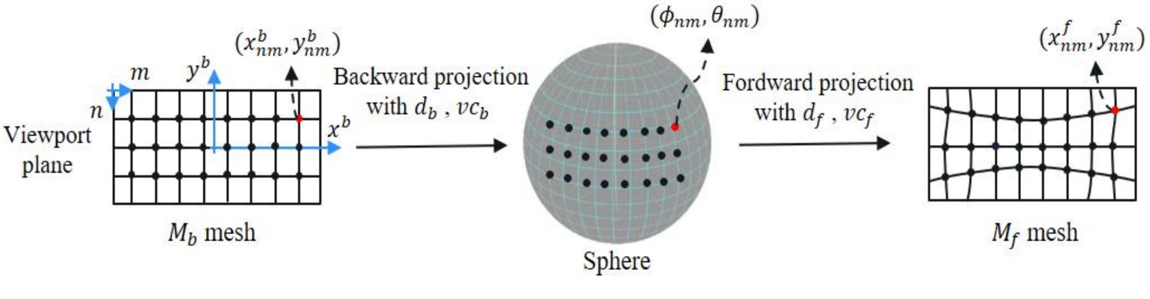

(x^b_{nm}, y^b_{nm})
m
y^b
n
x^b
Viewport plane
Backward projection with d_b , vc_b
M_b mesh
(ϕ_{nm}, θ_{nm})
Sphere
Fordward projection with d_f , vc_f
(x^f_{nm}, y^f_{nm})
M_f mesh

Figure 6.7. *M_b* and *M_f* meshes generation procedure.

Note that to get a uniform M_f mesh with the same resolution as M_b, the procedure was implemented in the other way around: for each integer position (n, m) associated with a vertex $\boldsymbol{f}_i$, the corresponding Cartesian coordinates $\left(x^f_{nm}, y^f_{nm}\right)$, were obtained by first back projecting to the sphere with (d_f, vc_f), and then forward projecting to the viewport plan with (d_b, vc_b). The pseudocode explaining this procedure is provided below.

Algorithm: Pannini meshes creation

```
1: Input: $d_f, vc_f, d_b, vc_b, W_m, H_m, F_h$
2: Output: $M_b, M_f$
3:  for $n = 1$ to $H_m$
4:    for $m$ = 1 to $W_m$
5:      compute $x_{nm}^b, y_{nm}^b$ using (6.3) to (6.5)
6:      compute $(\phi_{nm}, \theta_{nm})$ using (2.32) to (2.36) with $d_f, vc_f$
7:      compute $x_{nm}^f, y_{nm}^f$ using (2.29) to (2.31) with $d_b, vc_b$
8:    end
9:  end
```

C. Mesh Optimization

Based on [55], a mesh optimization algorithm is applied which iterates locally between M_b and M_f, adding smooth changes, to obtain an optimal mesh $M_o = \{\boldsymbol{o}_i\}$, having the following properties: *i)* object shapes are preserved; *ii)* straightness of background lines are preserved; *iii)* abrupt transitions at the object borders (due to the use of two different meshes) are avoided.

A mesh denoted as $M_v = \{\boldsymbol{v}_i\}$ is defined, consisting of a vertex set $\{\boldsymbol{v}_i\}$, where initially $\{\boldsymbol{v}_i\} = \{\boldsymbol{b}_i\}$. The optimized mesh results from minimizing the following cost function:

$$\{\boldsymbol{o}_i\} = \min_{\{\boldsymbol{v}_i\}} E_t(\{\boldsymbol{v}_i\}) \tag{6.6}$$

where E_t is a weighted sum of energy terms, expressed by

$$E_t = \lambda_c E_c + \lambda_b E_b + \lambda_s E_s + \lambda_a E_a \tag{6.7}$$

and $E_c, E_b, E_s,$ and E_a are, respectively, object conformality, line distortion, smoothness, and asymmetric energy terms; $\lambda_c, \lambda_b, \lambda_s$, and λ_a are the weights for the corresponding energy terms. Each one of these energy terms is explained below:

- **Object conformality term** - For each object identified in VP_b^{seg}, a conformality term is computed by

$$O_c = \sum_{i \in \boldsymbol{B}_k} m_i \|\boldsymbol{v}_i - \boldsymbol{f}_i\|_2^2 \tag{6.8}$$

where k is the object index; $\boldsymbol{B}_k$ is the set of vertices on the k-th object; m_i is the correction strength; $\boldsymbol{f}_i$ is the vertex in the M_f mesh; $\boldsymbol{v}_i$ is the vertex in the M_v mesh; $\|.\|_2^2$ denotes the squared Euclidean distance. The O_c encourages the object regions to follow the M_f mesh, where the object shapes are preserved.

Objects located at the viewport borders have higher distortions than the objects close to the viewport center, and thus require more correction; to account it, the following sigmoid function is defined

$$m_i = \frac{1}{1 + \exp\left(-\frac{r_i - r_1}{r_2}\right)} \tag{6.9}$$

where r_i is the radial distance of $\boldsymbol{b}_i$ from the viewport center, r_1 and r_2 are parameters controlling the attenuations of the correction strength and chosen such that $m_i = 0.01$ at the viewport center, and $m_i = 1$ at the viewport border.

The total object conformality is then computed by the sum of all objects conformality and is expressed by

$$E_c = \sum_k O_c(k) \tag{6.10}$$

- **Line distortion term -** To preserve the straightness of the lines on the boundaries between objects and background, where different projections are applied, the following line distortion energy term is computed

$$E_b = \sum_n \sum_{m \in N(n)} \|\boldsymbol{v}_n - \boldsymbol{v}_m \times e_{nm}\|_2^2 \tag{6.11}$$

where e_{nm} is the unit vector along the direction $\boldsymbol{b}_n - \boldsymbol{b}_m$ in the M_b mesh, that preserves the lines straightness, and × denotes the cross product.

- **Smoothness term -** To have a smooth transition at the object borders, the following smoothness term is computed

$$E_s = \sum_n \sum_{m \in N(n)} \|\boldsymbol{v}_n - \boldsymbol{v}_m\|_2^2 \,. \tag{6.12}$$

This term encourages smoothness between 4-way adjacent vertices and thus avoids abrupt changes in the final viewport.

- **Asymmetric cost term -** Due to the mesh optimization that tries to satisfy the previous terms, some visual artifacts (e.g., geometric distortions and black regions) may appear at regions close to the viewport borders. Thus, to reduce these artifacts, the following asymmetric cost term is computed

$$E_a = E_l + E_r + E_t + E_b \tag{6.13}$$

where $E_l, E_r, E_t,$ and E_b are, respectively, left, right, top, and bottom mesh boundary constraints, given by

$$E_l = \mathbb{I}(v_{i,x} > 0) \times \|v_{i,x}\|_2^2, \quad \forall i \in \partial_{left} \tag{6.14}$$

$$E_r = \mathbb{I}(v_{i,x} > W_m) \times \|v_{i,x} - W_m\|_2^2, \quad \forall i \in \partial_{right} \tag{6.15}$$

$$E_t = \mathbb{I}(v_{i,y} > 0) \times \|v_{i,y}\|_2^2, \quad \forall i \in \partial_{top} \tag{6.16}$$

$$E_b = \mathbb{I}(v_{i,y} > H_m) \times \|v_{i,y} - H_m\|_2^2, \quad \forall i \in \partial_{bottom} \tag{6.17}$$

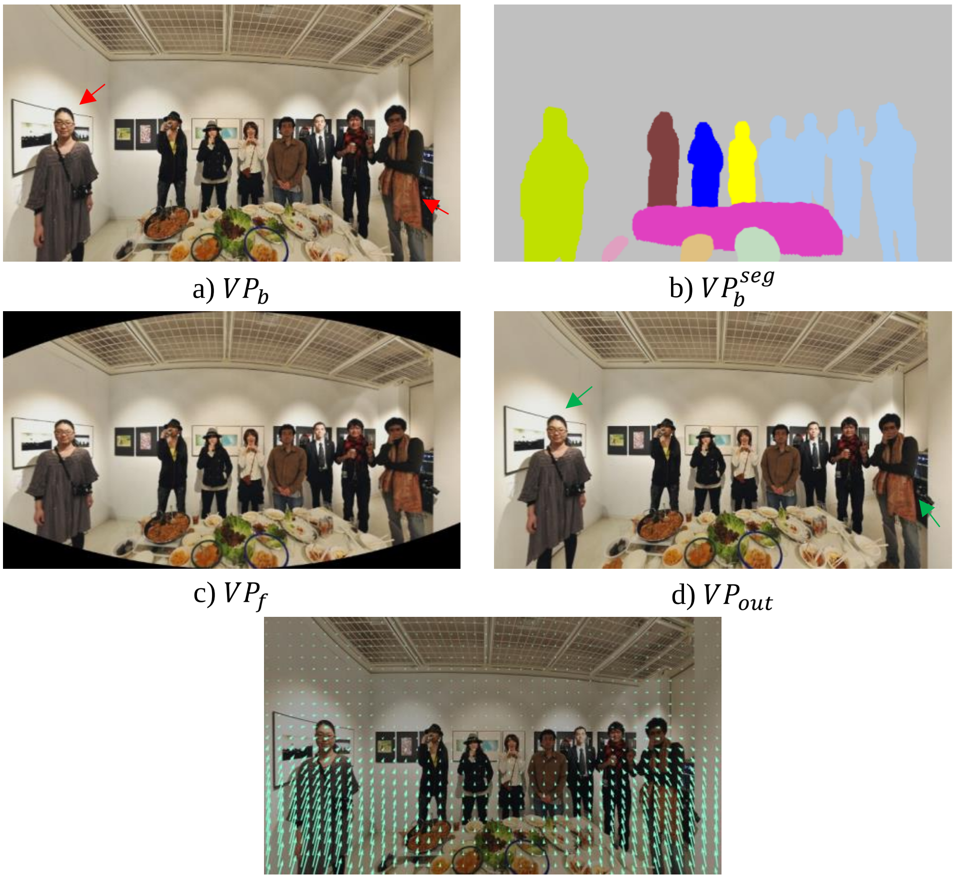
where $\mathbb{I}(.)$ is the indicator function that returns 1 for the true condition and 0 otherwise; ∂_* are the original mesh boundary.

A gradient-based algorithm [141], with 100 iterations and a learning rate of 0.02, was used for the mesh optimization. This method was implemented in PyTorch [142], which is computationally efficient and suitable for mesh optimization.

D. *Viewport Warping*

The final viewport, VP_{out}, is obtained by warping the globally optimized viewport, VP_b, according to the optimized mesh, M_o. The warping package available in [143] was used for this purpose. This process requires interpolation for non-integer pixel positions; in this work, bilinear interpolation was used.

Figure 6.8 depicts a viewport, VP_b, obtained from the globally optimized Pannini projection with $d_b = 0.5, vc_b = 0.6$; its segmentation map, VP_b^{seg}; a viewport, VP_f, obtained by warping VP_b according to a M_f mesh generated with $d_f = 0.5, vc_f = 0$; and the final optimized viewport, VP_{out}. As can be seen, the objects stretching presented in VP_b (e.g., the girl on the left side is vertically stretched), is reduced significantly in the VP_{out}, while straight lines in the background remain straight. Figure 6.8e) shows the optical flow mask [144] overlaid on VP_b. This mask was computed between the two meshes, M_b and M_o, and it shows the VP_b regions that are modified by the mesh optimization procedure. As shown in Figure 6.8e), the bottom-left and the bottom-right have the strongest flow (or projection modifications, to reduce the stretching) compared to other regions, which was expected since there are two objects (lady on the left and boy on the right, in Figure 6.8) located in these regions, too much stretched in the vertical direction.



a) VP_b

b) VP_b^{seg}

c) VP_f

d) VP_{out}

e) VP_b with optical flow mask

Figure 6.8. a) Globally optimized Pannini viewport with HFoV of 150º; b) its segmentation map; c) viewport obtained by warping VP_b, according to the M_f mesh; d) final output viewport; e) VP_b with optical flow mask.

6.3.2 GLA-PP Projection Parameters Selection

To obtain the M_f mesh, the corresponding (d_f, vc_f) Pannini projection parameters need to be found. While a stereographic Pannini $(d_f = 1, vc_f = 0)$ favours the conformality of the objects, it may result in visible distortions on the objects boundaries if the global projection parameters have very distinct values.

The appropriate values of (d_f, vc_f) were obtained by visual inspection of the optimized viewports, VP_{out}, for several omnidirectional images, varying d_f in the range of [0.1, 1] with a step $\Delta d = 0.1$, and $vc_f = 0$. The vc_f value was set to 0 since, for $d_f \neq 0$ and $vc_f > 0$, object stretching becomes visible. It was found that if $|d_b - d_f| > 0.2$, the regions close to the object boundaries may be distorted on the final viewport, particularly the straight lines. Accordingly, d_f was set to $d_b + 0.2$, being d_b automatically obtained by the global optimization procedure.

The cost function defined by (6.7) has four parameters λ_c , λ_b, λ_s , and λ_a. To tune these parameters, the following steps were applied:

1) Initialize the parameters according to $\lambda_c = 4$, $\lambda_b = 2$, $\lambda_s = 0.5$ and $\lambda_a = 4$. Although other values are possible, this initialization provided a good starting point.

2) Tune these parameters sequentially, one at a time, varying the parameter in the range [0.1, 6] with a step size of 0.1, and retain the value that leads to the best viewport quality, by visual inspection.

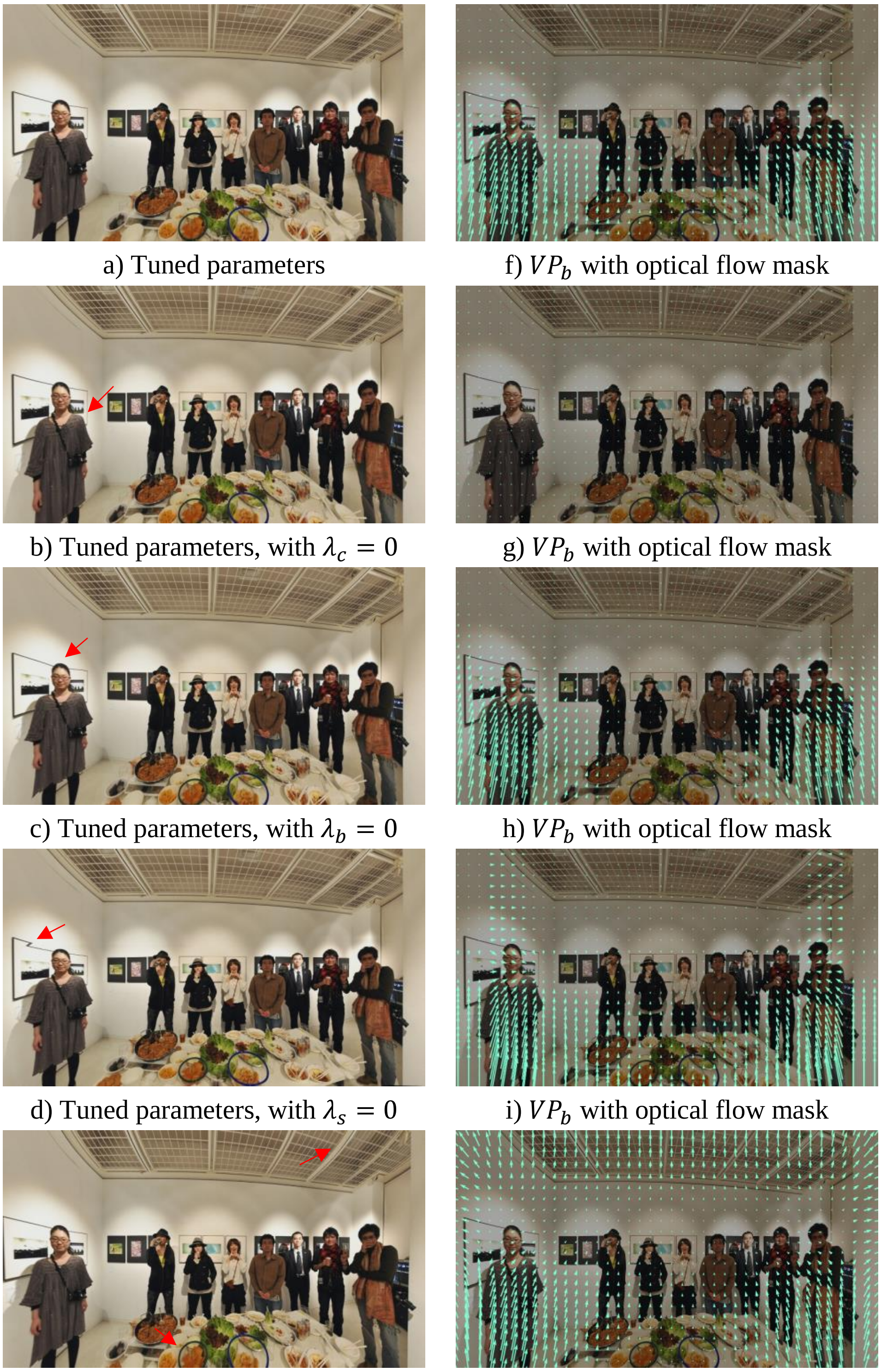
The steps were applied to several omnidirectional images, and the best values found for λ_c , λ_b, λ_s , and λ_a were, respectively, 0.3, 1.5, 0.5, 3. To evaluate the impact of these parameters on the final output, the GLA-PP viewport was obtained with the tuned parameter values, being the result shown in Figure 6.9a); after, the projection was repeated with each parameter set to 0, one at a time, and the results are shown in Figure 6.9b)-e). When $\lambda_c = 0$, the objects are stretched in the output viewport (*cf.* Figure 6.9b). When $\lambda_b = 0$, the straight lines between the objects and background are deformed, e.g., the radial lines behind the girl on the left side of Figure 6.9c). When $\lambda_s = 0$, dramatic changes happen for some image regions, e.g., the painting behind the girl on the left side of Figure 6.9d). When $\lambda_a = 0$, the regions close to the viewport borders are distorted (*cf.* Figure 6.9e).

6.4 Projection Performance Evaluation

This section describes the crowdsourcing subjective assessment of the proposed GA-PP and GLA-PP projections. For comparison purposes, the following benchmark projections were also included on the test: PP with fixed parameters, $(d = 0.5, vc = 0)$; GPP with fixed parameter, $d = 0.5$; and OP and MOP projections proposed in [10]. While the PP and GPP projections are widely established content-unaware projections, the OP and MOP are automatic content-aware projections, both based on the Pannini projection. The OP and MOP viewports were obtained from the authors of [10], since the source code was not available.

6.4.1 Test Conditions

The GA-PP and the GLA-PP viewports were obtained as described in Sections 6.2.2 and 6.3.2, respectively. The viewports had a horizontal FoV, F_h, of 150° and a spatial resolution of 960×540 pixels (aspect ratio, $AR = 16/9$), as in [10]. In GLA-PP, and to speed up the mesh optimization procedure, the mesh dimension was set to 192×108, which corresponds to $\left\lfloor \frac{W_{vp}}{5} \right\rfloor, \left\lfloor \frac{H_{vp}}{5} \right\rfloor$, where $\lfloor . \rfloor$ is the floor operator. After optimization, the optimized mesh was resized with bilinear interpolation to the viewport resolution.



a) Tuned parameters

f) VP_b with optical flow mask

b) Tuned parameters, with $\lambda_c = 0$

g) VP_b with optical flow mask

c) Tuned parameters, with $\lambda_b = 0$

h) VP_b with optical flow mask

d) Tuned parameters, with $\lambda_s = 0$

i) VP_b with optical flow mask

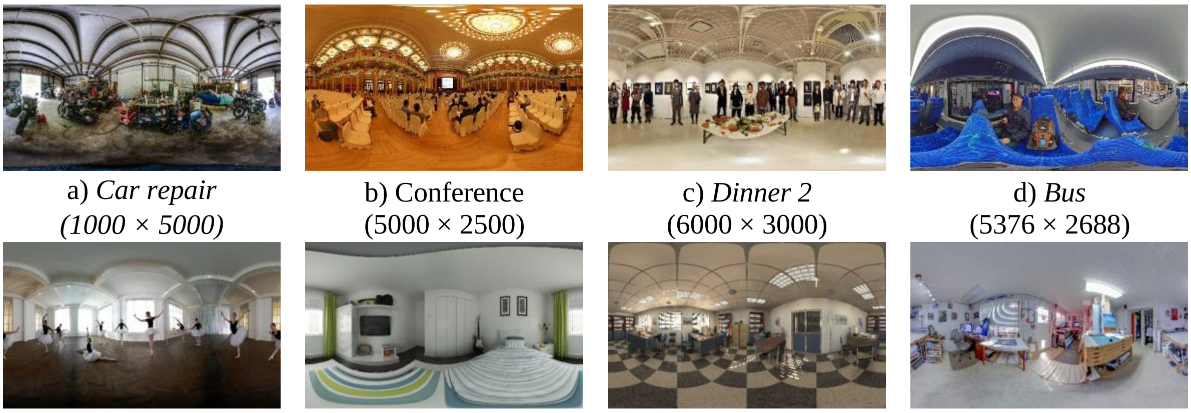
e) Tuned parameters, with $\lambda_a = 0$

j) VP_b with optical flow mask

Figure 6.9. Viewports on the left side were obtained with GLA-PP using a HFoV of 150º, using several parameters configuration; viewports on the right side correspond to VP_b with optical flow mask, showing (in green) the viewport regions modified by the mesh optimization procedure.

Table 6.1. The omnidirectional images, the used projections, and the total number of comparisons for each image group.

Group	360º images	Dataset	Projections	Number of Comparisons
G1	*Dance*, *Bedroom*, *Office 1*, *Office 4*	[10]	GPP, PP, OP, MOP, GA-PP, GLA-PP	60
G2	*Car repair*, *Conference*, *Dinner 2*, *Bus*	[52]	GPP, PP, GA-PP, GLA-PP	24



a) *Car repair* *(1000 × 5000)*

b) Conference (5000 × 2500)

c) *Dinner 2* (6000 × 3000)

d) *Bus* (5376 × 2688)

e) *Dance* (3840 × 1920)

f) *Bedroom* (2000 × 1000)

g) *Office 1* (8000 × 4000)

h) *Office 4* (8000 × 4000)

Figure 6.10. Omnidirectional images used in the subjective test, and their spatial resolutions.

Eight omnidirectional images in equirectangular format (*ERI*) were used in the subjective assessment. To have different image content characteristics, e.g., objects near and far away from the camera and the presence or absence of people, two groups of images, *G1* and *G2* (presented in Table 6.1), were taken from two different datasets: *G1* from [10] and *G2* from [52]. The images taken from [52] and also *Dance*, taken from [10], were already considered in the previous subjective tests. These three images were included to be able to compare the proposed projections to OP and MOP, since the source code was not available. Per image, one viewing direction was considered. Thus, six viewports were obtained, corresponding to the proposed and the benchmark projections. The omnidirectional images resolutions are depicted in Figure 6.10.

6.4.2 Subjective Evaluation Method

The pairwise comparison (PC) method was chosen for the subjective evaluation of projections. For each omnidirectional image, a complete set of comparisons was performed (i.e., all possible pairs of comparisons), which resulted in 15 comparisons per omnidirectional image in *G1*. However, to limit the test duration to less than half an hour, thus avoiding the observer fatigue, viewports from OP and MOP were excluded from the test in *G2*, thus reducing to six the number of comparisons per omnidirectional image. Table 6.1 presents the used omnidirectional images, projections, and the total number of comparisons, for *G1* $(15\,(\text{comparisons/image}) \times 4(\text{images}) = 60)$ and *G2* (6 (comparisons/image) × 4(images) $= 24)$.

As on the previous chapter, the subjective test was conducted online through a web-based crowdsourcing interface, described in the previous chapter (Section 5.2.2) that allows to display two viewports, 'A' and 'B', side by side, with random order and position. The observers were asked to select the viewport, ‘A’ or ‘B’, that has the best quality in his/her opinion, or option ‘A=B’ in case of no difference, to avoid random preference selections. The total number of observers that participated in the online subjective test was 30. The used viewports and the resulting PC subjective scores are available in [145].

6.4.3 Subjective Test Results and Analysis

The outliers were detected according to the procedure described in Chapter 3 (Section 3.3.2.C). Four outliers were detected, and their subjective scores were not further considered. Next, for each compared viewport pair (i, j), the winning frequency, w_{ij} and preference probability P_{ij} were computed. The preference probabilities were then translated to absolute quality scores using the Bradley-Terry (BT) model. The computation of winning frequencies, preference probabilities, and BT scores were detailed in Chapter 3 (Section 3.3.2.C).

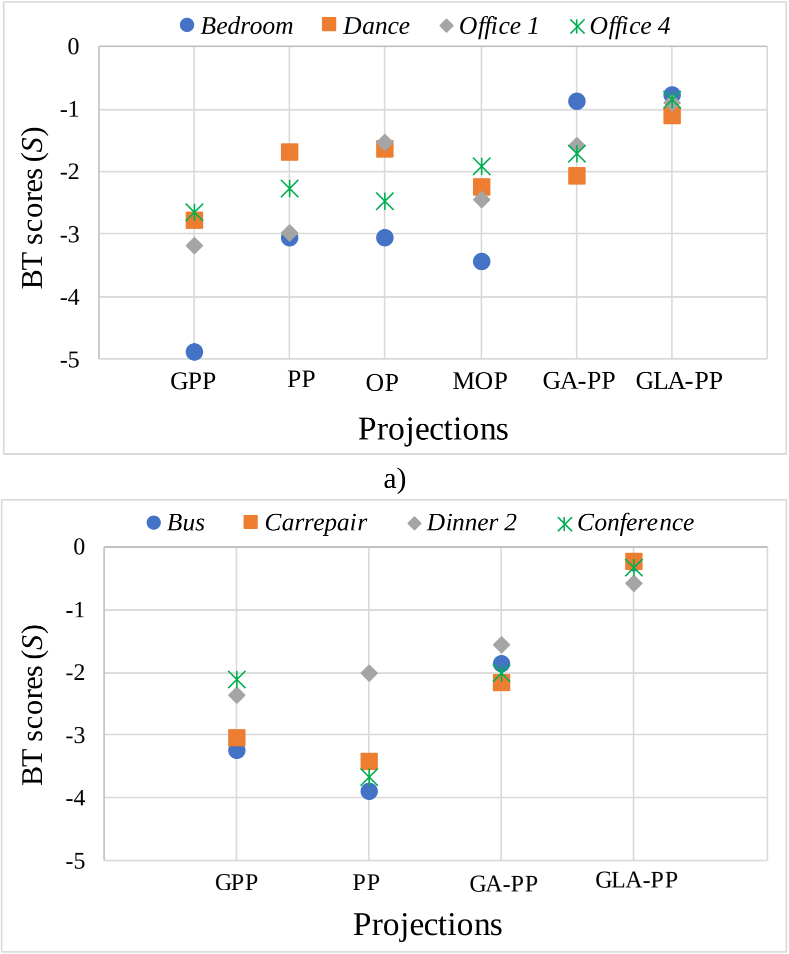
Table 6.2 presents the preferences probabilities for the considered projections, and per image group, averaged over the different images in each group. In Table 6.2, the values in green and blue color correspond, respectively, to the preference probabilities for the images in *G1* and in *G2*. Accordingly, the following conclusions can be taken:

- **GA-PP *vs* benchmark** - For the images in *G1*, the proposed GA-PP projection is preferred over all benchmark projections by 68% (minimum) to 75% (maximum) of the subjects. The GA-PP is prefereed over MOP by 74%. For the images in *G2*, the GA-PP projection is preferred over the GPP projections by 78%, and over PP by 77%, of the subjects.
- **GLA-PP *vs* benchmark** - For the images in *G1*, the proposed GLA-PP projection is preferred over all benchmark projections by 73% (minimum) to 84% (maximum) of the subjects. The GLA-PP outperforms the best content-aware benchmark projection available in the literature, MOP, by a large margin, since 79% of the subjects prefered it. For the images in *G2*, the GLA-PP is preferred over the GPP and PP, by 85% and 89% of the subjects, respectively.
- **GLA-PP *vs* GA-PP** - The GLA-PP is prefereed over GA-PP by 72% for the images in *G1*, and by a large margin, 91%, for images in *G2*, showing the advantage of having the projection locally adapted to the content.

Table 6.2. Preference probabilities for compared projections in *G1*/*G2*. NA corresponds to Not Available.

	GPP [14]	PP [17]	OP [28]	MOP [28]	GA-PP	GLA-PP
GPP	-	0.28/0.67	0.29/NA	0.19/NA	0.25/0.22	0.16/0.15
PP	0.72/0.33	-	0.39/NA	0.56/NA	0.30/0.23	0.16/0.11
OP	0.71/NA	0.61/NA	-	0.61/NA	0.32/NA	0.27/NA
MOP	0.81/NA	0.44/NA	0.39/NA	-	0.26/NA	0.21/NA
GA-PP	0.75/0.78	0.70/0.77	0.68/NA	0.74/NA	-	0.28/0.09
GLA-PP	**0.84/0.85**	**0.84/0.89**	**0.73/NA**	**0.79/NA**	**0.72/0.91**	-

Figure 6.11 depicts the resulting BT scores obtained for each projection and image group. As shown, the proposed GLA-PP obtained the highest quality scores for all images in both *G1* and *G2*. Interestingly, in *G1* the benchmark projections results are not consistent and highly depend on the image content, e.g. for *Office 4*, MOP and GA-PP result in a similar quality (the second



a)

b)

Figure 6.11. BT scores *vs*. projections for each considered 360º image in a) *G1* and b) *G2*.

highest for that image), while for the *Bedroom* the quality is even lower than the resulting for the content-unaware Pannini projection (PP). This behaviour also happened for GA-PP but not for GLA-PP. In *G1*, and excluding the GLA-PP projection, GA-PP obtained the highest quality scores for *Bedroom* and *Office 4*, while for *Dance* the highest quality scores were obtained for OP and PP, and for *Office 1* it were OP and GA-PP that got the best results. In *G2*, the GA-PP obtained better scores than GPP and PP; the quality scores are higher for GPP than for PP for images *Car repair*, *Conference*, and *Bus*. These images have more relevant horizontal lines than vertical lines, and the horizontal line bending is more visible on PP than on GPP viewports. If the relevant lines were vertical, the results would be more favorable to PP, since the vertical lines remain straight in the viewports rendered with PP. The GPP had the lowest quality scores in *G1*, but in *G2* it was PP the lowest performing projection, except for *Dinner* 2, where PP is slightly better than GPP.

In summary, the proposed projections, CA-PP and GLA-PP, lead to higher perceived quality compared to previous state-of-the-art, notable content-aware projections based on the Pannini projection.

6.4.4 Projection Qualitative Evaluations

Figure 6.12 depicts some viewport examples obtained for the proposed GLA-PP and for the OP and MOP proposed in [10], allowing the following comparisons:

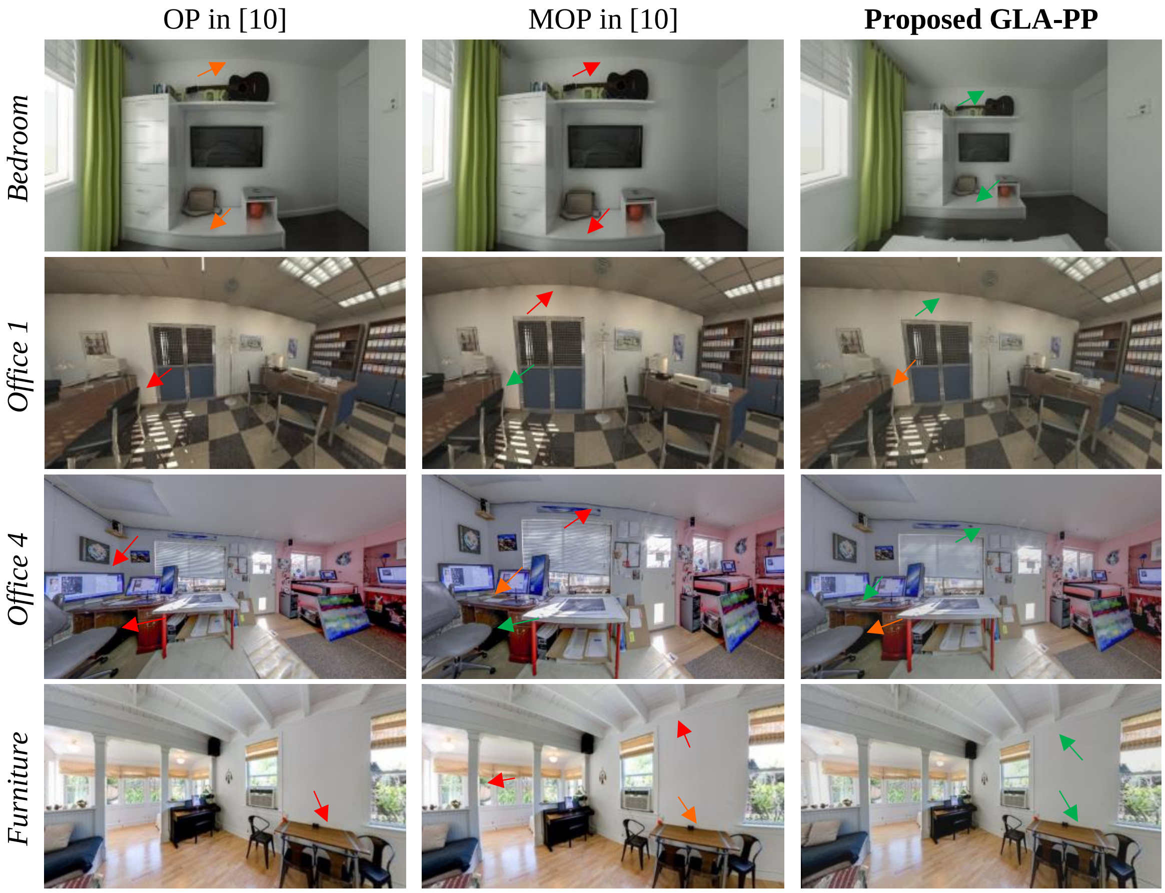


Figure 6.12. Viewport examples obtained with OP and MOP in [10], and with GLA-PP, using a HFoV of 150º. The red, orange, and green arrows indicate, respectively, the objects/regions with high, medium, and low geometric distortions.

- **GLA-PP *vs* OP -** The GLA-PP viewports have less geometric distortion than the viewports resulting from OP. For *Bedroom*, the horizontal lines on the ceiling and on the floor are straighter for GLA-PP. In *Office 1*, the chair on the left side is more conformal and the horizontal line on the ceiling is straighter for GLA-PP. In *Office 4*, the monitor and the chair on the left side are stretched too much for OP. In *Furniture,* GLA-PP kept the horizontal lines as straight as OP, but the objects shape (e.g., the table and the chairs on the right side) is more conformal for GLA-PP.

- **GLA-PP *vs* MOP -** The viewports obtained for MOP have more geometric distortions than the viewports resulting from GLA-PP. MOP has a poor balance between bending and stretching; the horizontal lines are too much bent, and some vertical lines are also bent for some images, e.g., in *Furniture*. Also in *Furniture*, the table on the right side is globally deformed.

- **OP vs MOP** - In OP, straight lines are better preserved than in MOP, but the objects are more stretched, e.g., the chair on the left side of Office 1, the monitor and the chair on the left side of Office 4, the table and the chairs on the right side of Furniture.

Figure 6.13 depicts examples of viewports obtained for GA-PP and GLA-PP. For this evaluation, the locally optimized projection (LOP) proposed in [55] was also considered, but for correcting general objects (and not just for correcting human faces, as in [55]). Figure 6.14

depicts the same viewports of Figure 6.13, but with cropped objects to better compare their conformality for different projections. The following conclusions can be taken:

- **GLA-PP *vs* LOP** - The GLA-PP viewports have a much better perceived quality than LOP viewports. The LOP stretches the objects too much since it uses a mixture of two projections, rectilinear and stereographic, and the former is known for a strong perspective effect and objects stretching, notably when a large FoV is used.
- **GLA-PP vs GA-PP** - The horizontal lines are less bent for the GLA-PP, particularly for the *Conference*, *Carrepair*, and *Bus* viewports. The stretching distortion, that is visible for some objects/regions, are significantly reduced in the GLA-PP viewports.
- **GA-PP vs LOP** - In general, the GA-PP viewports have a more pleasant visual quality than the LOP viewports. In LOP, the straight lines are better preserved than in GA-PP, but the perceptual impact of object distortions, in the former, is very strong and annoying (*cf.* Figure 6.14). The GA-PP provides a good balance between stretching and bending.

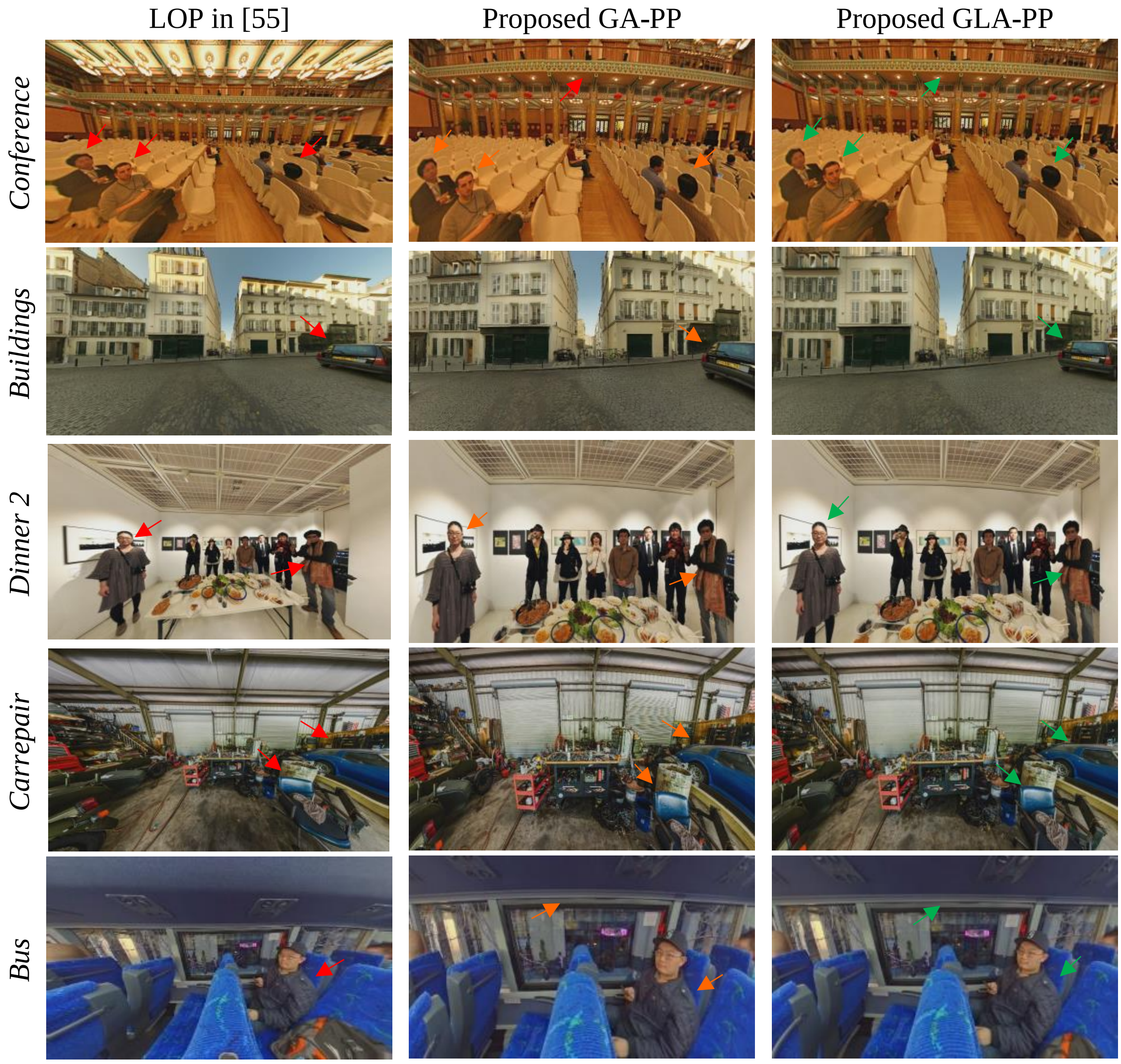


Figure 6.13. Viewport examples obtained with LOP from [55], and with GA-PP and GLA-PP projections, using a HFoV of 150º.

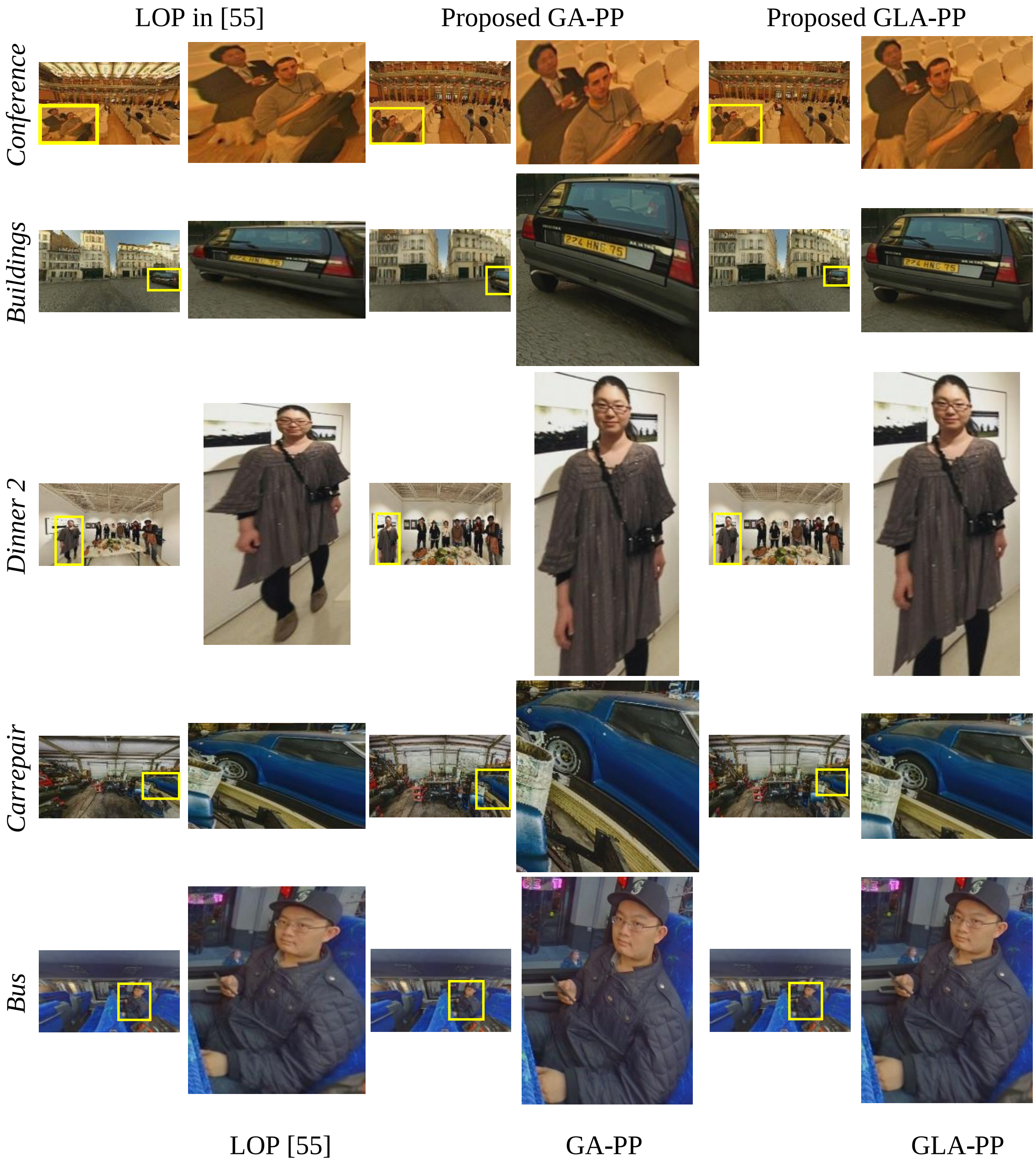


Figure 6.14. Comparing object conformality for viewport examples obtained with LOP from [55], and with GA-PP and GLA-PP projections, using a HFoV of 150º.

It is important to mention that the proposed GA-PP and GLA-PP could be useful for other interesting applications; for example, in photography, they could be applied for reducing the geometric distortions in photos taken from wide-angle cameras. In this case, and assuming that the distorted image can be approximated by a rectilinear image, an improved image could be obtained by transforming the distorted one to the spherical domain with the backward rectilinear projection, followed by a forward projection with the proposed GA-PP and GLA-PP.

6.5 Final Remarks

In this chapter, two fully automatic Pannini-based projections - the globally adapted Pannini (GA-PP) and the globally and locally adapted Pannini (GLA-PP) - were proposed for the viewport rendering of omnidirectional images, aiming to reduce the geometric distortions when high FoVs (~150º) are used. In GA-PP the projection parameters are globally optimized based on the viewport content, resulting in a single pair of parameters, $(d, vc),$ that is use in the whole viewport. In GLA-PP, the Pannini projections parameters are firstly globally optimized according to the image content (as in GA-PP), followed by a local conformality improvement of relevant viewport objects, where the human perception is more sensitive. A crowdsourcing subjective test was conducted to evaluate the proposed projections, showing that they were the most preferred solutions among the considered state-of-the-art, sphere to plan projections, producing viewports with a more pleasant visual quality. This may allow to enhance the user's QoE for several applications and services that uses omnidirectional images (e.g., VR and AR applications).

The GA-PP has been included in the journal paper (presented in the first row of Table 6.3), and the GLAP-PP has been submitted to a conference, and is presently under revision process (presented in the second row of Table 6.3).

Table 6.3. Publications related to this chapter.

Paper	Type
F. Jabar, J. Ascenso, and M.P. Queluz, "Object-Based Geometric Distortion Metric for Viewport Rendering of 360° Images", *IEEE Access*, vol.10, no.1, 13827-13843, Jan. 2022.	Journal
F. Jabar, J. Ascenso, and M.P. Queluz, "Globally and Locally Optimized Pannini Projection for Viewport Rendering of 360° Images", Submitted to J. Vis. Commun. Image Represent., Oct. 2022.	Journal

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The research topic of this Thesis is on the subjective and objective quality assessment of omnidirectional images viewport rendering, and its optimization. It considers the perceptual impact of the geometric distortions (such as stretching of image regions/objects and bending of straight lines) introduced during the rendering process, due to the sphere to plane projection. Subjective evaluation is essential to assess the perceptual impact of these geometric distortions reliably, and objective quality metrics are needed to automate the quality assessment process. In fact, objective quality metrics allow the optimization of the sphere to plane projection used for viewport rendering in a perceptual way, and thus to obtain viewports with enhanced quality.

Several subjective quality assessment studies were performed to evaluate the perceptual impact of geometric distortions. Based on the analysis of the subjective test results, novel content-dependent geometric distortion metrics were proposed. Moreover, the proposed metrics were used to optimize two well-known sphere to plane projections, namely the general perspective projection (GPP) and the Pannini projection (PP), for the viewport rendering of omnidirectional images.

The work developed throughout this Thesis, and the main conclusions that were reached, can be summarized as follows:

- **Chapter 2** - In this chapter, several content-unaware and content-aware sphere to plane projections were reviewed, and a new classification method for these projections was proposed; moreover, the relevant projections were qualitatively evaluated. The qualitative evaluation showed that the different projections present a trade-off between the different types of geometric distortions, and no projection can avoid the visibility of some of those distortions. Moreover, in general, content-aware projections have less visible geometric distortions than content-unaware projections.

- **Chapter 3** - In this chapter, subjective quality assessment test campaigns were conducted to evaluate: *i)* the impact of geometric distortions on the perceived viewport quality, using the GPP for viewport rendering; *ii)* the impact of the FoV on the user immersive experience, using the rectilinear projection for viewport rendering. In the first study, the subjective test results showed that the projection type, the considered FoV, and the image content characteristics, are the three main factors that influence the geometric distortion strength, and its visibility. An important part of this study is the resulting GPP viewport dataset and associated quality scores, which are needed for the development, and validation, of objective quality metrics. The second study showed that the best trade-off between user immersive experience and geometric distortion perception is achieved for a FoV close to 110º, regardless of the image content.

- **Chapter 4** - In this chapter, several content-dependent stretching and bending metrics were proposed to characterize and measure the stretching of image regions and the bending of straight lines. The stretching metrics are based on Tissot indicatrices, while the bending metrics use some characteristics of the projected lines. Both metrics were evaluated by correlating their values with the quality scores, obtained from the subjective test campaign; while the bending metrics showed to be well correlated with those scores, the stretching measures achieved a low performance, requiring further study. Moreover, a quality prediction model was built to automatically assess the geometric distortions and predict the viewport quality when the GPP is used for viewport rendering. The model is based on Support Vector Regression (SVR), and was built using the best performing stretching and bending metrics, proposed previously. The experimental results showed that this model can predict the viewport quality with a Pearson correlation coefficient close to 0.8. The last part of this chapter was dedicated to the automatic optimization of the GPP projection parameter, d, in a perceptual sense, resulting in content-aware general perspective projections, CA-GPP and CA-GPP*. In CA-GPP, d is obtained based on the proposed SVR-based quality prediction model. In CA-GPP*, d is obtained by minimizing a simple cost function that models the resulting geometric distortions through a linear combination of bending and stretching metrics. Both CA-GPP and CA-GPP* showed significant performance improvement when compared to the popular rectilinear and stereographic projections.

- **Chapter 5** - This chapter was dedicated to further improve the stretching distortion metric. First, a subjective crowdsourcing campaign was conducted to evaluate the perceptual impact of the stretching distortion, free of the influence of the bending, and to collect the required ground truth quality scores for the metric development and assessment. The key idea was to identify the relevant objects in the viewport, using semantic segmentation, and to compute the stretching distortion for each object. Two distinct approaches were exploited and evaluated: the first one, directly computes and compares object shape measures on the sphere and on the viewport; the second one is based on Tissot indicatrices, which are computed for individual objects in the viewport. The experimental results showed that while the Tissot based method performed slightly better than the direct shape measurement, both approaches outperformed the considered benchmark solutions; furthermore, they were able to classify the viewport quality, with respect to the ground truth quality scores, with a correct decision percentage close to 90%.

- **Chapter 6** - This chapter addressed the viewport rendering of omnidirectional images with large FoVs (~150º), using the Pannini projection. Two content-aware, Pannini-based projections were proposed: the globally adapted Pannini (GA-PP), and the globally and locally adapted Pannini (GLA-PP). In GA-PP, the projection parameters, d and vc, are globally optimized based on the viewport content, resulting in a single pair of parameters, (d, vc), that is use in the whole viewport. Accordingly, stretching and/or bending distortions may be still visible in some image regions and structures. In GLA-PP, the Pannini projections parameters are firstly globally optimized according to the image content (as in GA-PP), followed by a local conformality improvement of relevant viewport objects, where the human perception is more sensitive. A crowdsourcing subjective test was conducted to evaluate the proposed projections, showing that they were the most preferred solutions among the considered state-of-the-art, sphere to plane projections, producing viewports with a more pleasant visual quality.

7.2 Future Work

The research conducted in this Thesis has resulted in several subjective quality assessment studies and objective distortion metrics, which have shown good performance on the quality assessment, and optimization, of omnidirectional images viewport rendering. Still, additional research work could be conducted, notably:

- Subjectively assess the FoV impact for the projections proposed in this Thesis, namely CA-GPP, GAP and GLAP, by conducting tests similar to those described in Chapter 3.
- Integrate a line detection technique on the mesh optimization procedure, for GLAP, so that the mesh optimization procedure would not deform those lines.
- Subjectively assess the perceptual impact of the stretching distortion on different object classes, such as people, furniture, cars, and so on. This may allow to obtain weights for object classes, that could be integrated with the geometric distortions metrics to further improve their performance.

Besides the above future work, which is a direct extension of this Thesis developments, other and more challenging future research directions can be envisioned, notably:

- **Projection optimization under user navigation -** Except for the FoV impact, where user navigation was considered on the subjective tests, all studies and developments of this Thesis were conducted with static viewports. However, and depending on the omnidirectional image content, the best projection could vary with the viewing direction; yet, the subjective impact of the projection variation during navigation, and how to cope with it, needs to be further investigated, requiring additional and specific subjective tests where user interaction is allowed or simulated. Some existing work (e.g., [146]) has revealed that when the user navigation velocity is slow (changing the viewing direction slowly, or keeping a static direction) the users pay more attention to regions/objects and thus the distortions have a higher perceptual impact, compared to the case when the navigation velocity is fast (users actively search for the next salient regions/objects). Therefore, it could be sufficient to smoothly vary the projection during slow navigation. A related topic, also worthy to be investigated, is the projection optimization for omnidirectional videos rendering. In this case, other factors may have an impact on the optimization, such as objects motion, camera motion and the existence of scene cuts.
- **Projection optimization for HMD devices -** Only 2D displays, namely standard personal computer monitors, were considered in this Thesis. However, and as mentioned in Chapter 1, there are other ways to display omnidirectional visual content, including head-mounted displays (HMDs), smartphones and tablets, and it is expected that the geometric distortions impact will be not the same for all these devices, since they have different characteristics. In the case of HMDs, there are two displays close to the user's eyes, and thus the users may pay more attention to the regions near the point of fixation (foveated vision), compared to the regions away from that point (peripheral vision). Accordingly, the proposed objective

quality metrics and content-aware projections should be validated (and eventually modified, if needed) for other devices, requiring also additional subjective evaluation assessments.

- **Deep learning-based geometric distortion correction** - To further reduce viewport geometric distortions, another interesting direction for future work is to employ deep learning-based techniques. In [148], global geometric artifacts in 2D images, due to camera lens characteristics (e.g., barrel and pincushion distortions) are reduced as well as corrections to the viewing perspective (e.g., rotation, shearing, perspective change). In [148], convolutional neural networks are used with classical model fitting and a new resampling method to reduce 2D images geometric distortions and thus further improve their quality. In the context of this Thesis, this type of approach introduces other challenges such as the development of a large viewport dataset, which may require user interaction to have viewport-free geometric distortions images (reference) for which a known model is not available as in previous work. Moreover, if local adaptation is desired, multiple models may be necessary for a single image, which may require more complex fitting procedures.

References

[1] Moving Picture Experts Group (MPEG), “MPEG-I: Coded Representation of Immersive Media,” 2022. [Online]. Available: https://mpeg.chiariglione.org/standards/mpeg-i. [Accessed: 26-Apr-2022].

[2] K. Brunnström, S. A. Beker, K. de Moor, A. Dooms, and S. Egger, “Qualinet White Paper on Definitions of Quality of Experience,” in *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003),* Lausanne, Switzerland, Mar. 2013.

[3] M. Domański, O. Stankiewicz, K. Wegner, and T. Grajek, “Immersive Visual Media - MPEG-I: 360 Video, Virtual Navigation and Beyond,” in *International Conference on Systems, Signals and Image Processing,* Poznan, Poland, May 2017.

[4] ITU-T. Recommendation G.1035, “Influencing Factors on Quality of Experience for Virtual Reality Services,” ITU, Geneva, Switzerland, May 2020.

[5] M. Wien, J. M. Boyce, T. Stockhammer, and W.-H. Peng, “Standardization Status of Immersive Video Coding,” *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 9, no. 1, pp. 5–17, Mar. 2019.

[6] Wikipedia, “Six Degrees of Freedom.” [Online]. Available: https://en.wikipedia.org/wiki/Six_degrees_of_freedom. [Accessed: 05-Apr-2021].

[7] Wikipedia, “General Perspective Projection.” [Online]. Available: https://en.wikipedia.org/wiki/General_Perspective_projection. [Accessed: 12-Mar-2021].

[8] P. Hanhart, Y. He, and Y. Ye, “Viewport-based Subjective Evaluation of 360-degree Video Coding,” ISO/IEC JVET E0071, Geneva, Switzerland, Jan. 2017.

[9] A. Abbas, “GoPro VR Player: A Tool for VR Content Playback,” ISO/IEC JVET D0177, Chengdu, China, Oct. 2016.

[10] Y. W. Kim, D. Jo, C. Lee, H. Choi, Y. H. Kwon, and K. Yoon, “Automatic Content-aware Projection for 360° Videos,” in *2017 IEEE International Conference on Computer Vision (ICCV),* Venice, Italy, Dec. 2017.

[11] J. J. Lin, H. B.L. Duh, D. E, Parker, H. Abi-Rached, and T. A.Furness, “Effects of Field of View on Presence, Enjoyment, Memory, and Simulator Sickness in a Virtual Environment,” in *Proceedings IEEE Virtual Reality 2002,* Orlando, USA, Mar. 2002.

[12] N. F. Polysm, S. Kim, and D. A. Bowman, “Effects of Information Layout, Screen Size, and Field of View on User Performance in Information-Rich Virtual Environments,” *Comput. Animat. Virtual Worlds,* vol. 18, no. 1, pp. 19–38, Nov. 2006.

[13] C. Fan, W. Lo, Y. Pai, and C. Hsu, “A Survey on 360◦ Video Streaming: Acquisition, Transmission, and Display,” *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, Aug. 2019.

[14] Facebook, “Facebook Surround 360.” [Online]. Available: https://engineering.fb.com/2016/04/12/video-engineering/introducing-facebook-surround-360-an-open-high-quality-3d-360-video-capture-system/. [Accessed: 15-Mar-2021].

[15] GoPro, “GoPro Odyssey.” [Online]. Available: https://gopro.com/en/us/news/here-is-odyssey. [Accessed: 15-Mar-2021].

[16] R. G. de A. Azevedo, N. Birkbeck, F. De Simone, I. Janatra, B. Adsumilli, and P. Frossard, “Visual Distortions in 360° Videos,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8,

pp. 2524–2537, Aug. 2020.

[17] S. Knorr, S. Croci, and A. Smolic, “A Modular Scheme for Artifact Detection in Stereoscopic Omni-Directional Images,” in *Irish Machine Vision and Image Processing Conference*, Kildare, Ireland, Aug. 2017.

[18] Wikipedia, “Equirectangular Projection.” [Online]. Available: https://en.wikipedia.org/wiki/Equirectangular_projection. [Accessed: 04-Mar-2021].

[19] Wikipedia, “Cube Mapping.” [Online]. Available: https://en.wikipedia.org/wiki/Cube_mapping. [Accessed: 17-Mar-2021].

[20] Z. Chen, Y. Li, and Y. Zhang, “Recent Advances in Omnidirectional Video Coding for Virtual Reality: Projection and Evaluation,” *Signal Processing*, vol. 146, pp. 66–78, May 2018.

[21] M. Yu, H. Lakshman, and B. Girod, “A Framework to Evaluate Omnidirectional Video Coding Schemes,” in *IEEE International Symposium on Mixed and Augmented Reality*, Fukuoka, Japan, Jan. 2015.

[22] I. Hussain and O. Kwon, “Evaluation of 360◦ Image Projection Formats; Comparing Format Conversion Distortion Using Objective Quality Metrics,” *Imaging*, vol. 7, no. 137, Aug. 2021.

[23] T. Wiegand, G. J.Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC Video Coding Standard,” *Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Aug. 2003.

[24] G. J.Sullivan, J. R.Ohm, W. J.Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Sep. 2012.

[25] B. Bross, J. Chen, S. Liu, and Y. Wang, “Versatile Video Coding (Draft 9),” ISO/IEC JVET-R2001-vA, Geneva, Switzerland, Apr. 2020.

[26] R. G. Youvalari, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, “Efficient Coding of 360 Degree Pseudo-Cylindrical Panoramic Video for Virtual Reality Applications,” in *IEEE International Symposium on Multimedia (ISM)*, San Jose,CA, USA, Dec. 2016.

[27] Y. Liu, M. Xu, C. Li, S. Li, and Z. Wang, “A Novel Rate Control Scheme for Panoramic Video Coding,” in *IEEE International Conference on Multimedia and Expo (ICME)*, Hong Kong, China, Jul. 2017.

[28] Y. Li, J. Xu, and Z. Chen, “Spherical Domain Rate-distortion Optimization for 360-degree Video Coding,” in *IEEE International Conference on Multimedia and Expo (ICME)*, Hong Kong, China, Jul. 2017.

[29] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, “Viewport-Adaptive Encoding and Streaming of 360-Degree Video for Virtual Reality Applications,” in *IEEE International Symposium on Multimedia (ISM)*, San Jose, USA, Dec. 2016.

[30] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, “Viewport-adaptive Navigable 360-degree Video Delivery,” in *IEEE International Conference on Communications (ICC)*, Paris, France, May 2017.

[31] C. Zhou, Z. Li, J. Osgood, and Y. Liu, “On the Effectiveness of Offset Projections for 360-Degree Video Streaming,” *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 14, no. 3, pp. 1–24, Jun. 2018.

[32] M. Hosseini and V. Swaminathan, “Adaptive 360 VR Video Streaming: Divide and Conquer,” in *IEEE International Symposium on Multimedia (ISM)*, San Jos, USA, Dec. 2016.

[33] C. Ozcinar, A. De Abreu, and A. Smolic, “Viewport-aware Adaptive 360° Video Streaming Using Tiles for Virtual Reality,” in *EEE International Conference on Image Processing (ICIP)*, Beijing, China, Sep. 2017.

[34] L. Xie, Z. Xu, Y. Ban, and Z. Guo, “360ProbDASH: Improving QoE of 360 Video Streaming Using Tile-based HTTP Adaptive Streaming,” in *25th ACM Iernational Conference on Multimedia*, Mountain View, CA, USA, Oct. 2017.

[35] R. Shafi, W. Shuai, and M. U. Younus, “360-Degree Video Streaming: A Survey of the State of the Art,” *Symmetry (Basel).*, vol. 12, no. 9, p. 1491, Sep. 2020.

[36] R. Carroll, M. Agrawala, and A. Agarwala, “Optimizing Content-Preserving Projections for Wide-Angle Images,” *ACM Trans. Graph.*, vol. 28, no. 3, pp. 43–1, Aug. 2009.

[37] C. Chang, M. Hu, W. Cheng, and Y. Chuang, “Rectangling Stereographic Projection for Wide-Angle Image Visualization,” in *2013 IEEE International Conference on Computer Vision*, Sydney, Australia, Dec. 2013.

[38] Wikipedia, “List of Map Projections.” [Online]. Available: https://en.wikipedia.org/wiki/List_of_map_projections. [Accessed: 11-Mar-2022].

[39] Panotools, “Projections.” [Online]. Available: https://wiki.panotools.org/Main_Page. [Accessed: 19-Mar-2021].

[40] Wikipedia, “Map projection,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Map_projection. [Accessed: 01-Mar-2021].

[41] QGIS, “Coordinate Reference Systems.” [Online]. Available: https://docs.qgis.org/2.8/en/docs/gentle_gis_introduction/coordinate_reference_systems.html#figure-projection-families. [Accessed: 17-Mar-2021].

[42] Wikipedia, “Lambert Conformal Conic Projection.” [Online]. Available: https://en.wikipedia.org/wiki/Lambert_conformal_conic_projection#/media/File:Lambert_conformal_conic_projection_SW.jpg. [Accessed: 17-Mar-2021].

[43] L. Zelnik-Manor, G. Peters, and P. Perona, “Squaring the Circle in Panoramas,” in *IEEE International Conference on Computer Vision (ICCV)*, Beijing, China, Oct. 2005.

[44] T. K.Sharpless, B. Postle, and D. M.German, “Pannini : A New Projection for Rendering Wide Angle Perspective Images,” in *Proc. of the 6th Int. Conf. on Computational Aesthetics in Graphics, Visualization and Imaging*, London, United Kingdom, Jul. 2010.

[45] M. Kennedy and S. Kopp, *Understanding Map Projections*. US: Esri Press, Jul. 2001.

[46] J. P.Snyder, “Map Projections--A Working Manual,” *U.S. Geological Survey Professional Paper 1395 (Supersedes USGS Bulletin 1532),* 1987. [Online]. Available: https://pubs.usgs.gov/pp/1395/report.pdf.

[47] Wikipedia, “Azimuthal Equidistant Projection.” [Online]. Available: https://en.wikipedia.org/wiki/Azimuthal_equidistant_projection#/media/File:Azimuthal_equidistant_projection_SW.jpg. [Accessed: 05-Mar-2021].

[48] Wikipedia, “Lambert Cylindrical Equal-Area Projection.” [Online]. Available: https://en.wikipedia.org/wiki/Lambert_cylindrical_equal-area_projection#/media/File:Lambert_cylindrical_equal-area_projection_SW.jpg. [Accessed: 05-Mar-2021].

[49] Wikipedia, “Robinson Projection.” [Online]. Available: https://en.wikipedia.org/wiki/Robinson_projection. [Accessed: 05-Apr-2021].

[50] Cambridge in Colour, “Panoramic Image Projections.” [Online]. Available: https://www.cambridgeincolour.com/tutorials/image-projections.htm. [Accessed: 26-Mar-2021].

[51] J. P.Snyder and P. M.Voxland, *An Album of Map Projections*. U.S Department of the Interiorr, Jan., 1989.

[52] J. Gutiérrez, E. J. David, A. Coutrot, M. Silva, and P. Le Callet, "Introducing UN Salient360!Benchmark: A Platform for Evaluating Visual Attention Models for 360° contents," in *International Conference on Quality of Multimedia Experience (QoMEX),* Sardinia, Italy, May 2018.

[53] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, "JVET Common Test Conditions and Evaluation Procedures for 360° Video," ISO/IEC JVET H1030, Macao, China, Oct. 2017.

[54] J. Kopf, D. Lischinski, O. Deussen, D. Cohen-Or, and M. Cohen, "Locally Adapted Projections to Reduce Panorama Distortions," *Comput. Graph. Forum,* vol. 28, no. 4, pp. 1083–1089, Jun. 2009.

[55] Y. Shih, W. Lai, and C. Liang, "Distortion-Free Wide-Angle Portraits on Camera Phones," *ACM Trans. Graph.,* vol. 38, no. 4, pp. 1–12, Jul. 2019.

[56] C. Chang, W. Lai, and Y. Chuang, "Generating a Perspective Image from a Panoramic Image by the Swung-to Cylinder Projection," in *25th IEEE International Conference on Image Processing (ICIP),* Athens, Greece, Sep. 2018.

[57] K. Hormann and G. Greiner, "MIPS: An Efficient Global Parametrization Method," in *4th International Conference on Curves and Surfaces,* Saint-Malo, France, Jul. 1999.

[58] N. Wadhwa, R. Garg, D. Jacobs, B. Feldman, N. Kanazawa, and R. Carroll, "Synthetic Depth-of-Field with a Single-Camera Mobile Phone," *ACM Trans. Graph.,* vol. 37, no. 4, Jul. 2018.

[59] D. Zorin and A. H. Barr, "Correction of Geometric Perceptual Distortions in Pictures," in *Proc. of SIGGRAPH , ACM SIGGRAPH,* Los Angeles, USA, Sep. 1995.

[60] C. E. Duchon, "Lanczos Filtering in One and Two Dimensions," *J. Appl. Meteorol.,* vol. 18, no. 8, pp. 1016–1022, Aug. 1979.

[61] Gimp, "Gimp." [Online]. Available: https://www.gimp.org/. [Accessed: 13-Apr-2021].

[62] E. N. N.Mortensen and W. A.Barrett, "Intelligent Scissors for Image Composition," in *SIGGRAPH 95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques,* Jul. 1995.

[63] M. H. Pirenne, *Optics Painting and Photography*. USA: Cambridge University Press, Sep., 1970.

[64] R. G. von Gioi, J.´er´emie Jakubowicz, and J.-M. Morel, "LSD: A Line Segment Detector," *Image Process. Line,* vol. 1, no. 2, pp. 35–55, Mar. 2012.

[65] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the Objectness of Image Windows," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 34, no. 11, pp. 2189–2202, Jan. 2012.

[66] M. Uhrina, J. Bienik, and T. Mizdos, "QoE on H.264 and H.265: Crowdsourcing versus Laboratory Testing," in *2020 30th International Conference Radioelektronika,* Bratislava, Slovakia, Apr. 2020.

[67] M. Xu, C. Li, S. Zhang, and P. Le Callet, "State-of-the-Art in 360° Video/Image Processing: Perception, Assessment and Compression," *IEEE J. Sel. Top. Signal Process.,* vol. 14, no. 1, pp. 5–26, Jan. 2020.

[68] E. Upenik, M. Řeřábek, and T. Ebrahimi, "Testbed for Subjective Evaluation of Omnidirectional Visual Content," in *Picture Coding Symposium (PCS),* Nuremberg, Germany, Dec. 2016.

[69] ITU-T Recommendation P.910, "Subjective Video Quality Assessment Methods for Multimedia Applications," ITU, Geneva, Switzerland, Nov. 2021.

[70] ITU-R Recommendation BT.500-13, "Methodology for the Subjective Assessment of the Quality of Television Pictures," ITU, Geneva, Switzerland, Jan. 2012.

[71] ITU-T Recommendation P.800, "Methods for Subjective Determination of Transmission Quality," ITU, Geneva, Switzerland, Aug. 1996.

[72] ITU-TRecommendation J.149, “Method for Specifying Accuracy and Cross-Calibration of Video Quality Metrics,” ITU, Geneva, Switzerland, Mar. 2004.

[73] ITU-R Recommendation BT.2022, “General Viewing Conditions for Subjective Assessment of Quality of SDTV and HDTV Television Pictures on Flat Panel Displays,” ITU, Geneva, Switzerland, Aug. 2012.

[74] ITU-R Recommendation BT.500-11, “Methodology for the Subjective Assessment of the Quality of Television Pictures,” ITU, Geneva, Switzerland, Oct. 2002.

[75] P. C.Madhusudan and R. Soundararajan, “Subjective and Objective Quality Assessment of Stitched Images for Virtual Reality,” *IEEE Trans. Imgae Process.*, vol. 28, no. 28, pp. 5620–5635, Nov. 2019.

[76] K. Okarma, W. Chlewicki, M. Kopytek, B. Marciniak, and V. Lukin, “Entropy-Based Combined Metric for Automatic Objective Quality Assessment of Stitched Panoramic Images,” *Entropy*, vol. 23, no. 11, p. 1525, Nov. 2021.

[77] J. Li, K. Yu, Y. Zhao, Y. Zhang, and L. Xu, “Cross-Reference Stitching Quality Assessment for 360◦ Omnidirectional Images,” in *In Proceedings of the 27th ACM International Conference on Multimedia*, Nice, France, Oct., 2019.

[78] J. Boyce, E. Alshina, and Z. Deng, “Subjective Testing Method for Comparison of 360° Video Projection Formats Using HEVC,” ISO/IEC JTC 1/SC 29/WG 11 JVET-F1004v3, Hobart, Australia, Apr. 2017.

[79] P. Hanhart, Y. He, Y. Ye, J. Boyce, Z. Deng, and L. Xu, “360-Degree Video Quality Evaluation,” in *2018 Picture Coding Symposium (PCS)*, San Francisco, USA, Jun. 2018.

[80] V. Zakharchenko, K. P.Choi, and J. H.Park, “Quality Metric for Spherical Panoramic Video,” in *in SPIE 9970, Optics and Photonics for Information Processing X, 99700C*, Sep. 2016.

[81] R. Schatz, A. Sackl, C. Timmerer, and B. Gardlo, “Towards Subjective Quality of Experience Assessment for Omnidirectional Video Streaming,” in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, Erfurt, Germany, Jun. 2018.

[82] R. Schatz, A. Zabrovskiy, and C. Timmerer, “Tile-based Streaming of 8K Omnidirectional Video: Subjective and Objective QoE Evaluation,” in *Proc. of 11th International Conference on Quality of Multimedia Experience (QoMEX)*, Berlin, Germany, Jun. 2019.

[83] M. Xu, C. Li, Z. Chen, Z. Wang, and Z. Guan, “Assessing Visual Quality of Omnidirectional Videos,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 12, pp. 3516–3530, Dec. 2019.

[84] B. Zhang, L. Zhao, S. Yang, Y. Zhang, L. Wang, and Z. Fei, “Subjective and Objective Quality Assessment of Panoramic Videos in Virtual Reality Environments,” in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Hong Kong, China, Jul. 2018.

[85] F. Lopes, J. Ascenso, A. Rodrigues, and M. P.Queluz, “Subjective and Objective Quality Assessment of Omnidirectional Video,” in *SPIE Optical Engineering + Applications (OP18O), Applications of Digital Image Processing XLI Conference*, San Diego, CA, USA. Aug. 2018.

[86] W. Zou, L. Yang, F. Yang, Z. Ma, and Q. Zhao, “The Impact of Screen Resolution of HMD on Perceptual Quality of Immersive Videos,” in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, London, UK, Jul. 2020.

[87] V. Zakharchenko, K. P. Choi, and J. H.Park, “Quality Metric for Spherical Panoramic Video,” in *SPIE 9970, Optics and Photonics for Information Processing X, 99700C-1*, Sep. 2016.

[88] Y. Sun, A.Lu, and Y.Lu, “WS-PSNR for 360 video objective quality evaluation,” ISO/IEC JTC1/WG11, Geneva, Switzerland, May 2016.

[89] S. Chen, Y. Zhang, Y. Li, Z. Chen, and Z. Wang, “Spherical Structural Similarity Index for Objective Omnidirectional Video Quality Assessment,” in *IEEE International Conference on Multimedia and Expo (ICME)*, San Diego, CA, USA, Jul. 2018.

[90] Y. Zhou, M. Yu, H. Ma, H. Shao, and G. Jiang, “Weighted-to-Spherically-Uniform SSIM Objective Quality Evaluation for Panoramic Video,” in *2018 14th IEEE International Conference on Signal Processing (ICSP)*, China, Aug. 2018.

[91] E. Upenik, M. Rerabek, and T. Ebrahimi, “On the Performance of Objective Metrics for Omnidirectional Visual Content,” in *9th International Conference on Quality of Multimedia Experience (QoMEX 2017),* Erfurt, Germany, May 2017.

[92] H. T. T. Tran, N. Pham Ngoc, C. Manh Bui, M. Hong Pham, and T. Cong Thang, “An evaluation of quality metrics for 360 videos,” in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN),* Milan, Italy, Jul. 2017.

[93] Y. Liu, H. Yu, B. Huang, G. Yue, and B. Song, “Blind Omnidirectional Image Quality Assessment Based on Structure and Natural Features,” *IEEE Trans. Instrum. Meas.,* vol. 70, no. 1, Aug. 2021.

[94] R. G. de A. Azevedo, N. Birkbeck, I. Janatra, B. Adsumilli, and P. Frossard, “Multi-Feature 360 Video Quality Estimation,” *IEEE Open J. Circuits Syst.*, vol. 2, no. 1, pp. 338–349, May 2021.

[95] L. Cao, G. Jiang, Z. Jiang, M. Yu, Y. Qi, and Y.-S. Ho, “Quality Measurement for High Dynamic Range Omnidirectional Image Systems,” *IEEE Trans. Instrum. Meas.*, vol. 70, no. 1, Jul. 2021.

[96] A. Tissot, *Memoire sur la Representation des Surfaces et les Projections des Cartes Geographiques*. Paris, France: Gauthier Villars, Aug. 1881.

[97] Wikipedia, “Tissot’s indicatrix.” [Online]. Available:
https://en.wikipedia.org/wiki/Tissot%27s_indicatrix. [Accessed: 04-May-2021].

[98] Wikipedia, “Cauchy–Riemann Equations.” [Online]. Available:
https://en.wikipedia.org/wiki/Cauchy–Riemann_equations. [Accessed: 24-Sep-2021].

[99] F. Jabar, “IST GPP Dataset,” *IEEE Dataport*, 2020. [Online]. Available: https://ieee-dataport.org/documents/ist-gppdataset.

[100] Y. Rai, P. Le Callet, and P. Guillotel, “Which Saliency Weighting for Omnidirectional Image Quality Assessment?,” in *Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, Erfurt, Germany, May 2017.

[101] S. Ling, J. Gutiérrez, K. Gu, and P. Le Callet, “Prediction of the Influence of Navigation Scan-Path on Perceived Quality of Free-Viewpoint Videos,” *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 9, no. 1, pp. 204–216, Mar. 2019.

[102] StarVR, “The next-generation StarVR.” [Online]. Available: https://www.starvr.com/.
[Accessed: 10-Nov-2021].

[103] Valve Index, “Valve Index Headset.” [Online]. Available:
https://www.valvesoftware.com/en/index. [Accessed: 10-Nov-2021].

[104] Wikipedia, “Virtual Reality,” 2022. [Online]. Available:
https://en.wikipedia.org/wiki/Virtual_reality. [Accessed: 04-May-2022].

[105] J. Napieralla, “Comparing Graphical Projection Methods at High Degrees of Field of View,” Blekinge Institute of Technology, Jun. 2018.

[106] Z. M. Osman, J. Dupire, A. Topol, and P. Cubaud, “A Non Intrusive Method for Measuring Visual Attention Designed for the Study and Characterization of Users’ Behavior in Serious Games,” *Int. J. Adv. Internet Technol.,* vol. 7, no. 3, pp. 262–271, Dec. 2014.

[107] J. Boyce, R. Doré, A. Dziembowski, J. Fleureau, J. Jung, B. Kroon, and B. Salahieh, “MPEG Immersive Video Coding Standard,” *Proc. IEEE,* vol. 109, no. 9, pp. 1521–1536, Sep. 2021.

[108] Y. Rai, J. Gutiérrez, and P. Le Callet, “A Dataset of Head and Eye Movements for 360 Degree Images,” in *Proc. of the 8th ACM on Multimedia Systems Conf.*, Taipei, Taiwan, Jun. 2017.

[109] Y. Yu, S. Lee, J. Na, J. Kang, and G. Kim, “A Deep Ranking Model for Spatio-Temporal Highlight Detection from a 360◦ Video,” in *32nd AAAI Conference on Artificial Intelligence.*, New Orleans, USA, Feb. 2018.

[110] F. Jabar, “IST Viewport Navigation Video Dataset,” *IEEE Dataport,* 2021. [Online]. Available: https://ieee-dataport.org/documents/ist-viewport-navigation-video-dataset.

[111] Z. Zhang, J. Zhou, N. Liu, X. Gu, and Y. Zhang, “An Improved Pairwise Comparison Scaling Method for Subjective Image Quality Assessment,” in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB),* Cagliari, Italy, Jun. 2017.

[112] J.-S. Lee, F. De Simone, and T. Ebrahimi, “Subjective Quality Evaluation via Paired Comparison: Application to Scalable Video Coding,” *IEEE Trans. Multimed.*, vol. 13, no. 5, pp. 882–893, Oct. 2011.

[113] A. B. Watsona and L. Kreslakeb, “Measurement of Visual Impairment Scales for Digital Video,” in *Proc. of SPIE Electronic Imaging, Human Vision and Electronic Imaging,* San Jose, CA, USA, Jan. 2001.

[114] R. A. Bradley, *Paired Comparisons: Some Basic Procedures and Examples*. Elsevier Science Publishers, Feb. 1984.

[115] J. C. Handley, “Comparative Analysis of Bradley-Terry and Thurstone-Mosteller Paired Comparison Models for Image Quality Assessment,” in *PICS : Image: Processing, Quality, Capture, Systems,* Montreal, Canada, Apr. 2001.

[116] R. Azevedo, N. Birkbeck, F. De Simone, I. Janatra, B. Adsumilli, and P. Frossard, “Visual Distortions in 360-degree Videos,” *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, Aug. 2019.

[117] B. Jenny, B. Šavrič, and T. Patterson, “A Compromise Aspect-adaptive Cylindrical Projection for World Maps,” *Int. J. Geogr. Inf. Sci.*, vol. 29, no. 6, pp. 935–952, Mar. 2015.

[118] C. Akinlar, C. Topal., A. Cuney, T. Cihan, C. Akinlar, and T. Cihan, “EDLines: A Real-Time Line Segment Detector with a False Detection Control,” *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1633–1642, Oct. 2011.

[119] F. Jabar, M. P.Queluz, and J. Ascenso, “Objective Assessment of Line Distortions in Viewport Rendering of 360° Images,” in *1st IEEE International Conference on Artificial Intelligence and Virtual Reality,* Taichung, Taiwan, Dec. 2018.

[120] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, “A Deep Multi-Level Network for Saliency Prediction,” in *23rd International Conference on Pattern Recognition,* Cancun, Mexico, Dec. 2016.

[121] L. Yang, Z. Tan, Z. Huang, and G. Cheung, “A Content-Aware Metric for Stitched Panoramic Image Quality Assessment,” in *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, Venice, Italy. Oct. 2017.

[122] S. Ling, G. Cheung, and P. Le Callet, “No-Reference Quality Assessment for Stitched Panoramic Images Using Convolutional Sparse Coding and Compound Feature Selection,” in *IEEE International Conference on Multimedia and Expo (ICME),* San Diego, CA, USA, Jul. 2018.

[123] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[124] R. Monroy, S. Lutz, T. Chalasani, and A. Smolic, “SalNet360: Saliency Maps for Omni-directional Images with CNN,” *Signal Process. Image Commun.*, vol. 69, pp. 26–34, Nov. 2018.

[125] F. Jabar, “IST Pannini Dataset,” *IEEE Dataport,* 2021. [Online]. Available: https://ieee-dataport.org/documents/ist-panninidataset.

[126] P. Hanhart, M. Rerabek, and T. Ebrahimi, “Towards High Dynamic Range Extensions of HEVC: Subjective Evaluation of Potential Coding Technologies,” in *SPIE Optical Engineering + Applications*, San Diego, CA, United States, Sep. 2015.

[127] Wikipedia, “Binomial Distribution.” [Online]. Available: https://en.wikipedia.org/wiki/Binomial_distribution. [Accessed: 17-Jun-2021].

[128] Y. Xu, K. Wang, K. Yang, D. Sun, and Jia Fu, “Semantic Segmentation of Panoramic Images Using a Synthetic Dataset,” in *Proc of SPIE 11169, Artificial Intelligence and Machine Learning in Defense Applications*, Strasbourg, France, Sep. 2019.

[129] K. Yang, X. Hu, Y. Fang, K. Wang, and R. Stiefelhagen, “Omnisupervised Omnidirectional Semantic Segmentation,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–16, Sep. 2020.

[130] K. Yan, J. Zhang, S. Reiß, X. Hu, and R. Stiefelhagen, “Capturing Omni-Range Context for Omnidirectional Segmentation,” *Arxiv Prepr. arXiv2103.05687*, Mar. 2021.

[131] K. Yang, X. Hu, and R. Stiefelhagen, “Is Context-Aware CNN Ready for the Surroundings? Panoramic Semantic Segmentation in the Wild,” *IEEE Trans. IMAGE Process.*, vol. 30, Jan. 2021.

[132] C. Liu, L. Chen, F. Schroff, H. Adam, W. Hua, A. Yuille, and L. Fei-Fei, “Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019.

[133] O. Russakovsky, J. Deng, H. Su, J. Kraus, S. Satheesh, and S. Ma, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Apr. 2015.

[134] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. Zitnick, “Microsoft COCO: Common Objects in Context,” in *European Conference on Computer Vision*, Zurich, Switzerland, Sep. 2014.

[135] M. Everingham, S. M.A.Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge: A Retrospective,” *Int. J. Comput. Vis.*, vol. 111, pp. 98–136, Jun. 2014.

[136] Mathworks, “Bwconncomp.” [Online]. Available: https://www.mathworks.com/help/images/ref/bwconncomp.html. [Accessed: 23-Jun-2021].

[137] M. A. Wirth, “Shape Analysis & Measurement,” University of Guelph, Canada, Tech., Rep. 6320, Jun. 2002.

[138] L. da F. Costa and R. M. C. Jr, *Shape analysis and Classification: Theory and Practice.*, 1st ed. CRC Press, Dec. 2000.

[139] M. Basaraner and S. Cetinkaya, “New Measures for Analysis and Comparison of Shape Distortion in World Map Projections,” *Cartogr. Geogr. Inf. Sci.*, vol. 46, no. 6, pp. 518–531, Jan. 2019.

[140] P. Hanhart, L. Krasula, P. Le Callet, and T. Ebrahimi, “How to Benchmark Objective Quality Metrics From Paired Comparison Data,” in *8th International Conference on Quality of Multimedia Experience (QoMEX)*, Lisbon, Portugal, Jun. 2016.

[141] D. P.Kingma and J. Lei Ba, “Adam: A Method for Stochastic Optimization,” in *International Conference on Learning Representations*, San Diego, USA, May 2015.

[142] Pytorch, “Torch Optimization.” [Online]. Available: https://pytorch.org/docs/stable/optim.html. [Accessed: 25-Jan-2022].

[143] OpenCV, “Geometric Image Transformations.” [Online]. Available: https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html. [Accessed: 25-Jan-2022].

[144] Wikipedia, “Optical Flow.” [Online]. Available: https://en.wikipedia.org/wiki/Optical_flow#:~:text=Optical flow or optic flow,brightness pattern in an image. [Accessed: 09-May-2022].

[145] F. Jabar, “Subjective Assessment of Pannini Projections,” *IEEE Dataport*, 2022. [Online]. Available: https://ieee-dataport.org/documents/subjective-assessment-rendering-projections. [Accessed: 18-Apr-2022].

[146] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, “Saliency in VR: How Do People Explore Virtual Environments?,” *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 4, pp. 1633–1642, Jan. 2018.

[147] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, “Support Vector Machines,” *IEEE Intell. Syst. Their Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998.

[148] M. Sokolova and G. Lapalmeb, “A Systematic Analysis of Performance Measures for Classification Tasks,” *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009.

[149] I. Alhashim and P. Wonka, “High Quality Monocular Depth Estimation Via Transfer Learning,” *Arxiv Prepr. arXiv1812.11941v2*, Mar. 2019.

[148] X. Li, B. Zhang, P. V. Sander, and J. Liao, “Blind Geometric Distortion Correction on Images Through Deep Learning,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, USA, Jun. 2019.

Annex A

SVM-based Model for Objectively Assessing the GPP

This annex describes an SVM-based model for objectively assessing the general perspective projection (GPP), and which uses the line bending metrics and pooling functions detailed in Section 4.3; it exploits the fact that geometric distortions on structural lines have a high subjective impact. The model output is a label that indicates if the viewport quality obtained with a given projection center, $d_q \neq 0$ (or query d), is better, worse or similar, to the quality obtained with the rectilinear projection. Then, for a typical omnidirectional image rendering system, this model allows to decide if it is beneficial to use a projection other than the conventional rectilinear projection; in such a case, the perceptual quality of the viewport image may increase (in some cases rather significantly) the quality of experience when users interact with omnidirectional images.

A.1 Methodology

The model architecture is illustrated in Figure A.1. First, for an input equirectangular image (*ERI*), viewport viewing direction (ϕ_{VD}, θ_{VD}), spatial resolution (W_{vp}, H_{vp}), and horizontal FoV, F_h, the model renders the viewport with rectilinear projection $(d = 0)$. After, straight lines are detected from the obtained viewport image and then merged and filtered out (as described in Section 4.3.1). Then, several line bending metrics (or features), described in Section 4.3.2, are extracted considering a specific projection defined by parameter d_q, with $d_q \in]0,1]$. The features represent the bending and inclination of the projected lines, which reflect the distortion introduced by this process. From these features, a Support Vector Machine (SVM) classifier outputs the relative quality (or quality class) that could be obtained if d_q is used in the viewport rendering, instead of applying $d = 0$. The possible SVM output corresponds to the groups identified in Figure A.2: *i)* G1 - viewport quality with d_q is better than with $d = 0$; *ii)* G2 - viewport quality with d_q and with $d = 0$ are similar; and *iii)* G3 - viewport quality with d_q is worse than with $d = 0$. Note that, due to the way that features are computed, it is not necessary to compute the viewport projection for d_q. Thus, several d_q values can be quickly evaluated to find out if other projections, different from the rectilinear, may provide better quality.

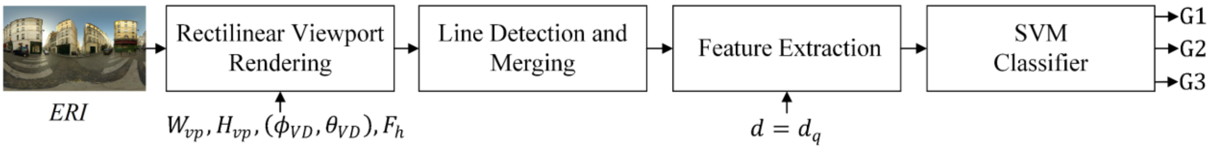

ERI
Rectilinear Viewport Rendering
Line Detection and Merging
Feature Extraction
SVM Classifier
G1
G2
G3
$W_{vp}, H_{vp}, (\phi_{VD}, \theta_{VD}), F_h$
$d = d_q$

Figure A.1. Proposed SVM-based model for objectively assessing the GPP projection.

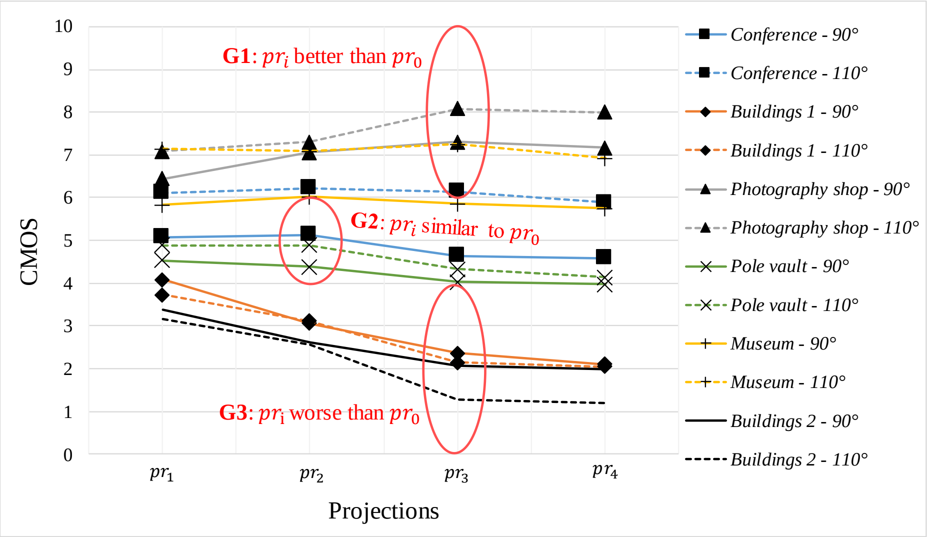


Figure A.2. CMOS values for six omnidirectional images (obtained in the subjective test presented in Chapter 3 (Section 3.2) and FoV of 90° and 110° (averaged over the three viewing directions), versus projection center, d. The GPP with $d = 0, 0.25, 0.5, 0.75, 1$ are referred to as pr_0, pr_1, pr_2, pr_3 and pr_4, respectively.

A detailed description of each step is presented in the following sections.

A.2 SVM-based Classifier

The SVM is a machine learning technique used for classification problems and thus it is rather suitable to assess the relative viewport image quality. Naturally, the SVM classifier needs to be trained to obtain a model to predict the relative quality of a viewport obtained with any projection center parameter. As described in Chapter 3, a dataset of viewports of omnidirectional images, obtained with the GPP for a set of projection centers (including $d = 0$), along with the corresponding opinion scores (CMOS), was produced. However, at the time of this work, the GPP viewport dataset included only six omnidirectional images (as shown in Figure A.2); thus, only these images were used for SVM training and testing. The dataset was split into training and testing sets, and a common cross-validation procedure was applied. To perform the quality classification of a viewport obtained with a projection center d_q, the following steps are applied:

- **Viewport Labelling** - The viewports used on the subjective tests were labeled according to the subjective scores shown in Figure A.2: scores in the range]6,10] were labeled as G1; in the range [4,6] ere labelled as G2; in the range [0,4[were labelled as G3. If more than three groups were considered, it will be difficult to establish a clear boundary between them, as shown in Figure A.2; in that case, it would also be quite difficult to design an automatic classifier to separate the groups.

- **Feature Extraction** - Line bending features are extracted as described in Section 4.3, and for all viewports except the reference ones (those with $d = 0$).

- **Training Step** - The SVM model is trained using the extracted features and the corresponding ground truth viewport labels. The trained SVM model can be used to classify any viewport obtained from an omnidirectional image, using the GPP with a projection center parameter in [0,1].

- **Testing Step** - Using the SVM model that was built during the training step, the features extracted from test viewports are mapped to one of the pre-defined groups, for some projection center values. The same set of projection centers was used to evaluate the proposed SVM model performance.

Since three quality groups were considered - G1, G2 and G3 - this work adopted the one-versus-one-based multiclass SVM approach. Regarding the C soft-margin penalty cost, also known as the regularization constant, it was optimized by full search. This parameter controls the margin of the hyperplane that separate groups and helps to prevent overfitting [147]. As usual, all features are normalized (by subtracting the mean and dividing by the variance) to avoid that features with large values dominate (or influence) the SVM distance metric. Also, the SVM training and testing steps were conducted with a cross-validation procedure. It consists in splitting the dataset into subsets (or folds), then train the model on some subsets (training sets), and test the model on the remaining subsets (testing sets). In this work, the viewport dataset was split into 10 folds; after repeating the process 10 times, each fold was used exactly once as the validation data, and the predicted labels are kept. Thus, all viewports are used for both training and testing.

A.3 Performance Evaluation

In this section, the proposed model performance is evaluated with some test conditions; a comparison with a metric defined in [36] is also presented.

i) Test Conditions

To evaluate the proposed model, the following three different scenarios were considered:

1) **Scenario 1** - Using measures LC and NLC only.

2) **Scenario 2** - Using measures LI and NLI only.

3) **Scenario 3** - Using all measures: LC, NLC, LI or NLI.

These scenarios allow to evaluate the impact, on the overall performance, of different types of features independently, namely those that measure the line curvature and those that measure the line inclination. The following statistical measures were used to assess the metric performance: Accuracy (Acc), Precision (Prec), Recall (Rec), and F1 score (F1), which are typical performance measures in classification problems. The values for these metrics are computed from the classifier confusion matrix, as described in [148]. To find the best value of p% for poolings P_5^l, P_6^l, and P_7^l, a classifier was built with "Scenario 3" including only P_5^l, P_6^l, and P_7^l; several values of p% were considered, and the resulting Acc, Prec, Rec, F1 were obtained. The best performance was obtained for p = 90%.

Table A.1. SVM feature selection process.

Features	$\overline{\text{Acc}}$ %
$P_1^l(LC)$	52.8
$P_1^l(LC)$, $P_2^l(LC)$	64.6
$P_1^l(LC)$, $P_2^l(LC)$, $P_3^l(LC)$	67.4
$P_1^l(LC)$, $P_2^l(LC)$, $P_3^l(LC)$, $P_5^l(LC)$	68.1
$P_1^l(LC)$, $P_2^l(LC)$, $P_3^l(LC)$, $P_5^l(LC)$, $P_6^l(LC)$	67.4
$P_1^l(LC)$, $P_2^l(LC)$, $P_3^l(LC)$, $P_5^l(LC)$, $P_6^l(LC)$, $P_2^l(NLC)$	72.9
$P_1^l(LC)$, $P_2^l(LC)$, $P_3^l(LC)$, $P_5^l(LC)$, $P_6^l(LC)$, $P_2^l(NLC)$, $P_4^l(NLC)$	76.4
$P_1^l(LC)$, $P_2^l(LC)$, $P_3^l(LC)$, $P_5^l(LC)$, $P_6^l(LC)$, $P_2^l(NLC)$, $P_4^l(NLC)$, $P_7^l(NLC)$	75.1

Also, the SVM model complexity was reduced by performing feature selection, so that only the most relevant features are used. For each scenario, an SVM model is created for a set of features, using the default C parameter value. Initially, this set contains only one feature and then the remaining features are added one-by-one and its impact evaluated. A feature is not kept on the set if the overall metric performance is maintained or decreased. Table A.1 presents the resulting average accuracy ($\overline{\text{Acc}}$), when features are added one-by-one, for scenario 1; in this case, features $P_6^l(LC)$ and $P_7^l(NLC)$ were excluded. The same process was applied to scenarios 2 and 3. Also, several values were tested for parameter C, and the best value found was 1.5, for all scenarios under evaluation.

Since the NLC measure was already proposed in [36], and used in [10] with two pooling functions, P_2^l and P_4^l, it was used as benchmark to assess how much improvement were obtained with the proposed metric. As in experimental validation of [36], the NLC measure was used with pooling functions P_2^l and P_4^l. Since only two features are available, to assess the relative image quality a simple neareast neighboor classifier was used. First, the feature values are normalized. Then, the value of each feature is scaled to the range of [0,10]. In the training procedure, three clusters are obtained by dividing the data into the three quality groups. As in the viewport labelling, described in Section A.2, values in the range]6,10] are assigned to a cluster that represents G3, values in the range [4,6] are assigned to the G2 cluster, and values in the range [0,4[are assigned to the G1 cluster. After, the mean value is computed for each cluster. During the testing step, feature values are mapped into one of the three clusters by selecting the cluster which has the minimum distance between the feature value and the cluster mean. Since each cluster represents a group, a label can be assigned and the classification performance (using Prec, Rec, F1, and Acc) can be computed using the ground-truth groups.

ii) Experimental Results

Table A.2 presents the performance measures per group, and the corresponding average values, for all the three scenarios under evaluation. The selected features for each scenario and the benchmark classifier performance are also presented.

The following conclusions can be taken:

- **Scenario 1** - Features based on the line curvature (bending of image structures) are evaluated. The proposed metric provides the best results for G3 and the worst results for G2. This was expected since when other than rectilinear projections are employed in images

Table A.2. Overall model performance measures for scenarios 1, 2, 3, and for the benchmark solution ($\boldsymbol{NLC}$ metric).

Scenario: 1				
Features: $P_1^l(LC)$, $P_2^l(LC)$, $P_3^l(LC)$, $P_5^l(LC)$, $P_2^l(NLC)$, $P_4^l(NLC)$				
Groups	G1	G2	G3	Average
Prec %	83.3	73.6	87.3	81.4
Rec %	62.5	81.3	100.0	81.3
F1 %	71.4	77.2	93.2	80.6
Acc %	83.3	84.0	95.1	87.5
Scenario: 2				
Features: $P_1^l(LI)$, $P_2^l(LI)$, $P_3^l(LI)$, $P_5^l(LI)$, $P_6^l(LI)$, $P_2^l(NLI)$				
Groups	G1	G2	G3	Average
Prec %	84.6	71.7	92.3	82.9
Rec %	68.8	79.2	100.0	82.6
F1 %	75.9	75.2	96.0	82.4
Acc %	85.4	82.6	97.2	88.4
Scenario: 3				
Features: $P_1^l(LC)$, $P_2^l(LC)$, $P_3^l(LC)$, $P_6^l(LC)$, $P_2^l(NLC)$, $P_4^l(NLC)$, $P_7^l(NLC)$, $P_1^l(LI)$, $P_5^l(LI\ )$,$P_2^l(NLI\ )$				
Groups	G1	G2	G3	Average
Prec %	84.8	84.4	90.6	85.0
Rec %	81.3	79.2	100.0	86.8
F1 %	83.0	81.7	95.0	86.6
Acc %	88.8	88.1	96.5	91.2
$\boldsymbol{NLC}$ metric				
Features: $P_2^l(NLC)$, $P_4^l(NLC)$				
Groups	G1	G2	G3	Average
Prec %	39.3	38.9	40.8	39.7
Rec %	57.3	41.7	18.8	39.2
F1 %	46.6	40.2	25.7	37.5
Acc %	56.3	58.7	63.5	59.5

with long lines, a negative perceptual impact is usually observed, which is easy to predict from line curvatures.

- **Scenario 2** - Features based on the line inclination are evaluated. In this case, the classification has the best performance for G3 and the worst performance for G2 as in the previous scenario. This scenario provides slightly better performance than the previous one, increasing all average performance values for G1 and G3; however, the average performance decreases for G2.

- **Scenario 3** - In this scenario, 10 features are used. By combining all these features, the best performance (for all metrics) was achieved; a higher improvement was obtained for G2, for which is harder to identify the correct class from the features.

- **$\boldsymbol{NLC}$ metric standalone** - As shown in Table A.2, when only the NLC measure is used, the performance is poor and the metric cannot provide a reliable estimate of the geometric distortions; in particular, for G1, the precision is rather low. The main problem is that NLC

is normalized with the line length and thus, for lines with high length values, the curvature of the corresponding projected line does not have much influence. This makes harder for the classifier to distinguish the groups using this type of feature (especially after the pooling of all lines in the image). However, this metric was used for the optimization of the parameters of the Panini projection in [10] and was not perceptually validated.

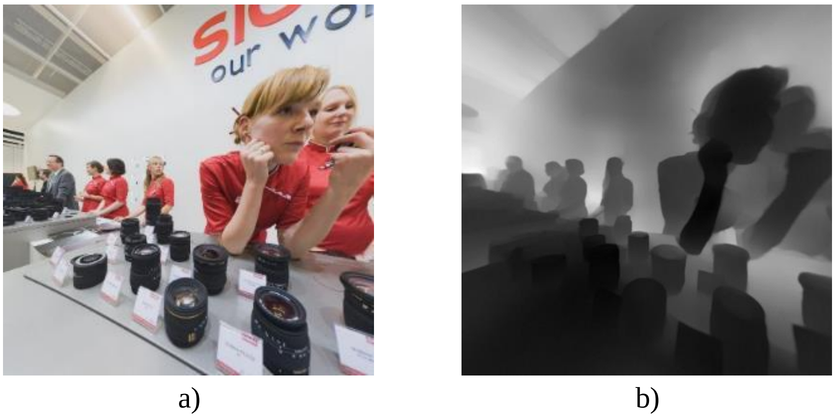
Concluding, the proposed model can evaluate the relative quality of viewports in all groups with a high performance. When both line curvature and inclination features are used, the proposed model achieves an average precision of 85.0%, average recall of 86.8%, average F1 of 86.6%, and an average accuracy of 91.2%. The proposed model has lower performance for G1 and G2 comparing to G3. In fact, group G3 is composed by two images having large buildings with several straight lines, that bent for $d \neq 0$, which is the main influencing factor for the perceived quality.

Annex B

Stretching Features Weighted by Depth Scores

This annex describes additional experiments that were performed to improve the stretching features proposed in Section 4.4.1.B, considering the viewport depth map instead of the viewport saliency map.

The viewport depth map was computed using the method proposed in [149], which is a deep learning-based approach designed for monocular depth estimation of 2D images. Figure B.1b) depicts an example of a depth map obtained for the viewport in Figure B.1a); this depth map is represented in grey scale, with values between 0 and 255 - the closest a region is to the camera, the lowest (and darkest) will be the corresponding pixel values in the depth map. To obtain the depth-based weighs, the grey level values are normalized to the range [0,1] and subtracted from 1; the stretching features are then weighted and evaluated as described in Section 4.4.2.A. Table B.1 presents the resulting PLCC values considering the saliency and the depth map. As shown, the stretching feature performance was not improved much with the use of depth-based weights.



a) b)

Figure B.1. a) Viewport rendered with rectilinear projection and with a square FoV of 110°; b) corresponding depth map obtained with the method proposed in [149].

Table B.1. Correlation (PLCC) between stretching feature and CMOS values, considering saliency-based and depth-based weighting.

Saliency-based weighting		**Depth-based weighting**	
SF_{dangle}	0.44	SF_{dangle}	0.46
SF_{dscale}	0.51	SF_{dscale}	0.54
SF_{darea}	0.54	SF_{darea}	0.58