Real-Time Image Processing manuscript

Luis Peñaranda · Luiz Velho ·
Leonardo Sacht

Real-time Correction of Panoramic Images using Hyperbolic Möbius Transformations

the date of receipt and acceptance should be inserted later

Abstract Wide-angle images gained a huge popularity in the last years due to the development of computational photography and imaging technological advances. They present the information of a scene in a way which is more natural for the human eye but, on the other hand, they introduce artifacts such as bent lines. These artifacts become more and more unnatural as the field of view increases.

In this work, we present a technique aimed to improve the perceptual quality of panorama visualization. The main ingredients of our approach are, on one hand, considering the viewing sphere as a Riemann sphere, what makes natural the application of Möbius (complex) transformations to the input image, and, on the other hand, a projection scheme which changes in function of the field of view used.

We also introduce an implementation of our method, compare it against images produced with other methods and show that the transformations can be done in real-time, which makes our technique very appealing for new settings, as well as for existing interactive panorama applications.

Keywords panorama · perspective projection · Möbius transformation · real-time · implementation

1 Introduction

An image is called panoramic when it represents a wide field of view (FOV, in photography, is the part of the world which is visible through the camera).

Some applications generate panoramic images from an image representing the *viewing sphere*, a sphere containing the scene, centered at the viewpoint. The viewing sphere is usually obtained with special cameras, software or a

IMPA – Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil
E-mail: {luisp,lvelho,leo-ks}@impa.br

combination of both. A panorama can be obtained by *stitching* together a set of regular pictures of a scene, producing an image inscribed on the surface of a sphere, and then the sphere projected on a plane. The entire process can be done by using programs like Hugin [6]. This software uses state-of-the-art methods to find common points in different images of the same scene, combine those images, correct colors and project the sphere. In the last years, however, cameras capable of directly obtaining a 360-degree sphere became popular; these cameras directly output a spherical image, without the need of special software to combine pictures [15]. There also exists the well-known fisheye lens [10], which permits to capture a 180-degree image directly on a plane.

There exists a number of transformation techniques to obtain a panoramic image from the viewing sphere (typically, a projection from the sphere to a plane). Each type of transformation has distinct properties, and the goal of using different transformations is usually to obtain a more realistic image. The subjective concept of *realistic* can be interpreted in different ways, typically as bending straight lines as less as possible or as preserving angles of the scene. Since lines and angles cannot be preserved at the same time [25], different projection schemes were proposed. Warping techniques were also developed [5,16]; however, they need long human interaction or optimization methods and thus cannot be used for real-time nor interactive applications.

While one kind of transformation might be good for some setting, it might be bad for other. This situation becomes evident on some interfaces where a user is able to interactively change the FOV of a scene. Usually, perspective or equirectangular projections (both known for providing good results for small to medium FOVs) are used for interactive visualizations. When the FOV becomes very wide, projections are perceptually less realist.

What we propose in this paper is to adopt a new approach in interactive applications for obtaining a panoramic image when the FOV becomes wide. When the user widens the FOV until surpassing a limit where common projections tend to produce bad results, we propose to simulate the widening of the FOV by performing a Möbius transformation, and then apply the initially intended common projection. The novelty of this approach is that we do not introduce a complicated projection method but, instead, we perform a transformation on the viewing sphere and then we project using a well-known perspective projection. Performing transformations of the image directly on the viewing sphere is specially important in some settings, on which it is not desired to project on a plane (for instance, when projecting on a dome) and, as far as we know, is also a novelty of our method.

Our main contribution is a projection method which performs a transformation on the viewing sphere, which makes it applicable on non-plane projections. Moreover, our method can be used in real-time applications. We also use a powerful tool such as complex transformations for our purpose, showing that they can be employed in Image Processing and open the quest for new applications in the field. Finally, we present an open-source implementation of the method.[1]

[1] Note to the reviewer: the implementation will be open-sourced when the paper is published.

A preliminary version of this work was published in [17]. The present full-version includes the theoretical foundations of the method, new experiments (comprising comparison with more state-of-the-art methods), argues about perceptual properties of the generated images, adds a discussion on the applications of the new technique and motivates future research. Additionally, the text was entirely rewritten.

The roadmap of the paper is as follows. Next Section argues about previous work on the field. Section 3 introduces some mathematical definitions to understand the rest of the paper. Our approach is formalized in Section 4. Sections 5 and 6 provide an analysis of the technique, while Section 7 introduces some scenarios where our technique can be applied. Finally, Section 8 discusses current work and future research directions.

2 Previous Work

Due to their increasing popularity and interest, panoramic images have become a theme of intense discussion in the Computer Graphics and Image Processing communities in the last twenty years. The impossibility of obtaining a global projection from the sphere to the plane that preserves all possible straight lines and object shapes was shown in the seminal work by Zorin and Barr [25]. This theoretical limitation motivated much research for obtaining perceptually realistic panoramas.

One approach for this problem was to use different perspective projections in the same scene [24]. The user specifies different projection planes and view directions to define the different projections. The discontinuities caused by using different projections for different regions of the panorama were hidden (if possible) by choosing the projection planes in a way that fit well orientation discontinuities that were already present in the scene.

Other approach consisted in investigating near-perspective projections used by ancient painters, such as Pannini projection [18]. This technique produces very good results, as proved in some ancient paintings.

Conformal mappings were also explored with the aim of preserving the shape of the objects in a panoramic image [8]. This approach consisted in investigating the stereographic projection and scaling of the complex plane, but only for artistic and exploratory purposes. Since the focus was on shape preservation, results present bent lines. In our work, we go beyond and improve these ideas to map wide fields of view to narrower ones and add a perspective re-projection step, which produces better quality results because the straight lines are less bent.

Other methods relied on both user interaction and energy-minimization formulations. Carroll *et al.* [5] used the important lines in the scene provided by the user and detected faces to control straight line preservation and conformality in these regions. Kopf *et al.* [11] used regions specified by the user where the projection should be nearly planar to formulate their optimization framework. Wei *et al.* [23] formulates the problem as the minimization of a quadratic energy based on user annotation, which has a closed form and implies in the solution of a sparse linear system. In these approaches, user

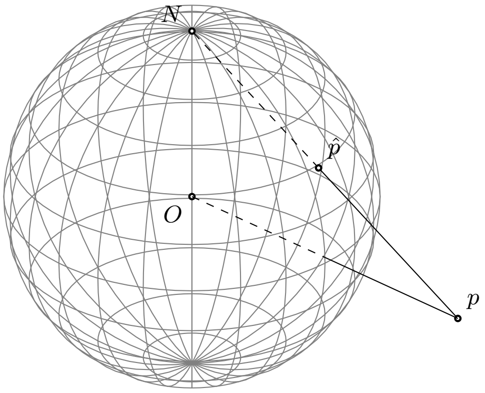


Fig. 1: Correspondence between a point $\hat{p}$ on the Riemann sphere and a point p on the complex plane.

interaction was usually laborious and the optimization formulations made them impossible to be implemented in real-time[2].

An interesting approach consisted in dynamically changing the projection used depending on the FOV [12]. This can also be achieved by our viewer by applying different shrink values as the FOV of the perspective projection changes, as explained in Section 4. Also, our visualization simulates better camera movements since it is a natural generalization of the perspective projection.

Another advantage of our method compared to previous ones is that we do not rely on heavy user interaction. The user is only asked to vary one parameter (which may be set automatically to a good value), what makes our panorama viewer a pleasant experience, instead of laborious. Also, our formulation is simple and does not rely on heavy optimizations, which makes possible to implement our method in real-time.

3 Definitions

Before describing our method, we formalize in this Section the ideas presented so far. We assume the given image is inscribed in the viewing sphere. The viewing sphere will be considered a Riemann sphere. Each point on the sphere corresponds thus to the representation of a complex number; Figure 1 shows the correspondence between points in the complex plane and on the Riemann sphere. A complex number $p = x + iy$ is a point on the complex plane, with coordinates x and y. This plane is the stereographic projection of the Riemann sphere. The point p in the complex plane is represented on the sphere by the point $\hat{p}$. p and $\hat{p}$ are related by stereographic equations. A complete study of stereographic projections is beyond the scope of the present

[2] For us, *real-time* means that the computations are done quicker than the rendering of a frame in the application. Even if the other methods are fast, they are not suitable to be implemented in real-time.

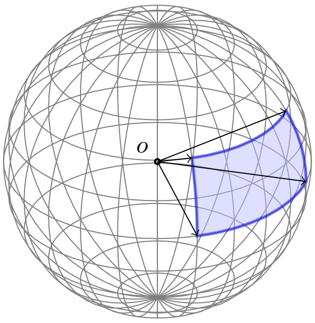


Fig. 2: The viewing sphere. The FOV (blue region) is to be mapped to a rectangular image. o is the observers position, in the center of the sphere.

work. We will only show the formulas of stereographic projections here and refer to Snyder's book [20] for details.

The stereographic projection maps complex points in points on the viewing sphere. It is defined as follows.

$$\begin{aligned} \mathbf{S} : \mathbb{S}^2 \setminus \{(0,0,-1)\} &\to \mathbb{C} \\ \hat{p}(\hat{x},\hat{y},\hat{z}) &\mapsto p\left(\tfrac{2\hat{x}}{\hat{z}+1}, \tfrac{2\hat{y}}{\hat{z}+1}\right) \end{aligned} \tag{1}$$

On the other hand, the inverse stereographic projections maps points on the viewing sphere to complex points, and is defined as follows.

$$\begin{aligned} \mathbf{S}^{-1} : \qquad \mathbb{C} &\to \mathbb{S}^2 \setminus \{(0,0,-1)\} \\ p(x,y) &\mapsto \hat{p}\left(\tfrac{4x}{x^2+y^2+4}, \tfrac{4y}{x^2+y^2+4}, \tfrac{x^2+y^2-4}{x^2+y^2+4}\right) \end{aligned} \tag{2}$$

We assume the observer stands in the center of the viewing sphere. She can do two basic things: rotate her head in any direction or change the FOV. For the sake of simplicity they will be studied separately, but it would not require much extra work to consider them together.

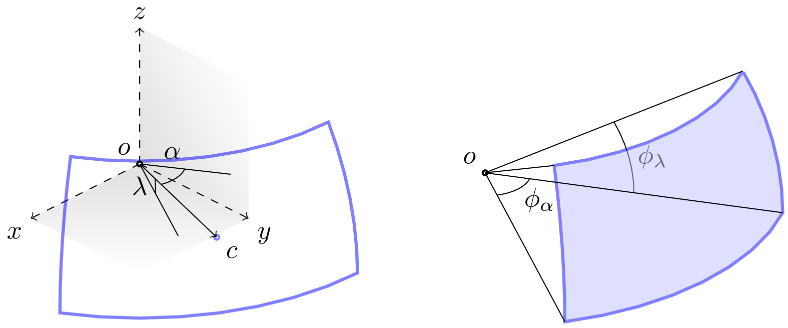
The rotation of the observer's head is represented as two angles, α (azimuth) and λ (altitude). These angles represent uniquely a point on the sphere, which is associated with the center of the produced image, see Figure 3a.

The FOV is again represented as two angles, ϕ_α (azimuthal FOV) and ϕ_λ (altitudinal FOV), as depicted in Figure 3b. However, the ratio between these two angles is defined by the aspect ratio of the output image.

The problem consists then in obtaining a plane image of the FOV (the part shaded in blue in Figure 2) in function of α, λ, ϕ_α and ϕ_λ.

The parameters α, λ and ϕ_α (ϕ_λ can be considered a function of ϕ_α) determine the points on the sphere to be processed. Recall that each point on the sphere is uniquely determined by ϕ_λ and ϕ_α (because the radius of the sphere is fixed) or by its cartesian coordinates in the original plane.

To study the variation of ϕ_α, we assume α and λ fixed to zero. This way, a value of ϕ_α uniquely determines the four points on the sphere which are the



(a) **oc** is the eyesight direction. α is the angle between vector **oc** and plane yz. λ is the angle between vector **oc** and plane xy.

(b) Modelling FOV: ϕ_α and ϕ_λ.

Fig. 3: Details of the viewing sphere and the view region.

vertices of the blue region of Figure 2. For a given ϕ_α, the FOV is computed, and points on the sphere are mapped to the output image by a projection function (which usually depends on the FOV).

Changes in viewing directions are given by changing α and λ. Since the FOV does not change, these changes can be modeled as rotations of the sphere. This approach has the advantage that the projection function does not need to be recomputed, although it must be composed with a rotation function on the sphere.

Finally, let us mention that the perspective projection is a key ingredient of our method. Since it is a well-known projection method, we will not describe it, but only mention that it is defined by the following equation.

$$\begin{aligned} \mathbf{P} : \mathbb{S}^2 \setminus \{z < 0\} &\to \mathbb{C} \\ \hat{p}(x, y, z) &\mapsto p\left(\tfrac{x}{z}, \tfrac{y}{z}\right) \end{aligned} \tag{3}$$

In practice, Equation 3 is scaled, in order for a given FOV to be projected in the extents of an image. In the sequel, we will denote a function performing a perspective projection of a FOV ϕ as $\mathbf{P}_\phi$. We refer the reader to the book by Snyder [20] for more details and a complete study of perspective projections.

4 The Möbius Projection Method

This Section introduces our method, aimed to visualize images with wide FOVs. Perspective projections work very well in practice for small to medium FOVs. When ϕ_α grows to a big value, it becomes difficult to conceive a good projection function. One approach is to replace the perspective projection $\mathbf{P}_{\phi_\alpha}$ by a warping which respects visual restrictions, such as conformality and preservation of straight lines, as much as possible. We refer to [16] for details on such methods. The drawback of these methods is that they are

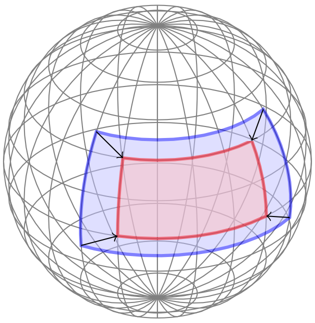


Fig. 4: The Möbius transformation, shrinking points in the new FOV (in blue) to the old FOV (in red).

slow and they usually need human interaction, what makes them unsuitable for interactive applications.

Our approach consists in using two different projection methods, depending on the desired FOV. To the best of our knowledge, the only work on which the projection scheme used varies as a function of the FOV is the paper by Kopf *et al.* [12]. They use an adaptive projection resulting from an interpolation between a perspective and a cylindrical projection.

We assume that there exists a value $\phi_{\max}$, such that the perspective projection does not introduce big distortions when $\phi_\alpha \leq \phi_{\max}$. When this happens, we propose to use a direct perspective projection, that is, without performing any shrink.

On the other hand, when $\phi_\alpha > \phi_{\max}$, the perspective projection fails to provide a realistic image. We propose, in this case, to perform a hyperbolic Möbius transformation [14] on the sphere points, in order to map points of the new FOV region to the old FOV region, as depicted in Figure 4 (this is achieved by first mapping the sphere points to the complex plane, then applying the Möbius transformation to the complex plane and, finally, lifting the points back to the sphere). After, the same projection $\mathbf{P}_{\phi_{\max}}$ maps points on the sphere to the output image.

Our projection schema is non-conformal and does not preserve areas (it inherits these properties from the perspective projection). In addition, since hyperbolic Möbius transformations do not preserve great circles on the Riemann sphere, our method looses the line-preservation property from perspective projections. We will show in Section 6 that, despite these properties, our method produces indeed very realistic images.

4.1 Computing the Shrink

As mentioned earlier, the problem is to find the Möbius transformation that shrinks the new FOV into the old FOV, as depicted in Figure 4. If we had the point c of Figure 3a coinciding with one of the poles, we could simply apply an hyperbolic transformation to shrink the FOV. This can be accomplished by rotating the sphere and making c coincide with the south pole (corresponding

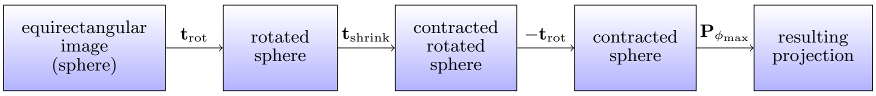

equirectangular image (sphere)
$\mathbf{t}_{\text{rot}}$
rotated sphere
$\mathbf{t}_{\text{shrink}}$
contracted rotated sphere
$-\mathbf{t}_{\text{rot}}$
contracted sphere
$\mathbf{P}_{\phi_{\max}}$
resulting projection

Fig. 5: Pipeline of the shrinking process. Recall that $\mathbf{t}_{\text{shrink}} = S^{-1} \circ M_s \circ S$.

to the origin on the complex plane). Mathematically, this means to do a translation by a vector $\mathbf{t}_{\text{rot}}(-\alpha, -\lambda)$ in polar coordinates (note that this translation must not be done in the equirectangular image). Without loss of generality, we assume in the rest of the paper that $\alpha = \lambda = 0$.

The following step is to perform a shrink centered in the origin. This can be accomplished by using a hyperbolic Möbius transformation. The only knowledge about Möbius transformations needed to understand the method is that the shrink transformation we need takes the form $M_s = \rho z$. We refer the reader to Appendix A for some more details on Möbius transformations.

At this point, after being introduced to Möbius transformations, the reader should ask why we emphasize that our method uses them, while it is only a polar scaling. The reason is that we believe this kind of transformations has a myriad of applications in Image Processing, and we want to encourage further research on them.

It remains to compute ρ. Since $\phi_{\max}$ and ϕ are angles, it is direct to compute, in the equirectangular domain, $\rho = \frac{\phi_{\max}}{\phi}$. But M_s is defined for the points on the complex plane: this means that a stereographic projection S must be first applied to the points on the sphere, then the old FOV is mapped to the new FOV using M_s and, finally, an inverse stereographic projection S^{-1} is used to map back points to the sphere. This means that the shrink transformation takes the form $\mathbf{t}_{\text{shrink}} = S^{-1} \circ M_s \circ S$. (Note that, if α and λ are not assumed to be zero, the translation $\mathbf{t}_{\text{rot}}$ must be applied before $\mathbf{t}_{\text{shrink}}$, and the the inverse translation $-\mathbf{t}_{\text{rot}}$ must be applied after.)

Finally, the desired projection is obtained by applying the transformation $\mathbf{P}_{\phi_{\max}}$ to the resulting shrink. The pipeline of the process is shown in Figure 5.

The shrink is a good approach to reach wide FOVs on panoramas. However, even with the shrink/projection scheme, a FOV close to 2π will produce an unrealistic output image, as well as introducing unacceptable distortions. It becomes interesting thus to determine the biggest ϕ that can be visualized with this technique, as well as computing a good value for $\phi_{\max}$. The first of these shall be determined by experimentation, while the second can be approximated by a function of ϕ, as well as determined experimentally. The next Section addresses the computation of $\phi_{\max}$.

5 A Perceptually-Good Value of $\phi_{\max}$

In this Section, we will argue about the value of the parameter $\phi_{\max}$. As mentioned earlier, a value of $\phi_{\max}$ can be determined experimentally, given a value of ϕ.

We will show first that our Möbius shrinking scheme produces the same distortion than the stereographic projection by setting $\phi_{\max} = \epsilon$, a small value. As explained in Section 4.1, given the image inscribed in the viewing sphere, our method $\mathbf{M}$ can be seen as a composition of functions (we will assume again, without loss of generality, that the sphere does not need to be rotated, in order to simplify the explanation).

$$\begin{aligned}\mathbf{M} &= \mathbf{P}_\epsilon \circ \quad \mathbf{t}_{\text{shrink}} \\ &= \mathbf{P}_\epsilon \circ S^{-1} \circ M_s \circ S \\ &\approx \; S \;\circ S^{-1} \circ M_s \circ S \\ &= \qquad\qquad M_s \circ S \\ &= \qquad\qquad\qquad S\end{aligned} \tag{4}$$

The approximation (in the third line of the equation) is due to the fact that ϵ is very small and, in this case, the perspective projection $\mathbf{P}_\epsilon$ is almost equal to the stereographic projection S. The last equality is forced by the fact that $M_s \circ S$ means to apply the stereographic projection to the sphere and then shrink the resulting image; since this shrink does not deform the image, we conclude that, in this setting, M_s has no effect. This result will be corroborated experimentally in Section 6.

Preservation of lengths and angles is important for many applications such as cartography, but the stereographic projection is not ideal for panorama visualization since it does not take into account perceptual aspects. In the sequel we propose a choice of $\phi_{\max}$ that approximates better human perception of the world.

It is known that the way images are formed on the human retina is very similar to a perspective projection. Furthermore, our eyes only see with clarity objects that are in the region called *central* FOV, which comprises 60 degrees or less [1,22]. We use this fact to propose the value $\phi_{\max} = 60°$. Using such value will map points on the portion of the viewing sphere to be visualized to a 60 degrees FOV and apply a perspective projection to this limited FOV, simulating then what one would see if was immersed in the Möbius transformed sphere.

5.1 Discussion

We stress that the choice $\phi_{\max} = 60°$ does not guarantee straight lines in the scene to be straight in the final result. This is due to the fact that the Möbius transformation on the sphere does not map segments of great circles (projection of straight lines on the sphere) to segments of great circles. If the user wants straight lines to be better preserved, it is still possible to set a higher value to $\phi_{\max}$. On the other hand, diminishing this value results in a better preservation of lengths and angles. This aspect is further discussed in the Section 6 and in the accompanying video.

Since lines are not preserved, it is of interest to question if there are some lines which are preserved, and the perceptual implications of the answer. To begin this study, let us clarify how pieces of straight lines (segments) on the scene are represented on the viewing sphere. One segment on the scene and the center of the sphere form a triangle. The intersection of two sides of the triangle with the sphere determines two points. The representation of the segment on the viewing sphere is the geodesic joining these two intersection points. The triangle is contained on a plane, and the intersection of this plane with the viewing sphere is a great circle. In general, straight lines on the scene are represented on the viewing sphere as great circles or arcs of great circles.

Under a hyperbolic Möbius transformation, some circles on the sphere are preserved, and some not. The great circles passing by the north pole[3] remain unchanged under hyperbolic Möbius transformations. The circles parallel to the Equator are mapped to different circles parallel to the Equator. The rest of the circles lose their shape when applying hyperbolic Möbius transformations. This analysis permits to see that, in our method, straight lines are deformed at different extents depending on the distance of their projections on the viewing sphere to the shrinking point.

It remains to discuss about how bending lines affect perception. It is widely accepted that a projection is better when it bends lines less, but let us postulate that this is not quite true in human perception. It is known that the human vision is very complex, far beyond the scope of this paper, and that the brain reconstructs the image captured by the eyes to form a three-dimensional model of the reality. Historically, camera lenses were manufactured to try to produce perceptually-good images. One meaningful example on this are the fisheye lenses for very wide FOVs, in which lines are bent. We believe in this respect that the line bending of our method is perceptually-acceptable for very wide FOVs. This fact will also be corroborated in the next Section.

Let us mention, to conclude the Section, that one approach to partially reduce line bending is to consider cylindrical projections. Cylindrical projections do not bend vertical lines, but the problem remains intact for horizontal lines. A direction to explore in the future would be to combine Möbius transformations with cylindrical projections, as Kopf *et al.* [12] do with perspective and cylindrical projections.

5.2 A Final Note on Distortion

It would be interesting to also perform the analysis from an orthogonal point of view, that is, a completely automated method which does not require human interaction. As far as we know, the only method aimed to quantify the quality of a plane projection of a sphere is due to Milnor [13]. Briefly, this methodology computes the *scale* of each pair of points on a projection (the scale is defined as the ratio between the geodesic distance on the sphere and the projected distance of them). The distortion is the logarithm of the ratio between the smallest and the biggest *scales* of the projection.

We used Sage Math [21] to implement the formulas of the Milnor distortion. For our experiments, we used a set of points on a grid of 500×500 points on the surface of the unit sphere. We repeated, for FOVs between 1° and 180°, the following experiment. We considered all the pairs of distinct points

[3] Recall that, in our method, the viewing sphere is rotated to make coincide the north pole with the center of stretching, and rotated back after the stretching. For simplicity, we will assume here that the shrinking occurs with the north pole as center, without considering rotations.

Table 1: Frame rates obtained by our technique while the user interacts, for different mesh resolutions (varying on the rows) and different input equirectangular image resolutions (varying on the columns). These results were generated in a screen resolution of 1024 × 768 pixels in a PC with an Intel Xeon Quad Core 2.13GHz with 12 GB of RAM and a GeForce GTX 470 GPU.

	4000 × 2000 pixels	8000 × 4000 pixels
200 × 200 vertices	93 fps	89 fps
400 × 400 vertices	85 fps	84 fps
800 × 800 vertices	35 fps	33 fps

of this set (inside our FOV). We computed the projections (using many different projection methods, including ours) of each point and, for each pair of points, we computed the *scale* under each projection method. This procedure let us compute the Milnor distortion, for a given FOV, for all different types of projections. We compared then the distortions for each FOV. The results were not the expected: the quantitative Milnor distortion of the images does not correspond to their perceptual quality. This fact opens a research direction for the future: to conceive quantitative methods for projections, consistent with the human perception.

One alternative to our experiment is to choose the set of points to compute Milnor distortion on by studying the content of images. However, the results will be associated with a specific image and will not be as general as we wanted when designed our experiment. We believe it deserves a full research paper to study quantitative distortions of specific images.

6 Implementation and Experiments

While in theory our method shows advantages over direct projections, we implemented it to see how it performs in practice and the perceptual quality of the produced images. We coded in C++, using OpenGL to deal with image operations and texture the surface of the viewing sphere, and Qt for the interface, what asserts portability. All geometric transformations on the vertices of the sphere can be performed in parallel, what made us implement them as a GLSL vertex shader.

The input of the shader algorithm is the viewing sphere, with the equirectangular image as a texture. Each one of the vertices forming the triangular grid approximating the sphere is processed by the vertex shader, computing its position on the projection plane (applying the operations described in Figure 5 on Page 8). Finally, the fragment shader then interpolates the texture values inside the resulting triangles, forming the output image. Algorithm 1 shows the pseudocode of the algorithm we implemented in the vertex shader. It should be noted that the shader works with homogeneous points but we used, in the pseudocode, three-dimensional points, since the homogeneous

Algorithm 1: Möbius projection method

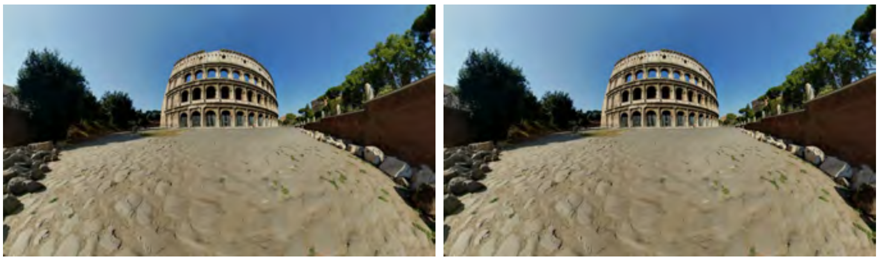
```
Input  : coordinates $(x, y, z)$ of a point in a plane image, angles $\lambda$ and $\phi$
         and a scale $s$ to scale the complex plane
Output : coordinates $(x_t, y_t, z_t)$ of the point, transformed using the Möbius
         projection method
// rotation of $-\lambda$ on the $xz$ plane
$(x, y, z) \leftarrow \big(\cos(-\lambda)x - \sin(-\lambda)z, y, \sin(-\lambda)x + \cos(-\lambda)z\big)$;
// rotation of $-\phi$ on the $xz$ plane
$(x, y, z) \leftarrow \big(x, \cos(-\phi)y - \sin(-\phi)z, \sin(-\phi)y + \cos(-\phi)z\big)$;
// stereographic projection
$(u, v) \leftarrow \big(2x/(1-z), 2y/(1-z)\big)$;
// convert from Cartesian to polar coordinates
$(\rho, \theta) \leftarrow \big(\sqrt{u^2 + v^2}, \arctan(u, v)\big)$;
// scaling the complex plane according to $s$
$\rho \leftarrow \rho s$;
// mapping back from polar to Cartesian coordinates
$(u, v) \leftarrow \big(-\rho\sin(\theta), \rho\cos(\theta)\big)$;
// mapping back from the complex plane to the unit sphere
$(x, y, z) \leftarrow \big(4u/(u^2 + v^2 + 4), 4v/(u^2 + v^2 + 4), (u^2 + v^2 - 4)/(u^2 + v^2 + 4)\big)$;
// perspective projection
$(x_t, y_t, z_t) \leftarrow \big(x/(-z), y/(-z), z\big)$;
return $(x_t, y_t, z_t)$;
```

component always equals one in our case. In addition, we can safely assume the z-coordinate of the input points to be zero.

The implementation consists in a window showing the image processed through our technique. The interface can be appreciated in the video accompanying this paper. It permits to pan the image with the mouse, as well as adjusting the FOV and the parameter $\phi_{\max}$ with the mouse wheel (the latter holding the Shift key). All these operations are done in real-time on a desktop PC, as illustrated in Table 1 on Page 11. The code is available on the accompanying material of the paper and will be open-sourced soon.

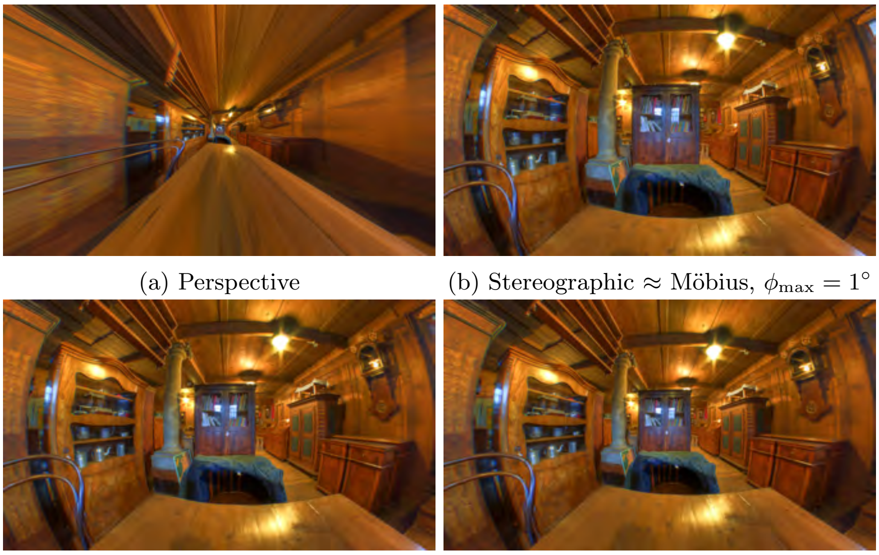
We reproduce here some results of our tests. First of all, let us corroborate the affirmation of Section 5, which stated that our method produces similar images than the stereographic projection. In Figure 6, the Roman Coliseum serves us to show that the stereographic projection and our method produced images which are indistinguishable for the human eye. This figure was produced using a very wide FOV.

Figure 7 shows how our method improves the quality of the images near the borders with respect to the perspective projection, while keeping the image on the center undistorted. From this figure, as well as from the video accompanying the paper, one can realize a meaning of the parameter $\phi_{\max}$. In the figure, line bending is easily distinguishable, since the scene is full of straight lines. Moreover, it is evident how lines passing by the center of shrinking are preserved.



(a) Stereographic projection (b) Möbius, $\phi_{\max} = 1° \approx 0.02\text{rad}$

Fig. 6: The stereographic projection and the Möbius method with a small value of $\phi_{\max}$ produce the same results, as predicted in Equation 4, on Section 5. In this case, $\phi = 240° \approx 4.2\text{rad}$.

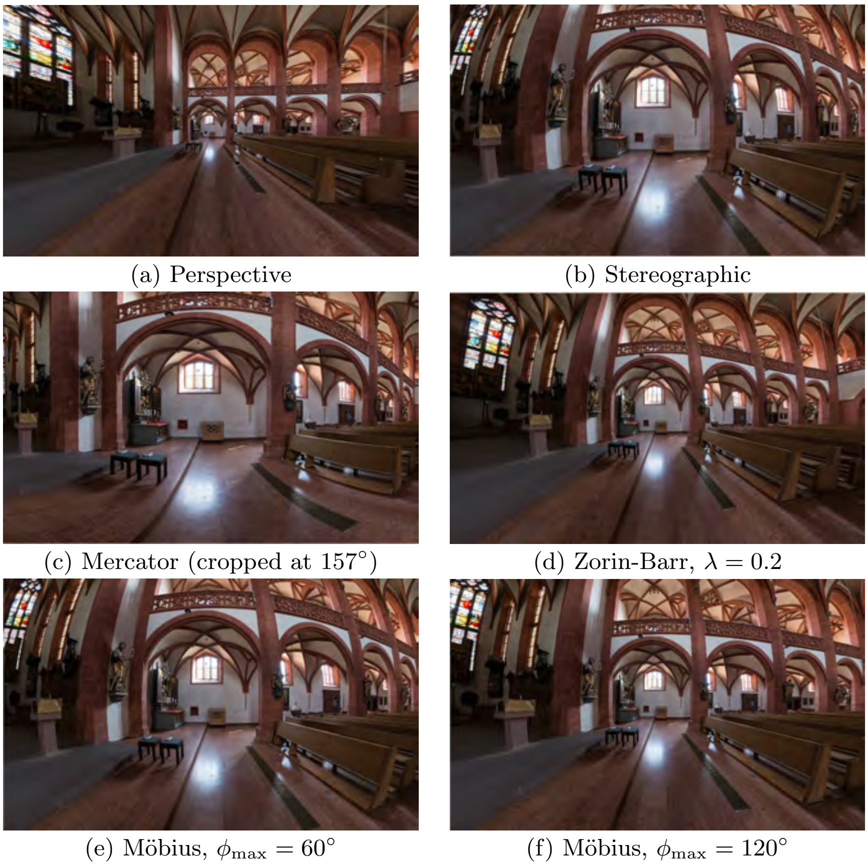


(a) Perspective (b) Stereographic $\approx$ Möbius, $\phi_{\max} = 1°$

(c) Möbius, $\phi_{\max} = 60°$ (d) Möbius, $\phi_{\max} = 90°$

Fig. 7: It is shown how the parameter $\phi_{\max}$ controls line bending. In all subfigures, $\phi = 172° \approx 3\text{rad}$.

A big value of $\phi_{\max}$ (bigger than or equal to the FOV) produces a perspective projection. Small values bend the lines on the scene, but correct artifacts of the perspective projection. Smaller values bend lines more and correct more artifacts. From the perceptual point of view, bent lines are by far more acceptable than distorted images, as can be appreciated in Figure 7. We can see $\phi_{\max}$ as a parameter which controls the ratio between line straightness and distortion near the borders. The price paid for obtaining images which are clear close to the borders is, in this case, loosing the preservation of straight lines.



(a) Perspective

(b) Stereographic

(c) Mercator (cropped at 157°)

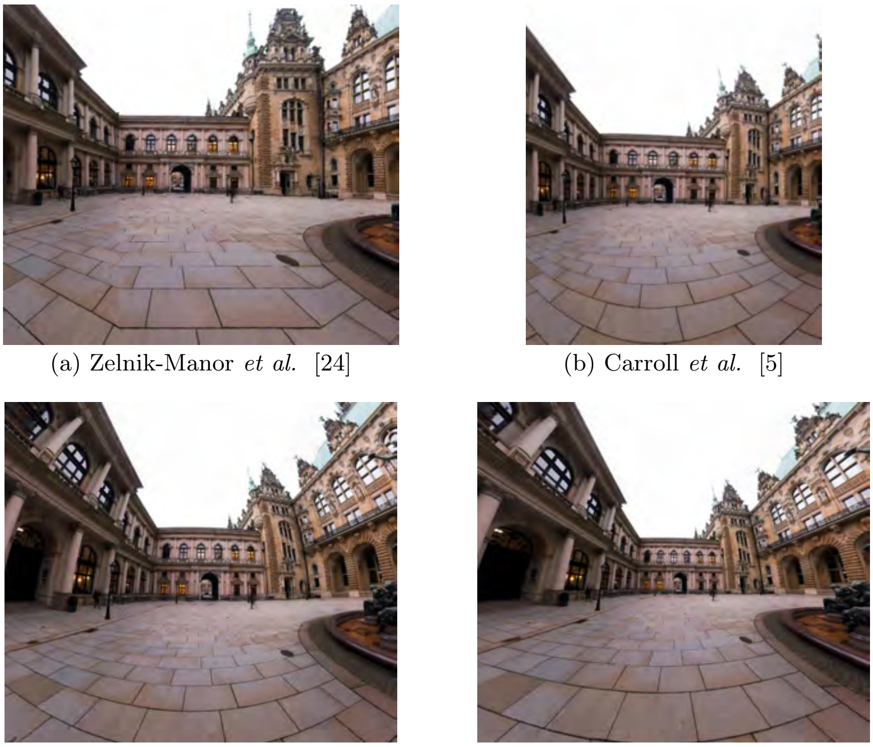
(d) Zorin-Barr, $\lambda = 0.2$

(e) Möbius, $\phi_{\max} = 60°$

(f) Möbius, $\phi_{\max} = 120°$

Fig. 8: Comparison with standard projections for $\phi = 160°$. Our method, (b), (e) and (f), has a good balance between straight line and object shape preservation, obtained by adjusting the parameter $\phi_{\max}$.

We also compare our technique to previous methods in the field of generation of panoramic images. We use first a view of the Rheingauer Dom to show in Figure 8 a comparison with standard projections from the viewing sphere to the image plane. We observe that for the field of view of 160 degrees the perspective projection (Figure 8a) distorts objects too much but preserves all straight lines. The stereographic projection, equal to the Möbius method when $\phi_{\max} = 1°$, (Figure 8b) improves the perceptual quality of the image by bending some lines. The Mercator projection (Figure 8c) preserves object shapes because it is conformal, but bends much the lines. The perspective projection corrected with the Zorin-Barr transformation, using parameters $\lambda = 0.2$ and $R = 0.5$ (Figure 8d), yields an image with a reasonable straight line preservation, but presents artifacts near the borders. The Möbius projection using the value of $\phi_{\max} = 60°$ (Figure 8f)



(a) Zelnik-Manor *et al.* [24]

(b) Carroll *et al.* [5]

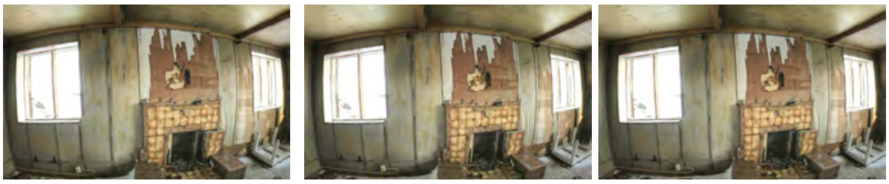
(c) Möbius, $\phi_{\max} = 60°$

(d) Möbius, $\phi_{\max} = 110°$

Fig. 9: Comparison with recent methods for $\phi = 150°$. In the result obtained by the method in subfigure (a) [24] the different perspective projections used for different areas of the image appear clear and unpleasant on the floor of the scene. The method in subfigure (b) [5] preserves all straight lines marked by the user, but fails to preserve the ones on the floor (which are too many to be marked by the user). Note also that marking a large number of lines would impose too many constraints on the optimization process and could compromise the final result. Subfigures (c) and (d) were produced with our method. In subfigure (c) we use the value of $\phi_{\max}$ we consider perceptually good. Subfigure (d) shows how the bending of lines on the floor was slightly corrected by augmenting the value of $\phi_{\max}$, but without obtaining a perceptually better image.

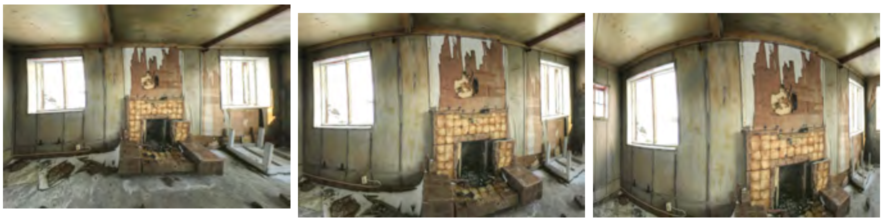
produced a perceptually good result. Finally, we set the parameter $\phi_{\max}$ to 120° (Figure 8f) to obtain an image with less line bending.

In Figure 9 we compare our method with recent works on the same topic with a view of the courtyard of the Hamburg town hall. The result obtained by the technique proposed by Zelnik-Manor *et al.* [24] (Figure 9a) shows discontinuities on the floor produced by using different projections for different areas of the image. This strategy works successfully for the building in the image, since these discontinuities are hidden by natural orientation



(a) $\phi_{\max} = 1°$ (b) $\phi_{\max} = 30°$ (c) $\phi_{\max} = 90°$

Fig. 10: Möbius, $\phi = 120°$



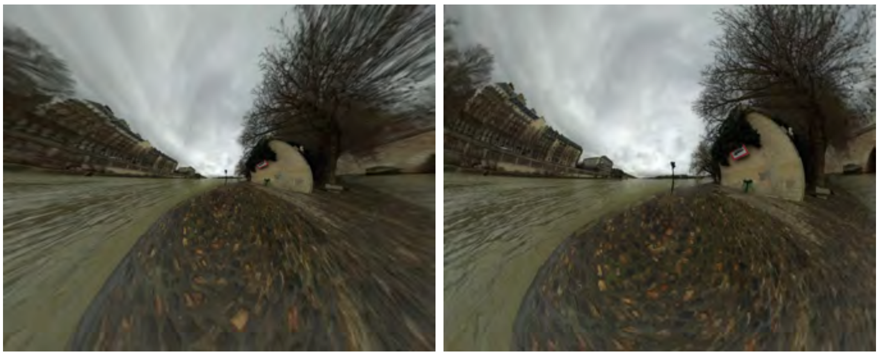
(a) $d = 0$ (b) $d = 1$ (c) $d \to \infty$

Fig. 11: Pannini, $\phi = 120°$

discontinuities in the scene, but it fails to fit the geometry of the floor. In Figure 9b we show a result produced by our implementation of the technique by Carroll *et al.* [5]. All straight lines specified by the user (see their work for more details) are well-preserved, but the lines on the floor appear bent, since they are too many to be marked by the user. Although their energy minimization formulation guarantees conformality and smoothness of the final result, it has the problem of taking some seconds to be performed. Our method (Figures 9c and d), does not rely on heavy user interaction nor on any optimization and is not restricted to scenes with any particular geometry.

Finally, we compare our approach with the Pannini projection. This projection method works as a cylindrical projection, but moving the projection center away from the center of the sphere. The projection center c is moved along the ray originating at the midpoint o of the cylinder central axis, in a direction perpendicular to the projected plane. Assuming the cylinder has radius 1, the parameter d represents the distance from o to c. This parameter d controls the image *compression*. If $d = 0$ (*i.e.*, o and c coincide), the resulting projection is a central cylindrical projection. If $d = 1$ (*i.e.*, c is on the surface of the cylinder, behind o), the projection is stereographic cylindrical (also called *basic Pannini*). When $d \to \infty$, the resulting projection is cylindrical orthographic. This way, when d grows, points in the borders of the resulting image are not distorted as in the perspective projection, producing perceptually pleasant results. Since it is a cylindrical projection, Pannini has the property of preserving all vertical lines.

Figures 10 and 11 show the effect of applying our method and the Pannini projection, both with different paramenters, to an image of a room with a FOV of 120° (the latter produced using an open-source implementation of



(a) $\phi_{\max} = 40° \approx 0.7\text{rad}$

(b) $\phi_{\max} = 1° \approx 0.02\text{rad}$

Fig. 12: Even if our method can handle very wide FOVs and displayed objects are distinguishable, the image seems unrealistic. Here, $\phi = 320° \approx 5.6\text{rad}$.

the Pannini method, provided by the Panotools software [7] called through Hugin [6]; please note that the Pannini projection was also implemented by its authors as a standalone package [19]). These two figures show the effect on line bending obtained by varying the value of the parameters of each method. We note first one of the strengths of the Pannini projection, the preservation of vertical lines. For the rest, it can be considered that $\phi_{\max}$ and d are somewhat analogous. We can also observe that big values of d allow for more line bending than small values of $\phi_{\max}$, but this fact does not mean that images obtained with big values of d are perceptually better.

To conclude this Section, we recall that our method produced very good results for all the tested cases, and was succesfully compared to state-of-the-art projection and warping methods. Nevertheless, we should point out that there are still some cases for whose our method cannot produce realistic results. For instance, Figure 12 shows what happens with a very wide FOV, close to $2\pi\text{rad} = 360°$, in a panoramic taken from the Île Saint-Louis in Paris. A perspective projection will simply not work in this case. Our method can produce a clear but unrealistic image. We believe that this cannot be overhauled, since the human vision is unable to observe such a wide FOV. The fact that our method handles such a large FOV makes it a candidate for visualization in different settings, such as those enumerated in the next Section.

7 Applications

The method described above does not depend on the content on the image, nor require human interaction to specify which regions of the image should be preserved (it is a global projection, not a local warping technique). The processing time needed is thus much smaller, making it suitable to be applied on panoramic videos and interactive panorama visualizers, such as Google Street View [9].

In interactive applications, apart from the resulting image, the response speed of the interface is very important. One improvement that can be incorporated to our technique is to have various input spheres (in practice, various equirectangular images), with different resolutions. This would allow to choose one input sphere, based on the current FOV, preventing the projection to process unnecessarily large amounts of points. The values of ϕ for which the sphere must be switched must be estimated experimentally.

Note that actually something similar to the latter approach is implemented in Google Street View where, from one input sphere, multiple projections with different resolutions are produced [2]; the viewing sphere is never downloaded (this also minimizes the amount of data transmitted, which is another constraint to keep in mind when working on a client/server environment).

Our method can be coupled with common user interfaces of panorama visualizers. For instance, movements with the mouse when holding a button pressed can be used to pan the image (this is, to modify α and λ) and the mouse wheel can be used to increase and decrease the FOV (that is, varying at the same time ϕ_α and ϕ_λ). These interactions can be alternatively done with keys (for instance, panning with cursor arrows while changing the FOV with page up and page down keys).

Since the method is not computationally expensive, considering current technological developments, it can be implemented by processing transformations on a GPU, using a shader. For instance, the image can be a texture to be applied on the surface of the viewing sphere. In this setting, when panning, neither the sphere nor the texture are altered. When zooming, the texture has to be updated in order to reflect the new Möbius transformation applied. We showed that applying Möbius transformations in the GPU, using GLSL, is viable and yields high frame rates (see Table 1).

It is important to stress that, in case of working through a client/server environment, the processing can be easily implemented in the client (with local GPU computations as described in the previous paragraph), in the server or even with a hybrid approach.

To conclude the present Section, let us mention projection on a dome, another important application of our method. This technique consists in projecting an image into the interior surface of a sphere, or a portion of a sphere. Historically, domes where used in planetariums: they showed the stars exactly as we see them. In the last decades, researchers realized that domes could produce very pleasant visual experience not only when visualizing stars; a new research direction in Computer Graphics born. A dome can provide an immersive environment for movies, display of data or games. While specialized hardware tends to be expensive, there are modern low-cost approaches which make domes very appealing for non-professional uses [3, 4]. Since the projection surface on a dome is not plane, plane projection techniques, which evolved since man started to draw maps, are useless. Dome projection methods must consider a sphere, or a portion of it, as input, and must project on the surface of the sphere. Points on the input sphere must be mapped onto the output sphere. In this setting, a Möbius transformation shall be used to directly shrink (or, in this case, also expand) the sphere to displace points to the visible part of the sphere, without using the last perspective projection step, obtaining thus a plane-projection-less dome visualization.

8 Conclusions, Discussion and Future Work

We introduced a technique based on Möbius transformations of the points on the viewing sphere, aimed at improving the perceptual quality of panoramas. We validated the new approach by implementing the method, to demonstrate both its quality and the speed at which it can run.

The methods by Zorin and Barr [25], by Carroll *et al.* [5] or by Wei *et al.* [23] consist in the minimization of energies, which can be used to quantify the perceptual quality of the image but, of course, they need human interaction. Milnor's method was presented more than 40 years ago, and it was conceived to help in a slightly different problem than ours (map projections). This fact motivates us to propose a future research direction: the development of theoretical methods aimed at quantifying the quality of projections without user interaction.

Throughout the paper, we use a constant value to estimate a good value for $\phi_{\max}$, under the assumption that this value is good on the perceptual side. Another interesting research direction would be to study the content of the images, in order to determine the value of $\phi_{\max}$ in function of the topology of the objects which are present in the image. This research may also provide another approach to measure automatically the perceptual quality of projections. Nevertheless, our method can lose the property of being executed in real-time. A related possible research direction would be to study the content of different frames in a 360° video, in order to dynamically change the projection on each frame. This research could also include comparisons with state-of-the-art methods for video correction [23].

As mentioned in Section 7, our method can be used to visualize 360° images on a dome. This has very important implications that should be discussed here. It follows that the well-studied methods that project the viewing sphere onto a plane or a cylinder are not useful on this setting. Thus, our method can be viewed as a member of another family of transformations on the sphere. We also mention that our method can be (and actually was) implemented in real-time, what permits to apply it on 360° videos.

A topic related to dome projection, out of the scope of the paper, which can be considered as an application of our method, is zooming images. The idea is simple when working on plane images or videos: zooming consists in narrowing or widening the FOV. However, this is no longer true in 360° images or videos, because the FOV cannot be adjusted. A zoom on a sphere consists in a transformation, shrinking some region but, naturally, stretching another region. This is exactly what the Möbius transformations presented in this paper do. Thus, the transformation of the viewing sphere we proposed throughout the paper is a zooming technique for 360° images.

Finally, let us mention that Möbius transformations are a simple and well-known mathematical tool which was not deeply explored in Computer Graphics and Image Processing. As mentioned earlier, we believe their applications can be extended beyond panorama visualization, namely, where an image is represented on a viewing sphere. This is an interesting direction to explore in the future. To uncover applications of Möbius transformations in visualization, it would be interesting to make a deeper study of the foundations of this method. Exploring notions such as the cross-ratio [14] can shed light to a myriad of new techniques for the transformation of images inscribed on the viewing sphere, such as changing the perspective of scenes with Möbius transformations.

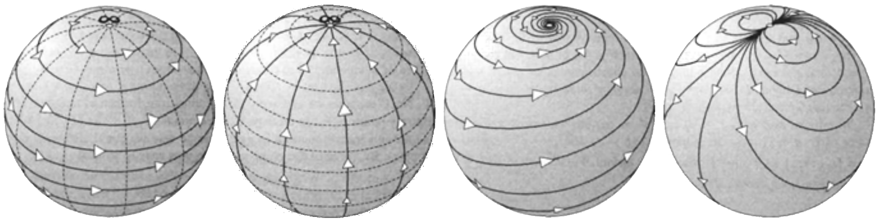
It also remains open the question about the perceptual quality of projections: is it possible to conceive an analytic method to quantify the quality in a way that is consistent with human perception?

Acknowledgements The authors thank the Flickr and Wikimedia Commons users who made available their equirectangular images under the Creative Commons license, used to obtain some figures of the paper: Gadl (Figures 6, 10, 11 and 12), Luca Biada (Figure 7)DXR (Figure 8) and HamburgerJung (Figure 9).

References

1. Cambridge in colour: Cameras vs. the human eye. `http://www.cambridgeincolour.com/tutorials/cameras-vs-human-eye.htm`. Accessed: 24/09/2013.
2. Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43, 2010.
3. Paul David Bourke. Spherical mirror: a new approach to hemispherical dome projection. In *3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, pages 281–284. ACM, 2005.
4. Paul David Bourke and Dalai Quintanilha Felinto. Blender and immersive gaming in a hemispherical dome. In *Computer Games, Multimedia and Allied Technology Conference*, 2010.
5. Robert Carroll, Maneesh Agrawal, and Aseem Agarwala. Optimizing content-preserving projections for wide-angle images. *ACM Trans. Graph.*, 28(3):43:1–43:9, July 2009.
6. Pablo D'Angelo et al. Hugin – panorama photo stitcher, version 2014.0.0. `http://hugin.sourceforge.net`.
7. Helmut Dersch et al. Panorama tools, version 2.9.19. `http://panotools.sourceforge.net`.
8. Daniel M. Germán, Lloyd Burchill, Alexandre Duret-Lutz, Sébastien Pérez-Duarte, Emmanuel Pérez-Duarte, and Josh Sommers. Flattening the viewable sphere. In *3rd Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics '07, pages 23–28. Eurographics, 2007.
9. Google street view. `http://www.google.com/streetview`.
10. R. Kingslake. *A History of the Photographic Lens.* Academic Press, 1989.
11. Johannes Kopf, Dani Lischinski, Oliver Deussen, Daniel Cohen-Or, and Michael Cohen. Locally adapted projections to reduce panorama distortions. *Computer Graphics Forum*, 28(4):1083–1089, 2009.
12. Johannes Kopf, Matt Uyttendaele, Oliver Deussen, and Michael F. Cohen. Capturing and viewing gigapixel images. *ACM Trans. Graph.*, 26(3), July 2007.
13. John Milnor. A problem in cartography. *The American Mathematical Monthly*, 76(10):1101–1112, December 1969.
14. Tristan Needham. *Visual complex analysis.* Clarendon Press, 1997.

15. Jonas Pfeil, Kristian Hildebrand, Carsten Gremzow, Bernd Bickel, and Marc Alexa. Throwable panoramic ball camera. In *SIGGRAPH Asia 2011 Emerging Technologies*, SA '11, pages 4:1–4:1. ACM, 2011.
16. Leonardo Sacht. Content-based projections for panoramic images and videos. Master's thesis, IMPA, April 2010.
17. Leonardo Sacht and Luiz Velho. Complex plane transformations for manipulation and visualization of panoramas. In *International Conference on Computer Graphics Theory and Applications (GRAPP 2013)*, 2013.
18. Thomas K. Sharpless, Bruno Postle, and Daniel M. Germán. Pannini: a new projection for rendering wide angle perspective images. In *6th international conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics'10, pages 9–16, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.
19. Tom Sharpless. Panini perspective tool, version 0.71.104. `http://pvqt.sourceforge.net`.
20. John Parr Snyder. *Map Projections–A Working Manual*, volume 1395 of *Geological Survey Bulletin Series*. U.S. G.P.O., 1987.
21. William A. Stein et al. *Sage Mathematics Software (Version 5.7)*. The Sage Development Team, 2013. `http://www.sagemath.org`.
22. Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufman, 2nd edition, 2004.
23. Jin Wei, Chen-Feng Li, Shi-Min Hu, Ralph R. Martin, and Chiew-Lan Tai. Fisheye video correction. *Visualization and Computer Graphics, IEEE Transactions on*, 18(10):1771–1783, Oct 2012.
24. Lihi Zelnik-Manor, Gabriele Peters, and Pietro Perona. Squaring the circles in panoramas. In *10th IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, pages 1292–1299, Washington, DC, USA, 2005. IEEE.
25. Denis Zorin and Alan H. Barr. Correction of geometric perceptual distortions in pictures. In *22nd annual conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 257–264, New York, NY, USA, 1995. ACM.



(a) elliptic (b) hyperbolic (c) loxodromic (d) parabolic

Fig. 13: Four kinds of Möbius transformations (drawings taken from [14]).

A Möbius Transformations

We recall in this appendix some properties of Möbius transformations. The information here should be enough to understand the paper. We refer to the book by Needham [14] for proofs and details.

A Möbius transformation is a mapping of the form

$$M(z) = \frac{az+b}{cz+d},$$

where a, b, c and d are complex constants. Multiplying these coefficients by a constant yields the same mapping, thus what matter are the ratios of the coefficients (usually the coefficients are normalized by scaling them to satisfy $ad-bc=1$). This fact shows that only three complex numbers are sufficient to determine uniquely a mapping. It can be shown that there exist a unique Möbius transformation which sends any three points to any other three points.

Möbius transformations have fixed points, computed through the equation $z = M(z)$. Since this equation is quadratic, it has at most two solutions (except for the identity mapping). When $c \neq 0$, both fixed points lie in the finite plane. When $c = 0$, at least one of the fixed points lies at infinity (as it can be deduced from from Figure 13, infinity is represented by the north pole N of the Riemann sphere). In this case, the transformation is simplified, taking the form $M(z) = Az+B$. We can write $A = \rho e^{i\alpha}$ in order to view $M(z)$ as the composition of a rotation of α centered in the origin, plus an expansion by ρ and a translation of B. This interpretation lets us view geometrically the Möbius transformation, as depicted in Figure 13.

When $\alpha > 0$, $\rho = 1$ and $B = 0$, $M(z)$ is a rotation of the complex plane (*i.e.*, a rotation of the sphere), as shown if Figure 13a. The fixed points of this transformation are the two poles of the sphere, which correspond, in the complex plane, to the origin and infinity. This is called an *elliptic* Möbius transformation.

Figure 13b illustrates the transformation when $\alpha = 0$, $\rho > 1$ and $B = 0$. $M(z)$ is, in this case, an expansion centered in the origin. The two fixed points are the same as in the previous case. If $\alpha = 0$, $\rho < 1$ and $B = 0$, it is an origin-centered contraction. These Möbius transformations are called *hyperbolic.*

When $\alpha \neq 0$, $\rho \neq 1$ and $B = 0$, the resulting transformation is a combination of the two former cases. This is called a *loxodromic* Möbius transform and it is illustrated in Figure 13c. The two fixed points of this transform are the same two as the above cases.

The last case study is the translation, occurring when $A = 0$ and $B \neq 0$. The only fixed point of this transformation is infinity, represented on the sphere by the north pole. This transformation, the *parabolic* Möbius transformation, is depicted in Figure 13d.