Laboratório VISGRAF
Instituto de Matemática Pura e Aplicada

Improving Projections of Panoramic Images with Moebius Transformations

Luis Peaaranda and Leonardo Sacht and Luiz Velho

Technical Report TR-13-03 Relatório Técnico

August - 2013 - Agosto

Improving Projections of Panoramic Images with Möbius Transformations

Luis Peñaranda, Leonardo Sacht and Luiz Velho

Abstract

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

In this work, we present a technique aimed to improve the visual quality of the panoramas. The main ingredients of our approach are, on one hand, considering the viewing sphere as a Riemann sphere, what makes it natural to apply 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.

Moreover, we present a method to quantify the distortion in projections of a sphere and argue, based on this, about the quality of the images produced by our approach. We also introduce an implementation of our approach, showing that the transformations can be done in real-time, which makes our method very appealing for existing interactive panorama applications.

Index Terms

computer graphics, panoramas, Möbius transformation, conformal projections, distortion

1 Introduction

An image is called panoramic when it represents a wide field of view (FOV, in photography, the part of the world that 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 obtained with special cameras and software.

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 [15], different projection schemes were proposed. Warping techniques were also developed [2], [11]; however, they need long human interaction or optimization methods and thus cannot be used for interactive applications.

One kind of transformation might be good for some setting, while bad for other. This situation becomes evident on some interfaces where a user is able to interactively change the FOV of a

- *L. Peñaranda is with IMPA–Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil.*
  E-mail: luisp@impa.br
- *L. Sacht is with IMPA–Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil, and ETH Zürich, Switzerland.*
  E-mail: leo-ks@impa.br
- *L. Velho is with IMPA–Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil.*
  E-mail: lvelho@impa.br

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 turn out to be less realist.

What we propose in this paper is to adopt a different approach for obtaining a panoramic image when the FOV becomes wide in interactive applications. 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 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. Section 4 presents the current approach to visualize panoramas, the perspective projection, while our approach is formalized in Section 5. Section 6 introduces some scenarios where our technique can be applied. Finally, Sections 7 and 8 provide an analysis of the algorithm in theory and in practice, while Section 9 discusses work to do and future 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 [15]. This theoretical limitation motivated much research for obtaining visually realistic panoramas.

One approach for this problem was to use different perspective projections in the same scene [14]. 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.

Conformal mappings were also explored with the aim of preserving the shape of the objects in a panoramic image [5]. 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 allows for better quality results because the straight lines are less bent.

Other methods relied on both user interaction and energy-minimization formulations. Carroll et al. [2] 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. [7] used regions specified by the user where the projection should be nearly planar to formulate their optimization framework. In these approaches, user interaction was usually laborious and the optimization formulations made them impossible to be implemented in real-time.

An interesting approach consisted in dynamically changing the projection used depending on the FOV [8]. 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 5. But, in our work, we let the user specify more parameters, what gives her or him more control. 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 two parameters, 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 our method possible to be implemented in real-time.

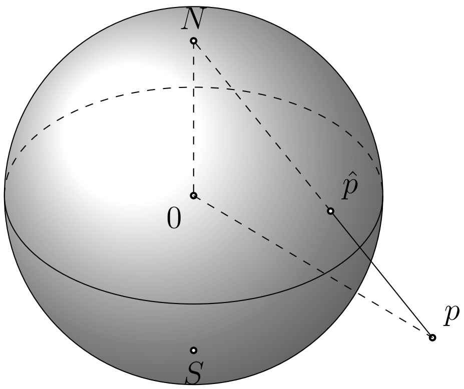


Fig. 1: Correspondence between a point $\hat{p}$ on the Riemann sphere and a point p on the complex plane.

A work that is not related to panoramic images but has some connection with ours is [3]. In this work, the authors look for conformal transformations of surfaces by associating $\mathbb{R}^3$ with the imaginary part of the quaternions $\Im(\mathbb{H})$. We work with the complex plane $\mathbb{C}$ instead of $\Im(\mathbb{H})$, since our surface of interest is the unit sphere $\mathbb{S}^2$ and $\mathbb{S}^2 \setminus \{(0,0,1)\}$ is confomally equivalent to the complex plane. Applying their work to our context would produce conformal results, but straight lines in the scene would appear bent.

3 Definitions

This section formalizes 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 a 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. Stereographic projections will not be studied here; we only show the following formulas for completeness[1].

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

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

We assume the observer standing 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.

[1] Note that, in case of using a projection other than stereographic, the results in the sequel still hold. We opted for the stereographic projection in this work because it is *conformal*. The composition of it with a Möbius transformation and the inverse of the stereographic projection (these both transformations are also conformal) will result in a conformal transformation of the input sphere, because compositions of conformal transformations also have this property. We refer to previous works for the importance of conformality for the generation of panoramic images.

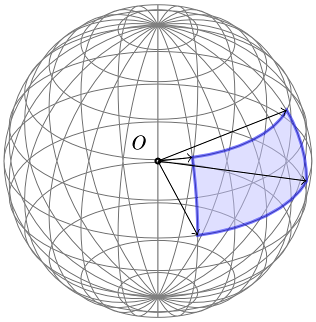
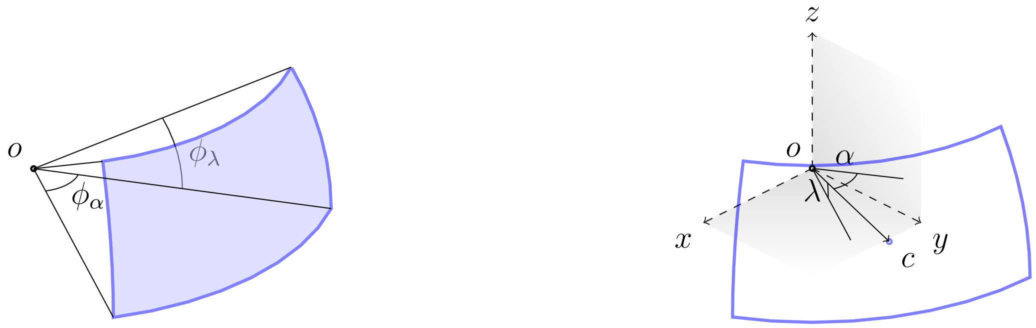


Fig. 2: The viewing sphere. The FOV (blue region) is to be mapped to a rectangular image. o is the observers position, in the center of the sphere.



(a) Modelling FOV: ϕ_α and ϕ_λ.

(b) $\vec{oc}$ is the eyesight direction. α is the angle between vector $\vec{oc}$ and plane yz. λ is the angle between vector $\vec{oc}$ and plane xy.

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

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 3b.

The FOV is again represented as two angles, ϕ_α (azimuthal FOV) and ϕ_λ (altitudinal FOV), as depicted in Figure 3a. 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. Next sections will formalize this and present our method. It should be stressed at this point that, in the rest of the paper, angles will be given in radians unless explicitly remarked.

4 The Perspective Projection

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. Thus, a value of ϕ_α uniquely determines the four points on the sphere which are the vertices of the blue region. For a given ϕ_α, the FOV is computed, and points on the sphere are mapped to the output image by a perspective projection $\mathbf{P}_{\phi_\alpha}$ (which depends on the FOV).

$$\mathbf{P}_{\phi_\alpha} : [-\phi_\alpha, \phi_\alpha] \times [-\phi_\lambda, \phi_\lambda] \rightarrow [0, r_x - 1] \times [0, r_y - 1] \tag{1}$$

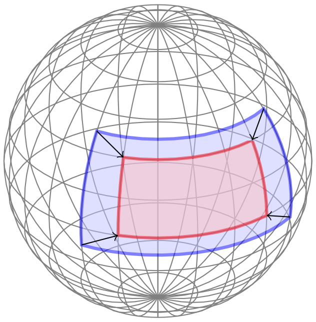


Fig. 4: The Möbius transformation, shrinking points in the new FOV (in blue) to the old FOV (in red).

See Section 7 for explicit formulas for the perspective projection.

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 $\mathbf{P}_{\phi_\alpha}$ must be composed with a rotation function on the sphere.

5 Visualizing 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 $\mathbf{P}_{\phi_\alpha}$. One approach is to replace $\mathbf{P}_{\phi_\alpha}$ by a warping which respects visual restrictions, such as conformality and preservation of straight lines, as much as possible. We refer to [11] for details on such methods. The drawback of these methods is that they are 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 in function of the current FOV is [8]. They use an adaptive projection resulting from an interpolation between a perspective and a cylindrical projection.

We assume that, when $\phi_\alpha \leq \phi_{\text{max}}$, the perspective projection does not introduce big distortions, thus we propose to use the projection method described in Section 4.

On the other hand, when $\phi_\alpha > \phi_{\text{max}}$, the perspective projection fails to provide a realistic image. We propose, in this case, to perform a Möbius transformation [10] on the sphere points, in order to map points of the new FOV region to the old FOV region, as depicted in Figure 4. After, the same projection $\mathbf{P}_{\phi_{\text{max}}}$ maps points on the sphere to the output image.

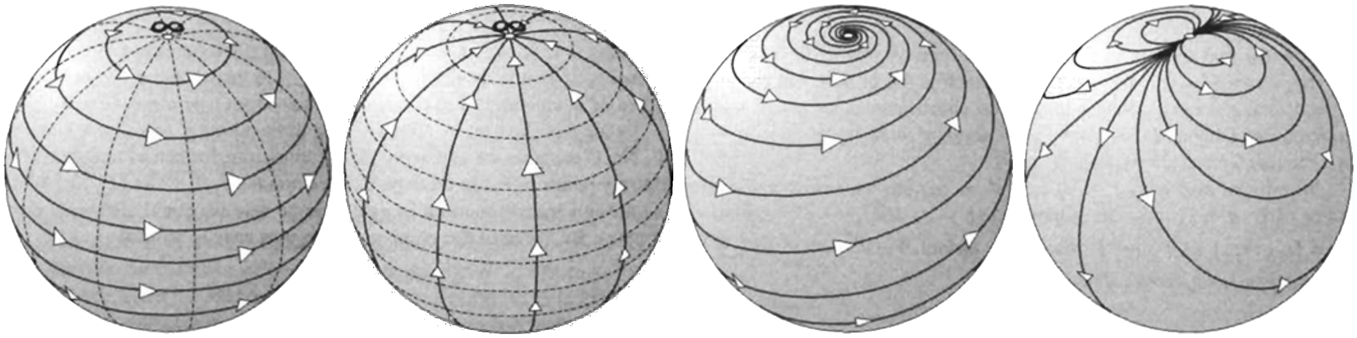
5.1 Möbius Transformations

Before going further, we will recall some properties of Möbius transformations. We refer to [10] 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.



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

Fig. 5: Four kinds of Möbius transformations (drawings taken from [10]).

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 4, 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 5.

When $\alpha > 0$, $\rho = 1$ and $B = 0$, $M(z)$ is a rotation of the complex plane, which traduces in a rotation of the sphere, as shown if Figure 5a. 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 5b 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 5c. 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 5d.

5.2 Computing the Shrink

The problem is to find the Möbius transformation that shrinks the new FOV into the old FOV. If we had the point c of Figure 3b 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 to the origin on the complex plane). Mathematically, this means to do a translation by a vector $\mathbf{t}(-\alpha, -\lambda)$ in polar coordinates (note that this translation must not be done in the equirectangular image). We assume, for simplicity in the rest of the section, that $\alpha = \lambda = 0$.

The following step is to perform a shrink centered in the origin. Of the four cases studied in Section 5.1, it is clear that it must be employed in our problem an hyperbolic Möbius transformation. The shrink transformation takes the form $M_s = \rho z$, and the only remaining problem is to compute ρ. Since ϕ_{max} and ϕ are angles, it is direct to compute, in the equirectangular domain, $\rho = \frac{\phi_{\text{max}}}{\phi}$. Applying M_s, we map the old FOV to the new FOV, as wanted.

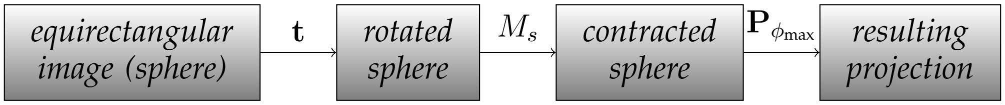


Fig. 6: Pipeline of the shrinking process.

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 6.

The shrink is a good approach to reach big FOVs of panoramas. However, even with the shrink/projection scheme, a FOV closer 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 $\phi_{\max}$. These shall be determined by experimentation.

6 Applications

The method described in Section 5 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 projection, not a 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 [6].

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 [1]; 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.

It is important to stress that, in case of working through a client/server environment, the processing can be easily implemented in the server, 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 another important application of our technique. When projecting on a dome, where typically the viewers always see a half-sphere, plane projections must be avoided. Points are on a sphere and they are projected on a sphere. A Möbius transformation shall be used to directly shrink (or, in this case, also expand) the sphere to displace points to the half-sphere, obtaining thus a projection-less dome visualization.

7 Visual Analysis

In this section, we will argue about the advantages of using the Möbius shrinking technique for wide FOVs over the common direct perspective projection.

It is widely known that perspective projections introduce a very small distortion when the FOV is small (it is typically considered that perspective projection produces acceptable results for a FOV of $\frac{\pi}{3}$ or less). Distortions can even be not visually noticeable in some cases. On the other hand, perspective projections of a wide FOV introduce gross distortions. As there is no silver bullet for this situation [15], many different projection techniques were developed (in fact, this is a very old research topic: the problem was originally formulated as obtaining a good projection for map drawing, see [12] for a survey on map projection techniques). The first argument in favor of our technique is thus that it avoids wide-angle projections.

It would also be interesting to perform a theoretical analysis of the distortion introduced by our method and to compare it to the distortion produced by the perspective projection. The first step for this would be to express the formulas of the perspective projection and of the Möbius shrinking technique. For these, we will fix a FOV ϕ_M (for simplicity, we will use the same FOV horizontally and vertically) and, for the second case, a value of $\phi_{\max}$. Moreover, we will assume that projections are done on a square region of the Euclidean plane.

The perspective projection in the square region $[-r, r] \times [-r, r]$ of the Euclidean plane of a point $p(\lambda, \phi)$ (given in angular coordinates) on a sphere is

$$P_{\phi_M}(\lambda, \phi) = \left(\frac{r}{\tan(\phi_M)} \tan(\lambda), \frac{r\cos(\phi_M)}{\tan(\phi_M)} \frac{\tan(\phi)}{\cos(\lambda)} \right). \tag{2}$$

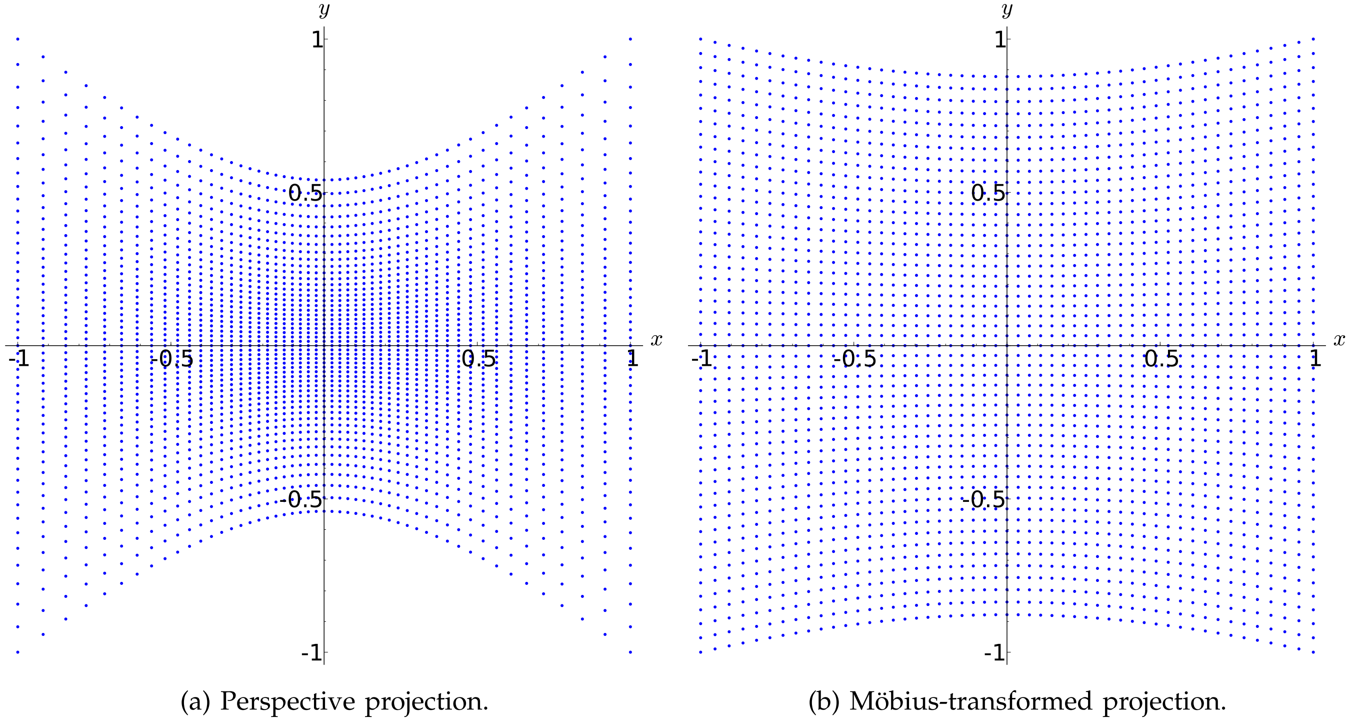
Viewed as a projection, the Möbius technique described above is obtained as the following formula:

$$Q_{\phi_M}(\lambda, \phi) = \left(\frac{r}{\tan(\phi_{\max})} \tan\left(\frac{\phi_{\max}}{\phi_M} \lambda \right), \frac{r\cos(\phi_{\max})}{\tan(\phi_{\max})} \frac{\tan\left(\frac{\phi_{\max}}{\phi_M}\phi\right)}{\cos\left(\frac{\phi_{\max}}{\phi_M}\lambda\right)} \right). \tag{3}$$

To compare the behaviour of the two approaches, we implemented Equations 2 and 3 in Sage [13]. Then, we gave as input to these functions points on a regular grid (a regular grid on the sphere is given by angular coordinates), plotting on the Euclidean plane their images under these functions. Figure 7a shows the known result of applying a perspective projection to the points on the viewing sphere: close to the corners, the input image is unacceptably bent. Figure 7b depicts how the applied Möbius transformation tries to avoid distortions: it can be viewed as a midpoint between a perspective projection and an equirectangular image.

To explain the behaviour shown in Figure 7, we can draw the following visual foundation. To obtain a perspective projection of a set of points on the sphere, we shoot rays from the origin to these points; the projection is obtained by the intersections of these rays with a plane lying behind the sphere. Then, the resulting image on the plane is scaled in order to fit in a given rectangle. In this setting, as the FOV increases, distances close to the borders are greatly distorted. On the other hand, the shrink performed on the first step of the Möbius scheme does not create important distortions in the picture on the surface of the sphere. The perspective projection applied afterwards only introduces a distortion corresponding to a perspective projection for a small FOV. Thus, as the value of $\phi_{\max}$ approaches zero, the result of the Möbius scheme tends to be an equirectangular image.

To quantify the distortion, we can use a method by Milnor [9], which we describe next. Assume we are working with a set of points U on the sphere (for us, U is a region defined by the FOV). Let $d_S(x, y)$ the geodesic distance on the sphere of two points x and y, and $d_E(a, b)$ the Euclidean distance of two points a and b on the Euclidean plane E. Then, the scale of a projection f with



(a) Perspective projection.

(b) Möbius-transformed projection.

Fig. 7: The two projection schemes. $r = 1$, $\phi_M = 1$ and $\phi_{\max} = 0.5$.

respect to a pair of distinct points x and y is defined by the ratio

$$\sigma = \frac{d_E\big(f(x), f(y)\big)}{d_S(x, y)}.$$

In general, the scale σ varies for each pair of points of U. Thus, we define the minimum and maximum scale, σ_1 and σ_2, defined as the infimum and the supremum of σ, respectively. Finally, we define the *distortion* of the projection as

$$\delta = \log\left(\frac{\sigma_2}{\sigma_1}\right). \tag{4}$$

To use these formulas, it would be first necessary to determine the functions d_S, d_{E_P} (the distance under perspective projection) and d_{E_Q} (the distance under the Möbius technique).

Consider now, for the computations of the distances, two points on the sphere of radius ρ given by spherical coordinates, $p_1(\lambda_1, \phi_1)$ and $p_2(\lambda_2, \phi_2)$. The geodesic distance between p_1 and p_2 is

$$d_S(p_1, p_2) = \rho \arccos\big(\cos(\lambda_2 - \lambda_1) + \cos(\phi_2 - \phi_1) - 1\big). \tag{5}$$

The computation of the Euclidean distances requires the formulas of the perspective projection and of the Möbius shrinking technique. For simplicity, we shall write Equation 2 as $P_{\phi_M}(\lambda, \phi) = \big(P_x(\lambda, \phi), P_y(\lambda, \phi)\big)$. Then, the Euclidean distance between the images under perspective projection of p_1 and p_2 is

$$d_{E_P}(p_1, p_2) = \sqrt{\big(P_x(\lambda_2, \phi_2) - P_x(\lambda_1, \phi_1)\big)^2 + \big(P_y(\lambda_2, \phi_2) - P_y(\lambda_1, \phi_1)\big)^2}, \tag{6}$$

and, calling by convenience $Q_{\phi_M}(\lambda, \phi) = \big(Q_x(\lambda, \phi), Q_y(\lambda, \phi)\big)$ in Equation 3, the Euclidean distance between the images under Q_{ϕ_M} of p_1 and p_2 is

$$d_{E_Q}(p_1, p_2) = \sqrt{\big(Q_x(\lambda_2, \phi_2) - Q_x(\lambda_1, \phi_1)\big)^2 + \big(Q_y(\lambda_2, \phi_2) - Q_y(\lambda_1, \phi_1)\big)^2}. \tag{7}$$

TABLE 1: Frame rates obtained by our technique while the user interacts, for different mesh resolutions (varying on the rows) and different input equi-rectangular 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

Equations 5, 6 and 7 suffice to compute σ_P and σ_M, the scales of the perspective projection and Möbius shrinking, respectively. And, with them, δ_P and δ_M, their respective distortions. As we can observe in Equation 4, what matters to compute distortion is the ratio between the maximum and the minimum scales.

Figure 7 helps to understand the behaviour of the two projection schemes in the context of the introduced theoretical distortion. Let us consider first distortions along a tropic (that is, for a fixed λ). We observe that, in perspective projection, distances between consecutive points on a line along a tropic are varying considerably. On the other hand, these distances vary much less in the Möbius-transformed projection. It is worth mentioning that the geodesic distance between consecutive points in the sphere is constant in this case, what means that the values of σ_M vary less than the values of σ_P.

If we consider distances between consecutive points along a meridian (that is, points with a fixed α), we observe exactly the same phenomenon. Since distances of points in the plane are related by the Pythagorean theorem, we infer that distances in the perspective projection vary more than distances in the Möbius-transformed projection. This is equivalent to stating that the ratio between the maximum and minimum values of σ_M is closer to one than the ratio between the maximum and minimum values of σ_P. Thus, the logarithm of the first is closer to zero than the logarithm of the second, that is, $\delta_M < \delta_P$.

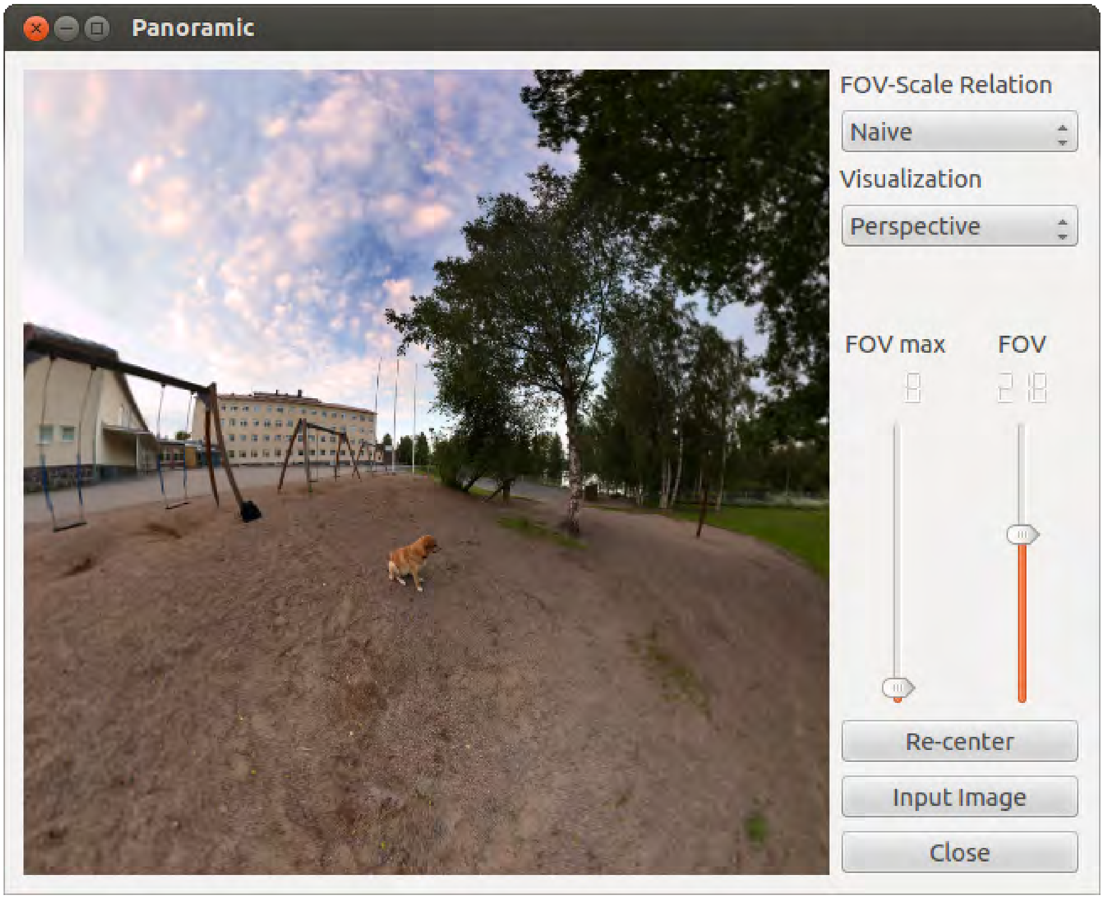
In the next Section, we will compare panoramas produced with our method and with the perspective projection.

8 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 visual 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 code consists in a window showing the image processed through our technique. The interface, shown in Figure 8, permits to pan the image with the mouse, as well as adjusting the FOV and the parameter ϕ_{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. The code is available upon request. All the images shown in this section where taken from Flickr [4].

We reproduce here some results of our tests. Figure 9 shows how our method improves the quality of the images near the borders, while keeping the image on the center undistorted. This behaviour was already shown in Figure 7, and was expected from the theoretic analysis.

Our implementation can also serve to find values of under different settings. For very wide FOVs, a small value of ϕ_{max} is needed. For smaller values of ϕ, the perspective projection suffices: in numbers, $\phi_{\text{max}} = \phi$ for $\phi < 60^\circ \approx 1$rad. For ϕ between 1rad and 2πrad (in degrees, $60^\circ \leq \phi \leq$



Panoramic

FOV-Scale Relation

Naive

Visualization

Perspective

FOV max

8

FOV

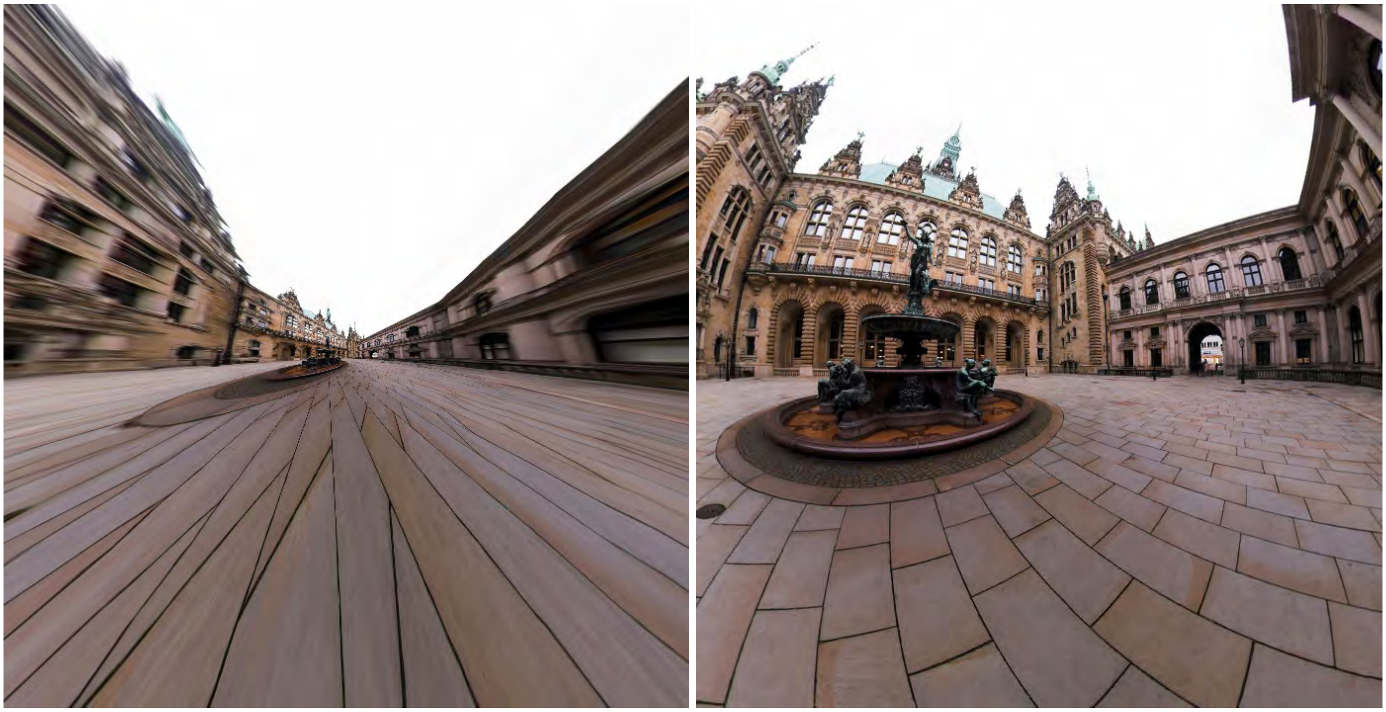
218

Fig. 8: The window of our implementation showing a panorama with a wide FOV (the values in degrees of ϕ and $\phi_{\max}$ are shown under the labels of the sliders). In the picture, it is possible to appreciate how realistic are visualizations with our method.

$360°$), intermediate values should be used. A function to determine the value of $\phi_{\max}$ in function of ϕ would be of great interest.

We also compare our technique to previous methods in the field of generation of panoramic images. We start by showing in Figure 10 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 10b) distorts objects too much but preserves all straight lines. On the other hand, Mercator (Figure 10c) projection preserves object shapes because it is conformal, but bends lines. The result obtained with our method (Figure 10d) using $\phi_{\max} = 120°$ is the only one that presents a good balance between these two properties.

In Figure 11 we compare our method with recent works on the same topic. The result obtained by the technique proposed in [14] (Figure 11a) 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 discontinuities in the scene, but it fails to fit the geometry of the floor. In Figure 11b we show a result produced by our implementation of the technique in [2]. All straight lines specified by the user (please 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 result (Figure 11c), for which we use $\phi_{\max} = 110°$, does not rely on heavy user interaction nor on any optimization and is not restricted to scenes with any particular geometry.



(a) perspective projection

(b) Möbius, $\phi_{\max} = 0.02\text{rad}$

Fig. 9: The figure shows how the perspective projection can be improved with our technique for big fields of view. The perspective projection is very distorted in the sides, while the Möbius technique avoids distortion. In the two figures, $\phi = 172° \approx 1.5\text{rad}$.

However, there are still some cases that are hard to deal with. For instance, Figure 12 shows what happens with a very wide FOV, close to $2\pi\text{rad} = 360°$. 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 these FOVs makes it a candidate for visualization in different settings, such as projections on domes.

9 Future Work

Since Möbius transformations are usually viewed as a transformation on a sphere, we believe they can be used not only in panorama visualization, but in many applications where an image is represented on a viewing sphere. This is an interesting direction to explore in the future. To uncover other 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 [10].

It would also help to determine some limits of our technique, such as the maximum FOV that can be realistically visualized. The implementation can be also used to have an insight on a function to determine $\phi_{\max}$ in function of ϕ.

Acknowledgement

The authors thank the Flickr users who made available their images under the Creative Commons license.

References

[1] 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.

[2] 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.

[3] Keenan Crane, Ulrich Pinkall, and Peter Schröder. Spin transformations of discrete surfaces. *ACM Trans. Graph.*, 30(4):104, 2011.

[4] Flickr. http://www.flickr.com.

[5] D. M. German, L. Burchill, A. Duret-Lutz, S. Pérez-Duarte, E. Pérez-Duarte, and J. Sommers. Flattening the viewable sphere. In *Proceedings of the Third Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics'07, pages 23–28, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics.

[6] Google street view. http://www.google.com/streetview.

[7] 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.

[8] Johannes Kopf, Matt Uyttendaele, Oliver Deussen, and Michael F. Cohen. Capturing and viewing gigapixel images. *ACM Trans. Graph.*, 26(3), July 2007.

[9] John Milnor. A problem in cartography. *The American Mathematical Monthly*, 76(10):1101–1112, December 1969.

[10] Tristan Needham. *Visual complex analysis*. Clarendon Press, 1997.

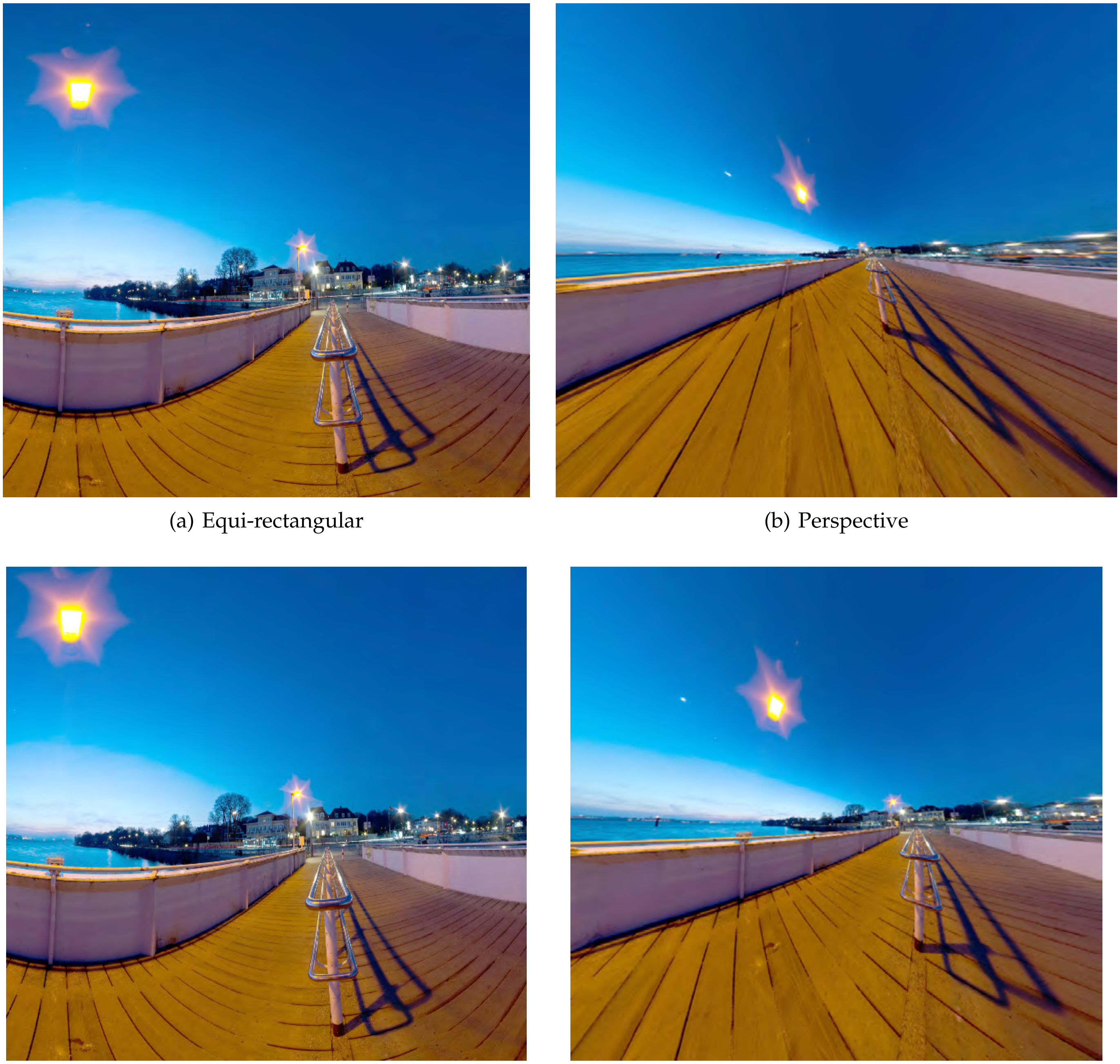
[11] Leonardo Sacht. Content-based projections for panoramic images and videos. Master's thesis, IMPA, April 2010.

[12] John Parr Snyder. *Map Projections–A Working Manual*, volume 1395 of *Geological Survey Bulletin Series*. U.S. G.P.O., 1987.

[13] William A. Stein et al. *Sage Mathematics Software (Version 5.7)*. The Sage Development Team, 2013. http://www.sagemath.org.

[14] Lihi Zelnik-Manor, Gabriele Peters, and Pietro Perona. Squaring the circles in panoramas. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, pages 1292–1299, Washington, DC, USA, 2005. IEEE.

[15] Denis Zorin and Alan H. Barr. Correction of geometric perceptual distortions in pictures. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 257–264, New York, NY, USA, 1995. ACM.



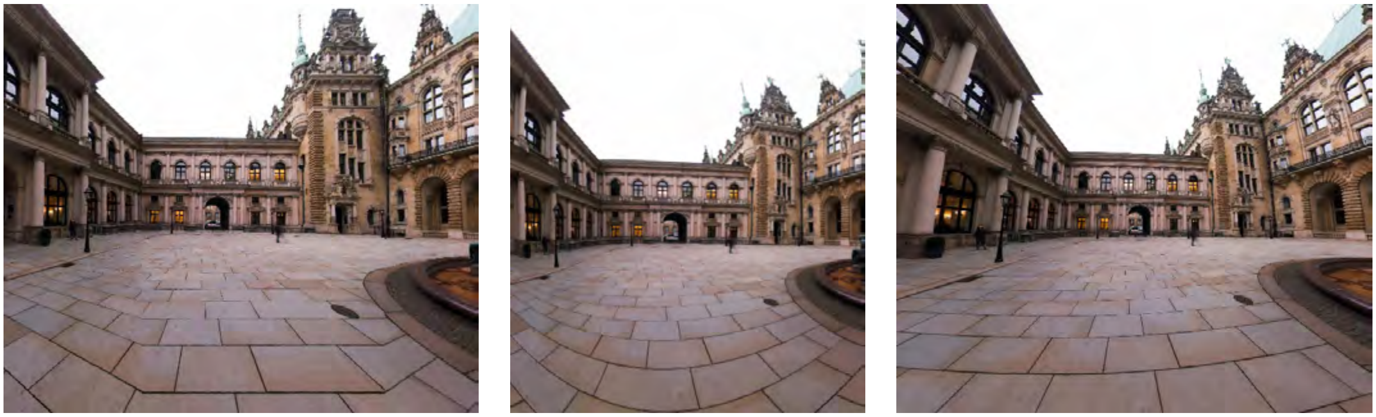
(a) Equi-rectangular

(b) Perspective

(c) Mercator

(d) Ours, $\phi_{\max} = 120°$

Fig. 10: Comparison with standard projections for $\phi = 160°$. Our result is the only one with a good balance between straight line and object shape preservation.



(a) Zelnik-Manor et al. [14] (b) Carroll et al. [2] (c) Ours, $\phi_{\max} = 110°$.

Fig. 11: Comparison with recent methods for $\phi = 150°$. In the result obtained by the method in [14] (a) the different perspective projections used for different areas of the image appear clear and unpleasant on the floor of the scene. The method by Carroll et al. [2] (b) 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). Our result (c) has all straight lines in the scene with very little bending.

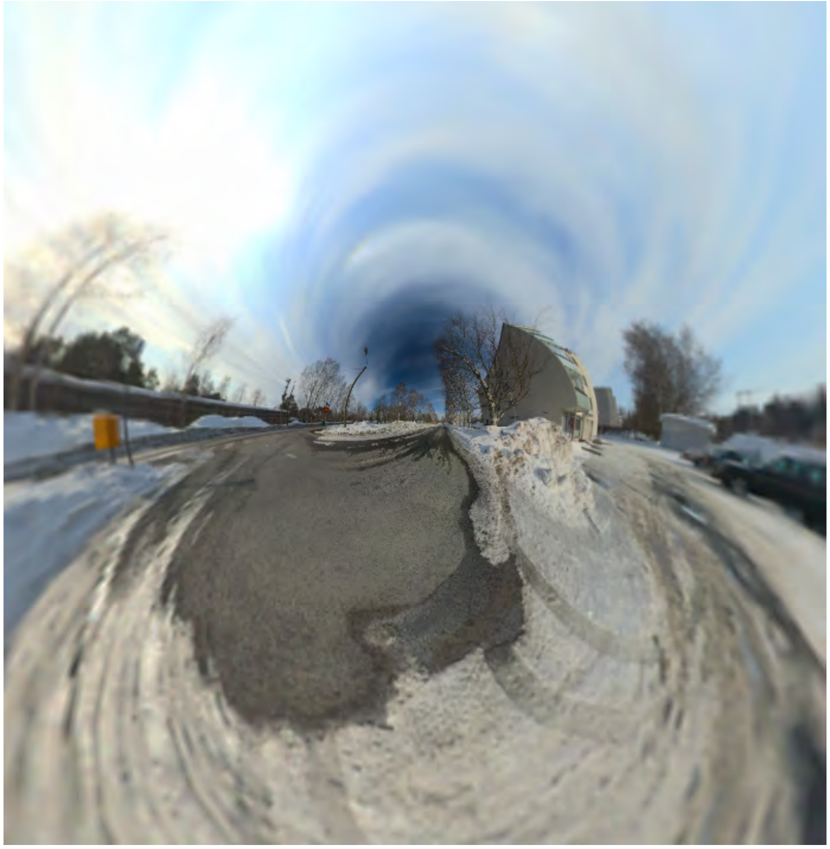


Fig. 12: Even if our method can handle very wide FOVs and displayed objects are distinguishable, the image seems unrealistic. Here, $\phi = 320° \approx 5.6\text{rad}$, $\phi_{\max} = 40° \approx 0.7\text{rad}$.