

Laboratório VISGRAF

Instituto de Matemática Pura e Aplicada

**Moebius Transformations Applied to Omnidirectional
Images**

Leonardo Souto
Leonardo Sacht
Luiz Velho

Technical Report TR-17-02 Relatório Técnico

February - 2017 - Fevereiro

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

Moebius Transformations Applied to Omnidirectional Images

Leonardo Souto
Leonardo Sacht
Luiz Velho

February 2017

Abstract

This work presents a new method to transform omnidirectional images based on a combination of Moebius transformations in the complex plane and Gaussian weights that limit the action of these mappings to regions of interest. Since no optimization or numerical methods are involved, it is possible to reach real-time performance with the method, making it applicable to streaming services. The results show the potential of this technique for areas such as surveillance and art in general.

1 Introduction

An omnidirectional image is an image that contains the visible information of the full spherical field of view around a point. Omnidirectional images became popular on the Internet and have been an intense research topic during the last twenty years. Due to distortions caused by planar parametrizations, they contain undesirable artifacts such as bent lines, shape distortions and oversampling near the poles. This work presents a new method to improve these images, making them more visually pleasing and allowing more technical uses of them.

A multitude of application areas can benefit from the visible information contained in omnidirectional images: they provide a greater immersive experience in virtual reality contexts, making the user able to look around and zoom in specific regions of the scene; in surveillance they may increase the effectiveness of security cameras; in sport broadcasting they are able to show greater areas of the game and of the audience without requiring camera movements; and in artistic applications they offer several new ways of filming and editing footage of prerecorded scenes. For these reasons, working to generate improvements and corrections of omnidirectional images has great potential to improve the contact that humanity possesses with these types of technology and applications.

Our method presents an extension of the work done in [4], where the authors projected the spherical image in the complex plane using the stereographic projection and then applied a particular case of Moebius Transformations to the complex plane. In our case we use general Moebius transformations that have the expression $M(z) = \frac{az+b}{cz+d}$. Based on the cross-ratio relation we enable the user to define the transformation using three points and their images. Additionally, we give the user the possibility of defining which regions of the image will be affected through the use of smooth weights that are higher on the regions of interest, generating a user interface with higher functionality. Another possibility of our method is to compose a sequence of transformations on the same image, allowing an even greater variety of results.

Since we use closed-form simple transformations, our approach allows us to generate real-time results for megapixel images and opens the possibility of adding this method to streaming equipments.

The text is structured as follows. In section 2 we discuss previous works and the literature in the area of wide-angle images. Section 3 formalizes the idea of an omnidirectional image and introduce the longitude/latitude

parametrization. Section 4 focuses on the foundations of Moebius Transformations and how they relate to spherical images. Section 5 focuses on how smoothing functions can provide a way for a Moebius Transformation to act locally. In section 6 we present the results and limitations of our technique. Section 7 briefly explains our implementation. In section 8 we make concluding remarks and discuss future ideas.

2 Related Work

The goal of correcting omnidirectional images has been a theme of intense research during the last twenty years. The most evident distortions present in these images are straight lines in the world appearing bent and objects appearing too stretched. The seminal work [8] showed that it is impossible to find a mapping from the sphere to the image plane that simultaneously preserves both all possible straight lines and objects.

Many mappings from the sphere to the plane were considered for the task of correcting these distortions. For example, the stereographic projection that identifies the sphere with the extended complex plane and the Mercator projection from cartography books are well known to preserve shapes but they bend straight lines. [5] presents an extensive discussion about these projections and shape preservation (also known as conformality). On the other hand, the perspective projection preserves straight lines but stretches objects too much if the field of view is too wide.

The limitations of these global projections and the impossibility of preserving all desired properties led to content-based methods that rectify panoramic images depending on each scene. For example, [7] proposed to use multiple perspective projections to the same image. This method works well as long as the orientation discontinuities introduced by the multiple projection planes are well hidden by the scene.

More sophisticated methods such as [1] and [3] formulate optimization problems that have the goal of satisfying user specified constraints. They produce high-quality results at the cost of solving large optimization problems that make their use prohibitive for real-time applications. On the contrary, we propose a method based on closed formulas that can generate results in real-time.

Our method apply Moebius transformations to panoramic images. This idea has already been explored in [4] and [6]. [4] limits the discussion to contractions and expansions in the complex plane with the goal of zooming in and out panoramic images. [6] includes rotations and apply their method to videos as well. We generalize this approach by allowing more general Moebius transformations, introducing their specification by the definition of the images of 3 points, and adapting them to act locally.

3 Omnidirectional Images

In this work all the scene observed from a fixed viewpoint is modeled as a unitary sphere centered at the viewpoint ($S^2 = \{(x, y, z) \in \mathbb{R}^3 | x^2 + y^2 + z^2 = 1\}$), where every point on the sphere has an associated color, the color that is seen when someone looks through this point. For convenience, it is assumed that the origin of \mathbb{R}^3 is the viewpoint.

This colored sphere is called *visible sphere* or *omnidirectional image*. Notice that the visible sphere fully represents the field of view with 360° longitude by 180° latitude.

A well-known and useful representation of the sphere S^2 is by longitude and latitude coordinates:

$$r : [-\pi, \pi] \times \left[\frac{-\pi}{2}, \frac{\pi}{2} \right] \rightarrow S^2$$

$$(\lambda, \phi) \iff (\cos \lambda \cos \phi, \sin \lambda \cos \phi, \sin \phi)$$

This representation is illustrated in Figure 1.

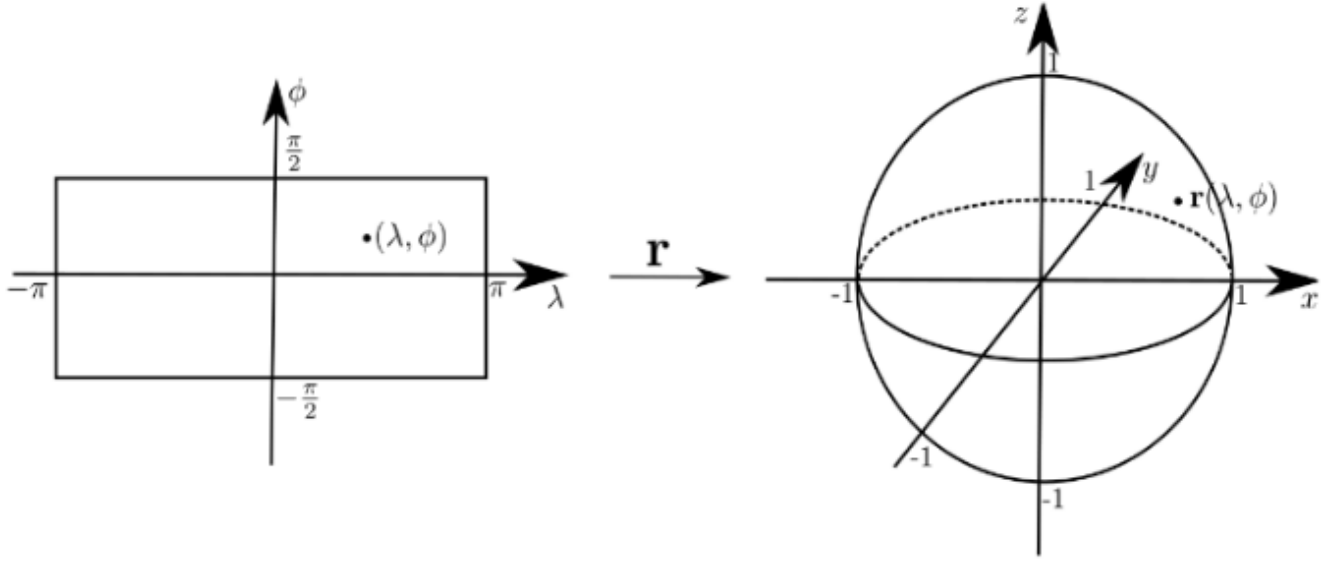


Figure 1: Longitude/latitude representation r

The longitude/latitude domain is *quirectangular domain*. The images in the equirectangular domain are called *quirectangular images* and they will be the input of our algorithm. An example of equirectangular image is shown in Figure 2.

4 Moebius Transformations

It is well known that the complex plane can be identified as the unit sphere (in our case, the visible sphere) via the stereographic projection [2]. Figure 3 illustrates this mapping. Identifying the point from where lines of projection emanate from as ∞ , we obtain a continuous mapping $SP : S^2 \rightarrow \mathbb{C}_\infty = \mathbb{C} \cup \infty$ that has a continuous inverse mapping $SP^{-1} : \mathbb{C}_\infty \rightarrow S^2$. An important property of the stereographic projection is its conformality (formal definition and details in [5]), i.e., it preserves angles and maps circles to circles or lines. This way of interpreting the extended complex plane \mathbb{C}_∞ as a sphere was formulated by Riemann and receives the name of *Riemann Sphere*. In our work, we will consider the visible sphere as being a Riemann sphere with ∞ being the opposite point to the projection plane.

Definition 1. A mapping $S : \mathbb{C}_\infty \rightarrow \mathbb{C}_\infty$ in the form $S(z) = \frac{az+b}{cz+d}$ is called a fractional linear transformation. If a, b, c and d satisfy $ad - bc \neq 0$ then S is called a Moebius Transformation.

If S is a Moebius Transformation, so $S^{-1}(z) = \frac{dz-b}{-cz+a}$ satisfies $S \circ S^{-1}(z) = S^{-1} \circ S(z) = z$; That is, S^{-1} is the inverse mapping of S . If S and T are both fractional linear transformations, so it follows that $S \circ T$ is too. Hence, the set of Moebius mappings constitute a group under composition. This work will treat only about fractional linear transformations that are Moebius Transformations.

Some properties that follow from these definitions are useful for our work:

Proposition 1. If S is a Moebius Transformation, then S is a composition of translations, scalings and inversions.



Figure 2: Example of equirectangular image.

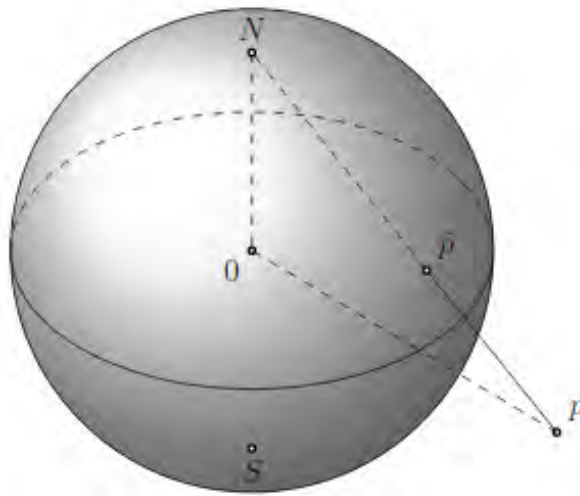


Figure 3: The Relation between a Riemann sphere and the extended complex plane

Proof. First, suppose $c = 0$. Then $S(z) = (a/d)z + b/d$. so if $S_1(z) = (a/d)z$ (scaling), $S_2 = z + b/d$ (translation), then $S_2 \circ S_1 = S$ and the results is proven.

Now, take $c \neq 0$ and make $S_1(z) = z + d/c$ (translation), $S_2(z) = 1/z$ (inversion), $S_3(z) = \frac{(bc-ad)}{c^2}z$ (scaling and rotation) and $S_4(z) = z + a/c$ (another translation). Then $S = S_4 \circ S_3 \circ S_2 \circ S_1$. \square

Proposition 2. *The Moebius Transformations are bijections in \mathbb{C}_∞ .*

Proof. The translation $z \rightarrow z + k, \infty \rightarrow \infty$ is a bijection with inverse $z \rightarrow z - k, \infty \rightarrow \infty$.

If $k \neq 0$, then the scaling $z \rightarrow kz, /, \infty \rightarrow \infty$ is a bijection with inverse $z \rightarrow z/k, \infty \rightarrow \infty$. Notice that the scaling $S_3(z) = \frac{(bc-ad)}{c^2}z$ in the proof of the previous proposition is such that $k = \frac{(bc-ad)}{c^2} \neq 0$, since $ad - bc \neq 0$.

The inversion $z \rightarrow 1/z, 0 \rightarrow \infty, \infty \rightarrow 0$ is a bijection with itself being its inverse.

According to the previous proposition and the fact that compositions of bijections are bijections, we conclude that Moebius transformations are bijections. \square

Definition 2. $z \in \mathbb{C}_\infty$ is a *fixed point* of S if

$$S(z) = z$$

From the definition we have that the fixed points of a Moebius Transformation $M(z)$ are the solutions of

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

The equation can be rewritten in the following form

$$cz^2 + (d - a)z - b = 0$$

So, Moebius transformations possess at most two fixed points or are the identity transformation (the latter happens only if $a = d, b = 0$ and $c = 0$).

Now, let S be a Moebius transformation and take p, q, s distinct points in \mathbb{C}_∞ with $\bar{p} = S(p), \bar{q} = S(q), \bar{s} = S(s)$. Suppose that T is another mapping with this property. Then $T^{-1} \circ S = I$, i.e., $S = T$. Hence, a Moebius mapping is uniquely determined by its action over any three distinct points in \mathbb{C}_∞ .

Let z_1, z_2, z_3 points in \mathbb{C}_∞ . Define $S : \mathbb{C}_\infty \rightarrow \mathbb{C}_\infty$ by

$$S(z) = \left(\frac{z - z_2}{z - z_3} \right) / \left(\frac{z_1 - z_2}{z_1 - z_3} \right) \quad \text{if } z_1, z_2, z_3 \in \mathbb{C};$$

$$S(z) = \frac{z - z_2}{z - z_3} \quad \text{if } z_1 = \infty;$$

$$S(z) = \frac{z_1 - z_3}{z - z_3} \quad \text{if } z_2 = \infty;$$

$$S(z) = \frac{z - z_2}{z_1 - z_2} \quad \text{if } z_3 = \infty.$$

In any case $S(z_1) = 1, S(z_2) = 0, S(z_3) = \infty$ and S is the only transformation with this property.

Definition 3. If $z \in \mathbb{C}_\infty$, then (z, z_1, z_2, z_3) (the *cross ratio* of z, z_1, z_2 and z_3) is the image of z under the unique Moebius Transformation that maps z_1 to 0, z_2 to 1 and z_3 to ∞ .

I.e.

$$M(z) = \frac{(z - z_1)(z_2 - z_3)}{(z - z_3)(z_2 - z_1)} \quad \text{if } M(z_1) = 0, M(z_2) = 1, M(z_3) = \infty.$$

Proposition 3. If z_1, z_2, z_3 are distinct points in \mathbb{C}_∞ and T is a Moebius Transformation and $T(z_i) = w_i, i = 1, 2, 3$. Then

$$(z, z_1, z_2, z_3) = (w, w_1, w_2, w_3) \quad \forall z \in \mathbb{C}_\infty.$$

Proof. Let $S(z) = (z, z_1, z_2, z_3)$; then S is a Moebius mapping. If $M = S \circ T^{-1}$, then $M(w_1) = 0, M(w_2) = 1, M(w_3) = \infty$; hence, $S \circ T^{-1}(z) = (z, w_1, w_2, w_3) \forall z \in \mathbb{C}_\infty$. In particular, if $z = w$, and the desired result follows. \square

Proposition 4. If z_1, z_2, z_3 are distinct points in \mathbb{C}_∞ and w_1, w_2, w_3 are distinct points in \mathbb{C}_∞ too, then there is one and only one Moebius Transformation S such that $S(z_1) = w_1, S(z_2) = w_2, S(z_3) = w_3$.

Proof. Let $T(z) = (z, z_1, z_2, z_3), M(z) = (z, w_1, w_2, w_3)$ and let $S = M^{-1} \circ T$. Clearly S has the desired property. If R is another Moebius mapping, with $R(z_j) = w_j$ for $j = 1, 2, 3$ then $R^{-1} \circ S$ has three fixed points (z_1, z_2, z_3) . Hence, $R^{-1} \circ S = I$, or $S = R$. \square

From the definition and the properties above, we can say that, for any distinct $z_i, w_i \in \mathbb{C}_\infty, i = 1, 2, 3$, there is one, and only one, Moebius Transformation M , such that

$$M(z_i) = w_i; \quad \text{and} \quad \frac{(w - w_1)(w_2 - w_3)}{(w - w_3)(w_2 - w_1)} = \frac{(z - z_1)(z_2 - z_3)}{(z - z_3)(z_2 - z_1)}$$

With all these results, we have a way to calculate the coefficients of a Moebius Transformation defined by three points and their images, as proven in the next proposition.

Proposition 5. Given a Moebius Transformation $M(z) = \frac{az+b}{cz+d}$, where $w_i = M(z_i)$ for $i = 1, 2, 3$, we have:

$$a = \det \begin{pmatrix} w_1 z_1 & w_1 & 1 \\ w_2 z_2 & w_2 & 1 \\ w_3 z_3 & w_3 & 1 \end{pmatrix}, \quad b = \det \begin{pmatrix} w_1 z_1 & z_1 & w_1 \\ w_2 z_2 & z_2 & w_2 \\ w_3 z_3 & z_3 & w_3 \end{pmatrix},$$

$$c = \det \begin{pmatrix} z_1 & w_1 & 1 \\ z_2 & w_2 & 1 \\ z_3 & w_3 & 1 \end{pmatrix}, \quad d = \det \begin{pmatrix} w_1 z_1 & z_1 & 1 \\ w_2 z_2 & z_2 & 1 \\ w_3 z_3 & z_3 & 1 \end{pmatrix}.$$

Proof. We can rewrite the equation of Moebius Transformation M on the following form. Taking $w = M(z)$ we have

$$cwz - az + dw - b = 0$$

We know that the equation passes through the pair of points (z_i, w_i) and in an arbitrary pair (z, w) , so, we can set up the following linear system.

$$\begin{aligned} cwz - az + dw - b &= 0 \\ cw_1 z_1 - az_1 + dw_1 - b &= 0 \\ cw_2 z_2 - az_2 + dw_2 - b &= 0 \\ cw_3 z_3 - az_3 + dw_3 - b &= 0 \end{aligned}$$

Rewriting in matrix form, we have

$$\begin{pmatrix} wz & -z & w & -1 \\ w_1 z_1 & -z_1 & w_1 & -1 \\ w_2 z_2 & -z_2 & w_2 & -1 \\ w_3 z_3 & -z_3 & w_3 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

As we do not want the trivial solution, then

$$\det \begin{pmatrix} wz & -z & w & -1 \\ w_1 z_1 & -z_1 & w_1 & -1 \\ w_2 z_2 & -z_2 & w_2 & -1 \\ w_3 z_3 & -z_3 & w_3 & -1 \end{pmatrix} = 0$$

Doing cofactor expansion over this determinant in the first line, we get

$$\begin{aligned} & wz \cdot \det \begin{pmatrix} -z_1 & w_1 & -1 \\ -z_2 & w_2 & -1 \\ -z_3 & w_3 & -1 \end{pmatrix} + z \cdot \det \begin{pmatrix} w_1 z_1 & w_1 & -1 \\ w_2 z_2 & w_2 & -1 \\ w_3 z_3 & w_3 & -1 \end{pmatrix} \\ & + w \cdot \det \begin{pmatrix} w_1 z_1 & -z_1 & -1 \\ w_2 z_2 & -z_2 & -1 \\ w_3 z_3 & -z_3 & -1 \end{pmatrix} + \det \begin{pmatrix} w_1 z_1 & -z_1 & w_1 \\ w_2 z_2 & -z_2 & w_2 \\ w_3 z_3 & -z_3 & w_3 \end{pmatrix} = 0 \end{aligned}$$

Organizing the signals

$$\begin{aligned} & wz \cdot \det \begin{pmatrix} z_1 & w_1 & 1 \\ z_2 & w_2 & 1 \\ z_3 & w_3 & 1 \end{pmatrix} - z \cdot \det \begin{pmatrix} w_1 z_1 & w_1 & 1 \\ w_2 z_2 & w_2 & 1 \\ w_3 z_3 & w_3 & 1 \end{pmatrix} \\ & + w \cdot \det \begin{pmatrix} w_1 z_1 & z_1 & 1 \\ w_2 z_2 & z_2 & 1 \\ w_3 z_3 & z_3 & 1 \end{pmatrix} - \det \begin{pmatrix} w_1 z_1 & z_1 & w_1 \\ w_2 z_2 & z_2 & w_2 \\ w_3 z_3 & z_3 & w_3 \end{pmatrix} = 0 \end{aligned}$$

Since the equation

$$0 = cwz - az + dw - b$$

also holds, we can equate the coefficients of these expressions to obtain

$$\begin{aligned} c &= \det \begin{pmatrix} z_1 & w_1 & 1 \\ z_2 & w_2 & 1 \\ z_3 & w_3 & 1 \end{pmatrix}, \quad a = \det \begin{pmatrix} w_1 z_1 & w_1 & 1 \\ w_2 z_2 & w_2 & 1 \\ w_3 z_3 & w_3 & 1 \end{pmatrix}, \\ d &= \det \begin{pmatrix} w_1 z_1 & z_1 & 1 \\ w_2 z_2 & z_2 & 1 \\ w_3 z_3 & z_3 & 1 \end{pmatrix}, \quad b = \det \begin{pmatrix} w_1 z_1 & z_1 & w_1 \\ w_2 z_2 & z_2 & w_2 \\ w_3 z_3 & z_3 & w_3 \end{pmatrix}. \end{aligned}$$

□

Before we continue, we need to establish some notation for a better understanding of the transformations. As our algorithm works like a black box, just needing the initial and final points, the natural notation for a Moebius Transformation is the following:

$$T : [z_1, z_2, z_3] \rightarrow [w_1, w_2, w_3], \quad z_1, z_2, z_3, w_1, w_2, w_3 \in E$$

This notation says that T is the Moebius Transformation in \mathbb{C}_∞ that maps $z_i \in E$ to $w_i \in E$, where E notates the equirectangular domain.

The method works in the following form: we ask the user to specify initial and final positions of three points in the input image. Applying inverse equirectangular projection, we map the image and these coordinates to the visible sphere and then we project the sphere to the complex plane using stereographic projection. Therefore, using

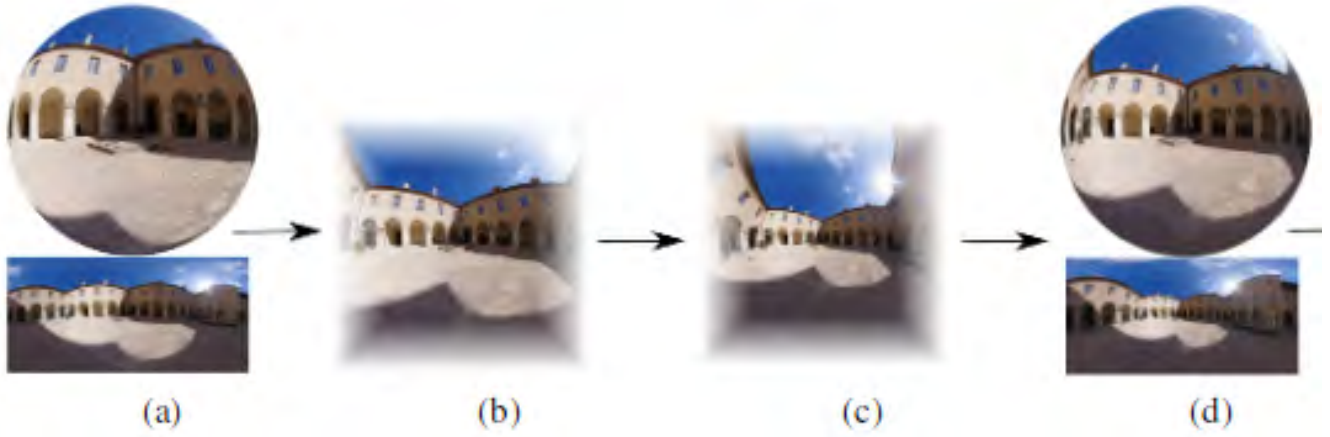


Figure 4: Scheme of the method. (4a) Map the equirectangular image to the sphere; (4b) Project to the complex plane using stereographic projection; (4c) Apply the Moebius Transformation; (4d) Return to the sphere and project back to the equirectangular domain.

proposition 5 and the prescribed points, we define a Moebius Transformation and apply it over the image. Using inverse stereographic projection we map the transformed image to the sphere and project back to the equirectangular domain. Figure 4 shows the idea schematically.

If $C = 0$ in the general Moebius Transformation $M(z) = \frac{Az+B}{Cz+D}$ then $M(z) = Az + B$. If we write $A = \rho e^{i\alpha}$, then it can be viewed as a composition of an origin-centered rotation by α , an origin-centered expansion by ρ and a translation by B . These transformations can be viewed on the Riemann sphere Σ in Figure 5.

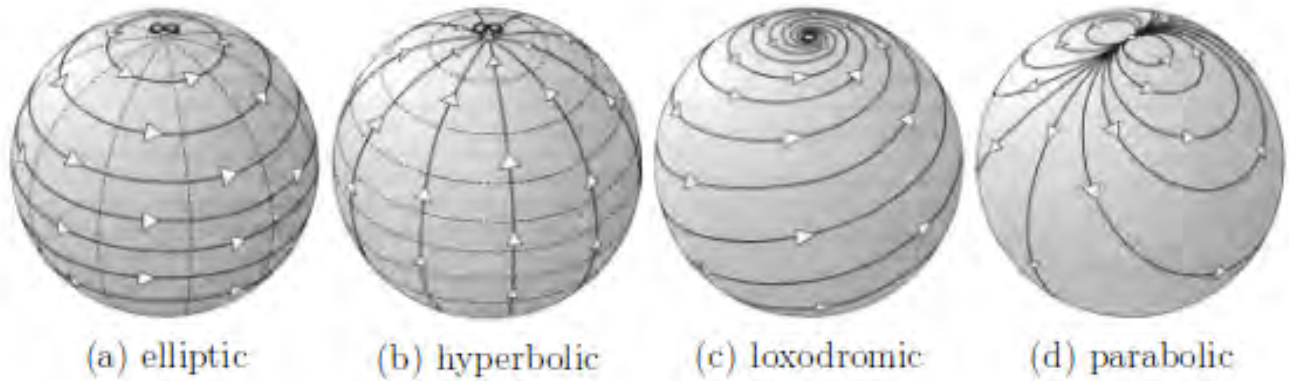


Figure 5: Canonic Moebius Transformations in the Riemann sphere Σ .

With $\alpha \neq 0$, Figure 5a illustrates that the rotation $z \rightarrow e^{i\alpha} z$ in \mathbb{C}_∞ induces a rotation of Σ about the vertical axis through its centre. Horizontal circles on Σ rotate into themselves and are therefore called *invariant curves* of the transformation. This pure rotation is an example of an *elliptic* Moebius Transformation. Figure 6 show us an

example of an image in the equirectangular domain rotated by an elliptic Moebius Transformation.



Figure 6: An example of elliptic Moebius Transformation. (6b) the result of a rotation by 10° degrees ($T : [(-0.3 \ 0), (0 \ 0), (0.3 \ 0)] \rightarrow [(-0.2954 \ -0.0521), (0 \ 0), (0.2954 \ 0.0521)]$).

If $\rho > 1$, Figure 5b illustrate the transformation on Σ induced by a corresponding origin-centered expansion in \mathbb{C}_∞ , $z \rightarrow \rho z$. If $\rho < 1$, then we have a contraction of \mathbb{C}_∞ , this time, the invariant curves are the great circles through the fixed points at the poles. This pure expansion is an example of *hyperbolic* Moebius Transformation. This is the kind of transformation explored in [4], which has an effect of changing the field of view in the sphere. Figure 7 show us an example of images in the equirectangular domain scaled by a hyperbolic Moebius Transformation.

Figure 5c illustrates the combined effect of rotation and expansion. The invariant curves are the illustrated “spirals”. This combination is an example of a *loxodromic* Moebius Transformation. Figure 8 shows an example of loxodromic Moebius Transformation in complex plane.

Finally, Figure 5d illustrates a translation. Since the invariant curves in \mathbb{C}_∞ are the family of parallel lines in the direction of the translation, the invariant curves on Σ are the family of circles whose common tangent at ∞ is parallel to the invariant lines in \mathbb{C}_∞ . Since ∞ is the only fixed point, a pure translation is an example of a *parabolic* Moebius Transformation. Figure 9 shows the result of a translation in the complex plane. (Obs: the high non-linearity of stereographic projection makes a different effect than a simple translation).

Figure 10 shows how the specification of three points can be used to rectify lines that are straight in the world but appear curved in the equirectangular image.

5 Local Moebius Transformations

The method we have just presented affects all the equirectangular image, as can be seen in Figure 10b. For localized manipulations such as line rectification or local zoom, it is desired the Moebius transformation to affect only regions of interest. In this section, we introduce a new method to apply Moebius Transformations locally.

From the grid of pixels positions of a given image A, X for the x-position and Y for the y-position, we produce a new image B using the method in the previous section, with a new grid of pixels positions U and V, respectively. Then, we get the difference of positions $U - X = \Delta U$ and $V - Y = \Delta V$. Therefore, we apply a smoothing function $F(X, Y)$ over ΔU and ΔV , generating a smooth repositioning of pixels $\bar{\Delta U}$ and $\bar{\Delta V}$. Then we add these new differences to the original image, getting $\bar{U} = X + \bar{\Delta U}$ and $\bar{V} = Y + \bar{\Delta V}$. Interpolating the pixels to these new positions, we get a new image \bar{B} with a localized Moebius Transformation.

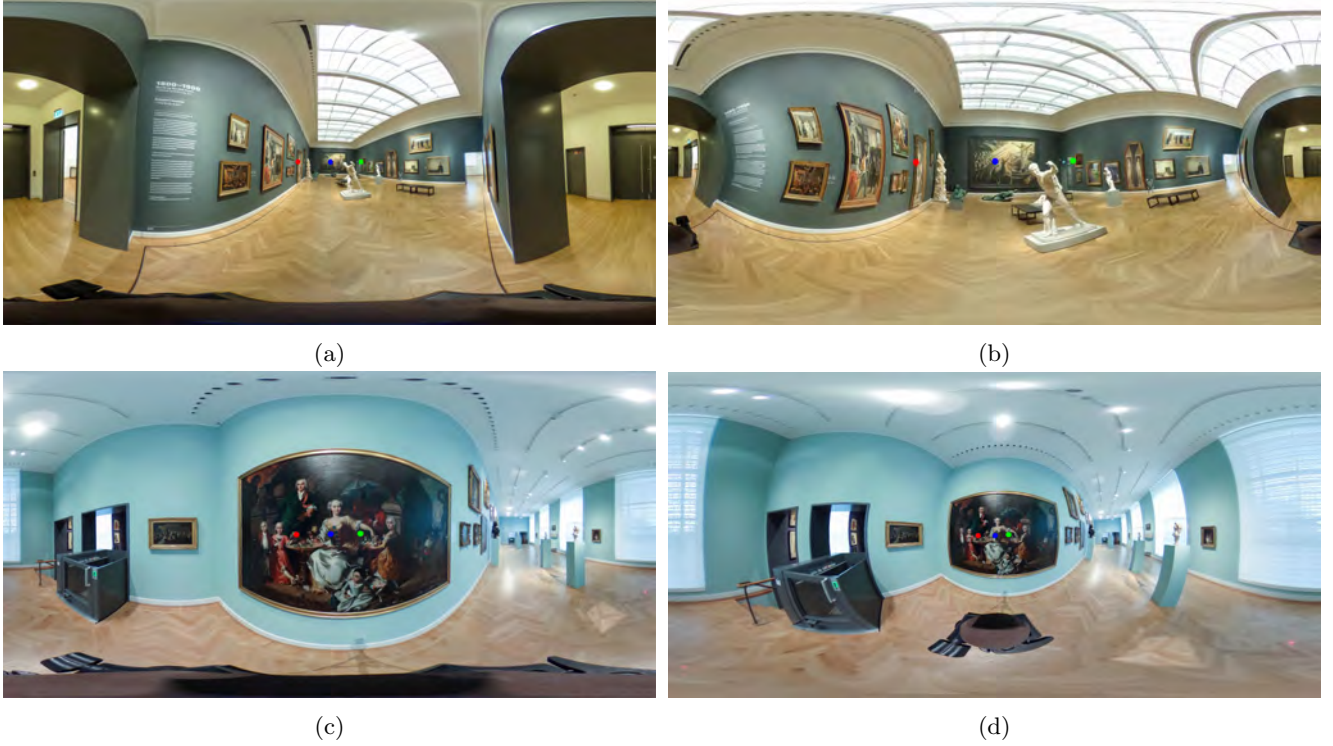


Figure 7: An example of hyperbolic Moebius Transformation. (7b) the result of a scaling by 3 ($T : [(-0.304 \ 0), (0 \ 0), (0.304 \ 0)] \rightarrow [(-0.7521 \ 0), (0 \ 0), (0.7521 \ 0)]$); (7d) the result of a scaling by 0.5 ($T : [(-0.304 \ 0), (0 \ 0), (0.304 \ 0)] \rightarrow [(-0.1497 \ 0), (0 \ 0), (0.1497 \ 0)]$).

In our MATLAB GUIDE[®] implementation, the chosen smoothing function was a general 2-D gaussian function

$$G(x, y) = A \cdot \exp(-\alpha(x - x_0)^2 - 2\beta(x - x_0)(y - y_0) + \gamma(y - y_0)^2)$$

where

$$\alpha = \frac{\cos^2 \theta}{2\sigma_x^2} + \frac{\sin^2 \theta}{2\sigma_y^2}, \beta = \frac{\sin 2\theta}{4\sigma_x^2} - \frac{\sin 2\theta}{4\sigma_y^2} \text{ and } \gamma = \frac{\sin^2 \theta}{2\sigma_x^2} + \frac{\cos^2 \theta}{2\sigma_y^2}.$$

In the interface, the user specifies the parameter of some elliptical region centered in (x_0, y_0) with axis of the value $R_x = 2\sigma_x$ and $R_y = 2\sigma_y$ and with a θ counterclockwise rotation, this region will be affected by a gaussian function with amplitude 1, generating a smooth dissipation of the Moebius Transformation specified by 3 input/output coordinates. The Figure 11 shows a comparison between the two methods. The next sections will discuss how obtain these results and the features of the interface.

6 Results and Limitations

Now, we present some results and discuss them and the limitations of our method.

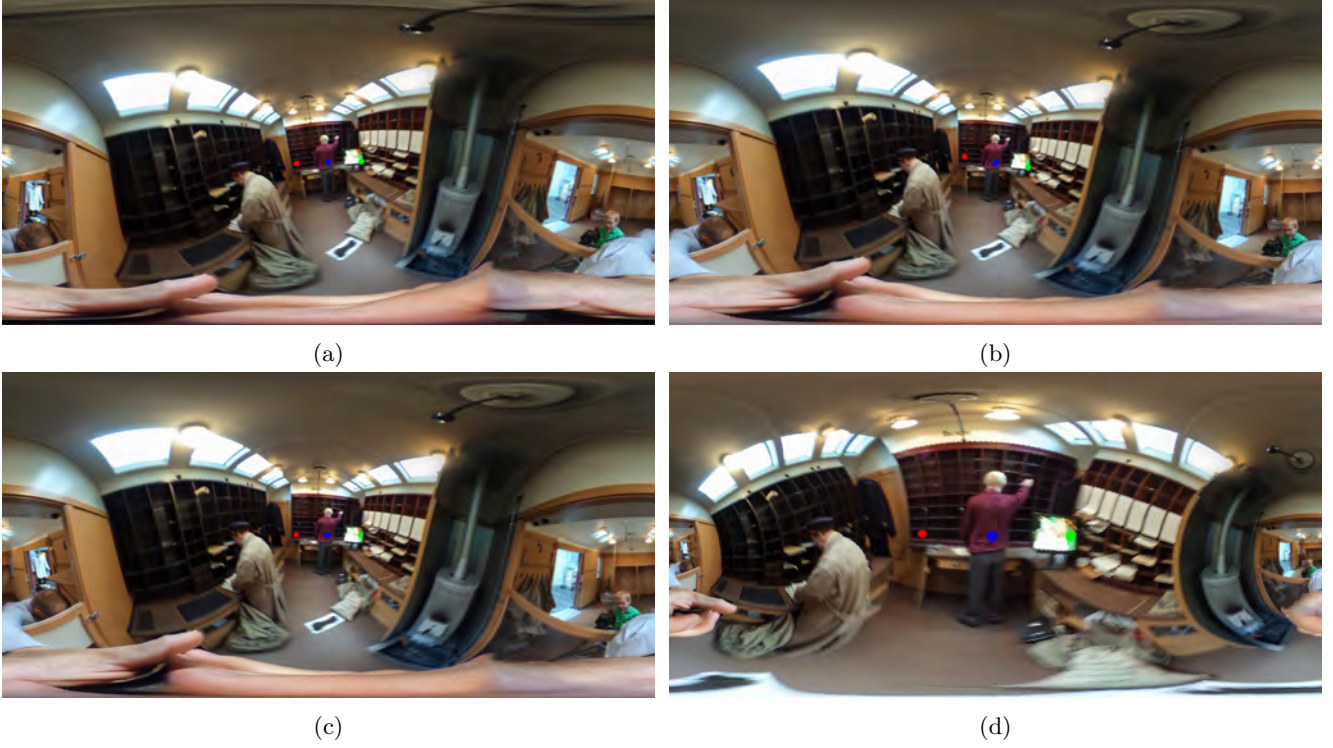


Figure 8: An example of loxodromic Moebius Transformation. (8d) the result of a rotation by 10° degree and a scaling by 2.5 ($T_1 : [(-0.3 \ 0), (0 \ 0), (0.3 \ 0)] \rightarrow [(-0.2954 \ -0.0521), (0 \ 0), (0.2954 \ 0.0521)]$) ($T_2 : [(-0.304 \ 0), (0 \ 0), (0.304 \ 0)] \rightarrow [(-0.7175 \ 0), (0 \ 0), (0.7175 \ 0)]$).

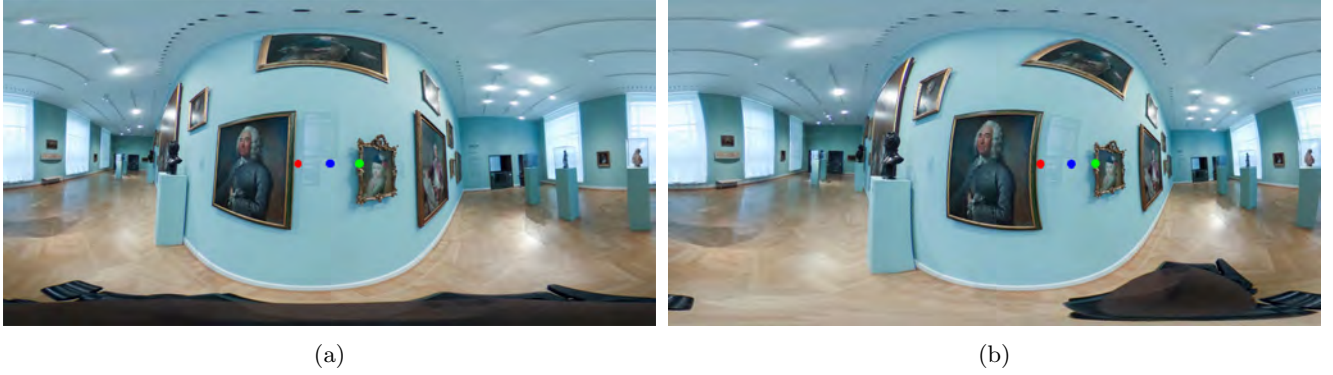


Figure 9: An example of parabolic Moebius Transformation. (9b) the result of a translation by 0.8 in complex plane ($T : [(-0.304 \ 0), (0 \ 0), (0.304 \ 0)] \rightarrow [(0.444 \ 0), (0.727 \ 0), (0.985 \ 0)]$).

First of all, we have line rectification without affecting undesired regions of the image. The most effective approach consists in selecting the region with a narrow gaussian, with two fixed points and a free point, mostly the



(a)

(b)

Figure 10: Line rectification with Moebius Transformation. 10b The result of line rectification ($T : [(-1.06 \ 0.32), (1.08 \ 0.32), (0 \ 0.6)] \rightarrow [(-1.06 \ 0.32), (1.08 \ 0.32), (0 \ 0.32)]$)



(a)

(b)

Figure 11: Comparison between methods. (11a) The result of global method ($T : [(-1.06 \ 0.32), (1.08 \ 0.32), (0 \ 0.6)] \rightarrow [(-1.06 \ 0.32), (1.08 \ 0.32), (0 \ 0.32)]$); (11b) The result of local method ($T : [(-1.06 \ 0.3), (1.08 \ 0.32), (0 \ 0.7)] \rightarrow [(-1.06 \ 0.3), (1.08 \ 0.32), (0 \ 0.42)]$, $C = (0.01 \ 0.45)$, $R_x = 2$, $R_y = 0.56$, $\theta = 0$).

point in the middle. Figure 12 show us an example:

This approach have some limitations. When the image possess too much information around the line, the rectification causes inappreciable distortions, as can be seen in Figure 13b.

When the line to be rectified is too long, the smoothness of the function causes incomplete rectification, as you can see in Figure 14b. A way to work around this issue is to make successive transformations over the line, but this fixing requires more work from the user as well. The result in Figure 14d illustrates this point.

Now we present the application of zooming in and out without affecting undesired regions. This application can be useful for security footage to identify the suspect without compromising the other regions covered by the camera. An effective approach to perform a zoom specifying three points is move two points away from the region to be zoomed while maintaining a fixed point. Another way is setting a triangle with the three points over the region to be affected, and move them away. Zoom out can be achieved in a similar way, making points closer. Figure 15, show us an example of local zoom.

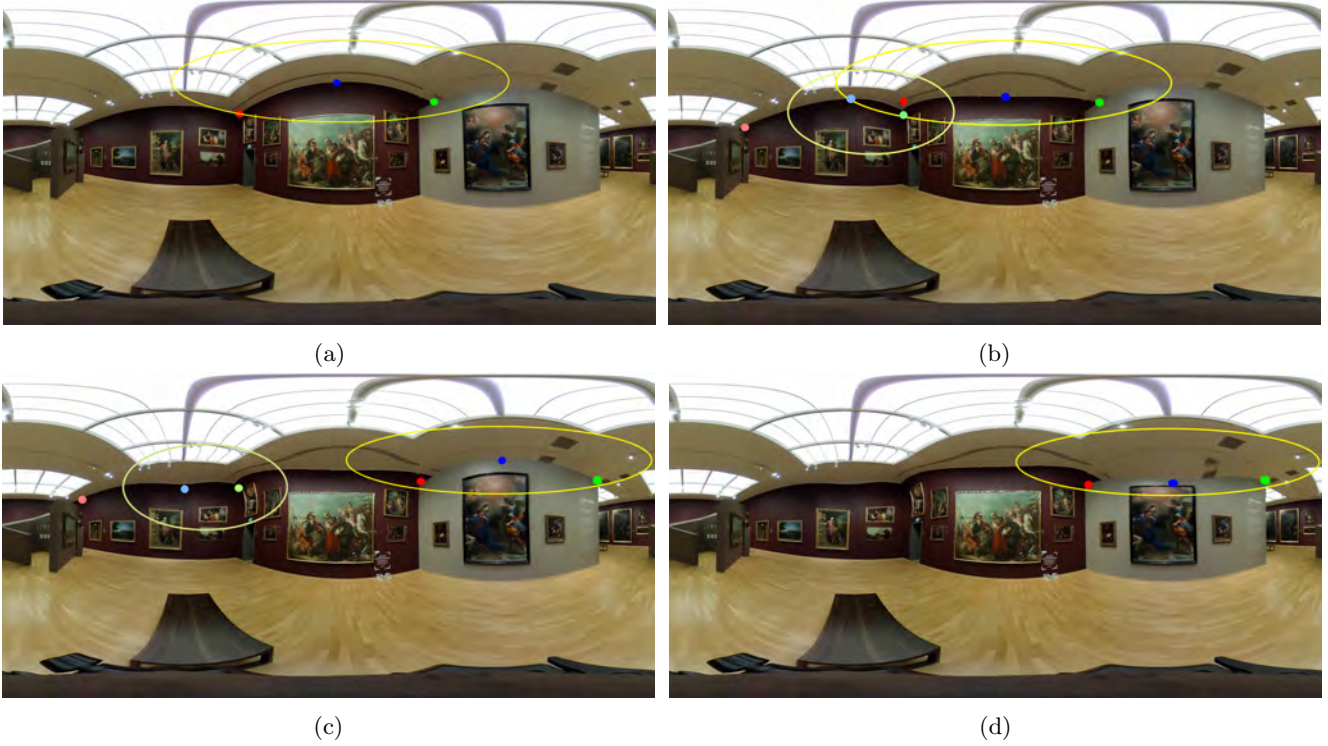


Figure 12: An example of line rectification with local Moebius Transformation. $(T_1 : [(-0.88 \ 0.46), (0.99 \ 0.56), (0.02 \ 0.76)] \rightarrow [(-0.88 \ 0.58), (0.99 \ 0.56), (0.08 \ 0.62)], C = (0.08 \ 0.75), R_x = 1.6, R_y = 0.40, \theta = 0)$; (12c) $(T_2 : [(-2.40 \ 0.33), (-0.9 \ 0.45), (-1.4 \ 0.6)] \rightarrow [(-2.40 \ 0.33), (-0.9 \ 0.45), (-1.42 \ 0.46)], C = (-1.21 \ 0.48), R_x = 0.8, R_y = 0.40, \theta = 0)$; (12d) The final result $(T_3 : [(0.88 \ 0.5), (2.6 \ 0.52), (1.66 \ 0.72)] \rightarrow [(0.88 \ 0.5), (2.6 \ 0.52), (1.66 \ 0.5)], C = (1.66 \ 0.7), R_x = 1.45, R_y = 0.31, \theta = 0)$.

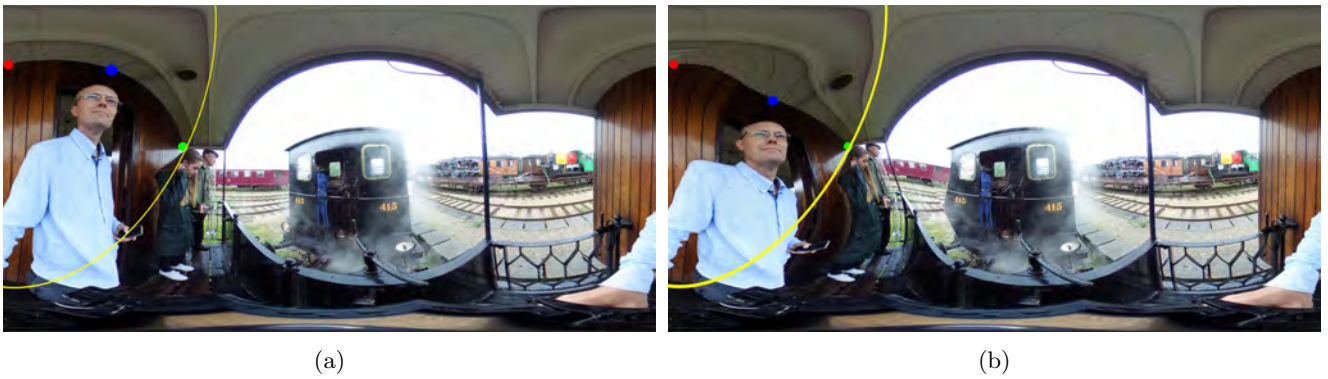


Figure 13: Limitation in line rectification. (13b) a bad result $(T : [(-3.1 \ 1), (-1.4 \ 0.2), (-2.2 \ 0.99)] \rightarrow [(-3.1 \ 1), (-1.4 \ 0.2), (-2.16 \ 0.66)], C = (-2.34 \ 0.82), R_x = 1.1, R_y = 2.04, \theta = -0.4)$

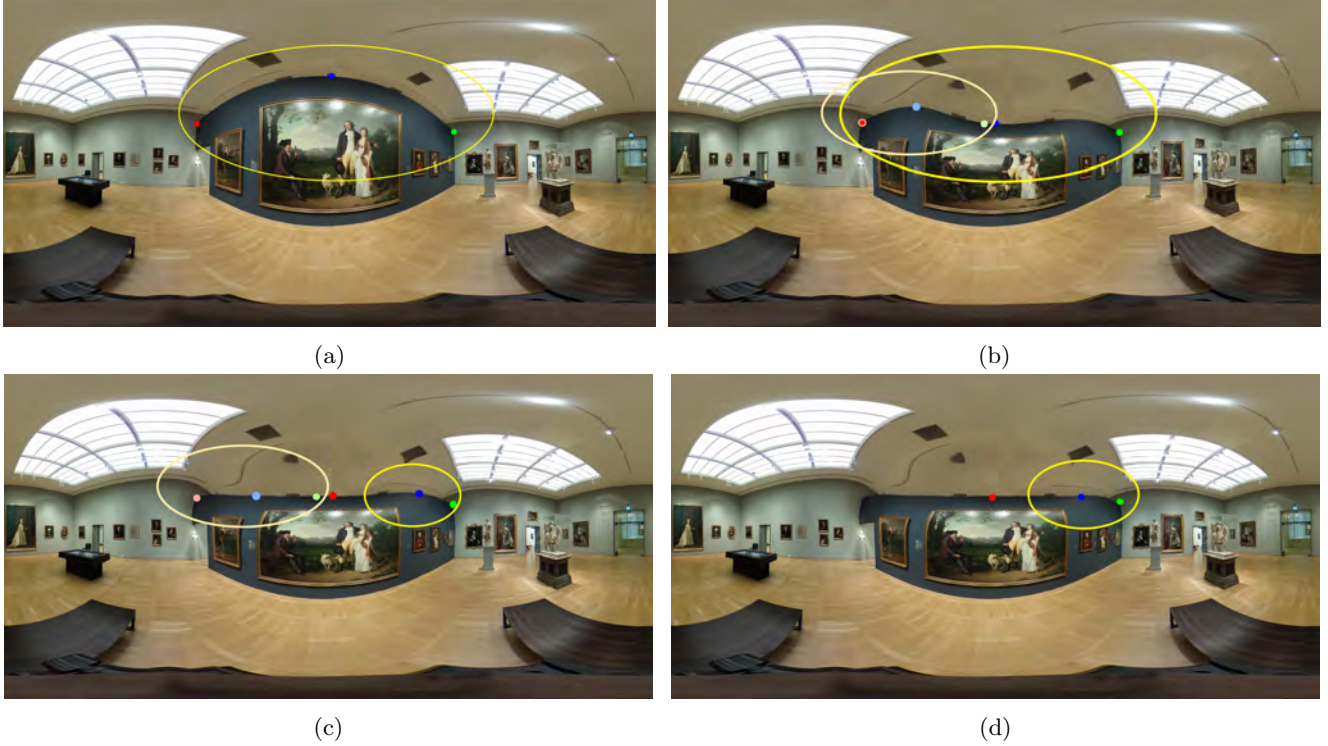


Figure 14: An example of bigger line rectification with local Moebius Transformation. (14b) Incorrect result $(T_1 : [(-1.26 \ 0.4), (1.2 \ 0.32), (0 \ 0.84)] \rightarrow [(-1.26 \ 0.4), (1.2 \ 0.32), (0 \ 0.37)], C = (0.04 \ 0.48), R_x = 1.52, R_y = 0.64, \theta = 0)$; (14c) $(T_2 : [(-1.3 \ 0.38), (-0.1 \ 0.38), (-0.76 \ 0.54)] \rightarrow [(-1.3 \ 0.36), (-0.1 \ 0.38), (-0.72 \ 0.38)], C = (-0.8 \ 0.48), R_x = 0.8, R_y = 0.4, \theta = 0)$ (14d) Better result $(T_3 : [(0 \ 0.36), (1.2 \ 0.3), (0.85 \ 0.4)] \rightarrow [(0 \ 0.36), (1.22 \ 0.34), (0.85 \ 0.375)], C = (0.8 \ 0.4), R_x = 0.45, R_y = 0.32, \theta = 0)$.

Another result that can be useful for security footage is the translation of an object near the poles of the sphere to the equatorial region, assisting the recognition of too much distorted objects or people. One limitation of this result is the pixeling in the image. This occurs because the points in the poles are too close together. So, for a better visualisation, it is recommended the use of high resolution images, what can be a practical limitation. Fish-eye cameras have better resolutions near the poles, so the problem of pixeling can be avoided. The example in Figure 16 used a 2048x1024 image. Another way of helping the recognition is zooming out the center of the image (an hyperbolic transformation) in such a way that pixels near the poles come closer and then zooming in the object. This approach will cause high distortion, so, we apply a correction using a 2 fixed point rotation. A good point of this second approach is that it does not require a high resolution (we used a 2048x1024 image, that can be computed in almost real time). Figure 17 show us a step-by-step result of this approach over the same image of the previous example.

To conclude the result section we present the side effects of zooming in and zooming out on final quality of the image. Figure 18 shows an example of these effects. The first part of the test was done performing a contraction around the central points, then restoring to the initial positions. The greater the contraction, the greater is the loss of image information around the central points. The chosen contraction values are $k_1 = 0.5, k_2 = 0.3, k_3 = 0.15$.



(a)

(b)

Figure 15: Zoom in $(T_1 : [(-1.1 \ -0.3), (-0.6 \ -0.3), (-0.86 \ 0)] \rightarrow [(-1.3 \ -0.5), (-0.4 \ -0.5), (-0.86 \ 0.44)], C = (-0.86 \ -0.22), R_x = 0.28, R_y = 0.3, \theta = 0, T_2 : [(2.5 \ -0.3), (2.9 \ -0.3), (2.7 \ 0)] \rightarrow [(2.1 \ -0.6), (3.1 \ -0.6), (2.7 \ 0.6)], C = (2.7 \ -0.2), R_x = 0.2, R_y = 0.32, \theta = 0)$.



(a)

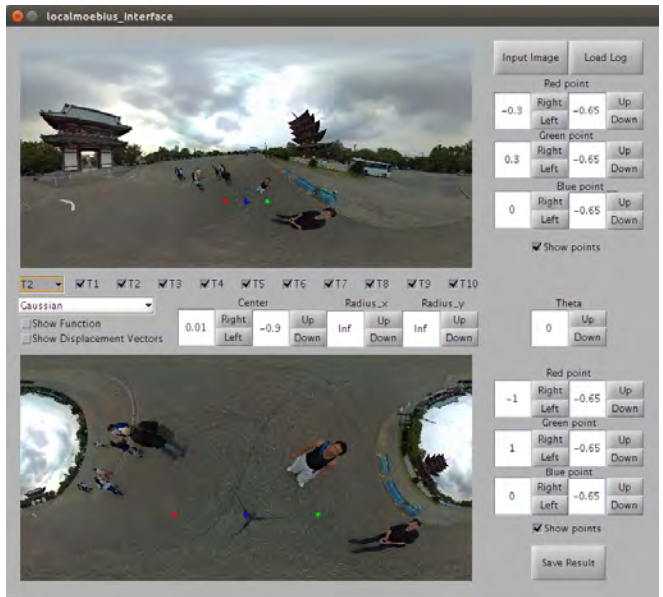
(b)

Figure 16: Recognition of very distorted objects. $(T : [(0 \ -1.5), (0.3 \ -1.5), (0.6 \ -1.5)] \rightarrow [(0 \ -1), (0.3 \ -1), (0.6 \ -1)], C = (0 \ 0), R_x = \infty, R_y = \infty, \theta = 0)$.

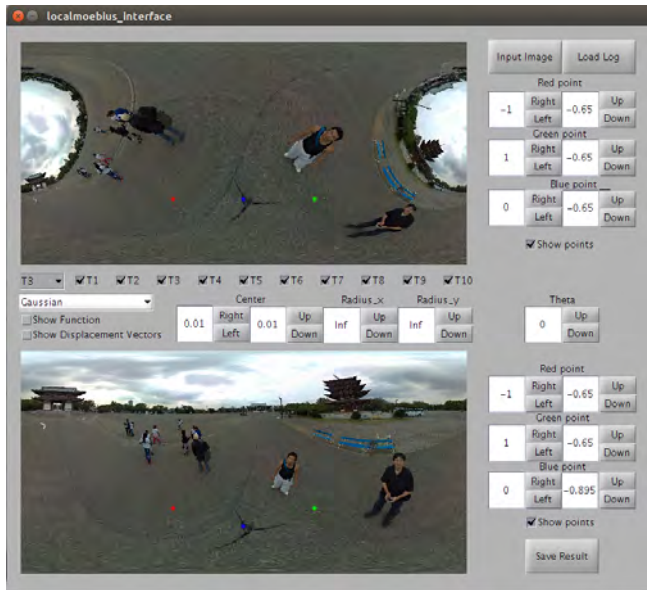
The second part was done by doing an expansion around central points, then restoring to the initial positions. The greater the expansion, the greater is the loss of image information at the peripherals points. The chosen expansion values are $k_4 = 2.7$ and $k_5 = 5.0$. The result was generated using a 2048x1024 equirectangular image.



(a)



(b)



(c)



(d)

Figure 17: Recognition of very distorted objects (The transformations parameters are in the images).

7 Implementation Features

We now discuss the features of our implementation in MATLAB GUIDE[®]. A screen shot of the interface is available in Figure 19.

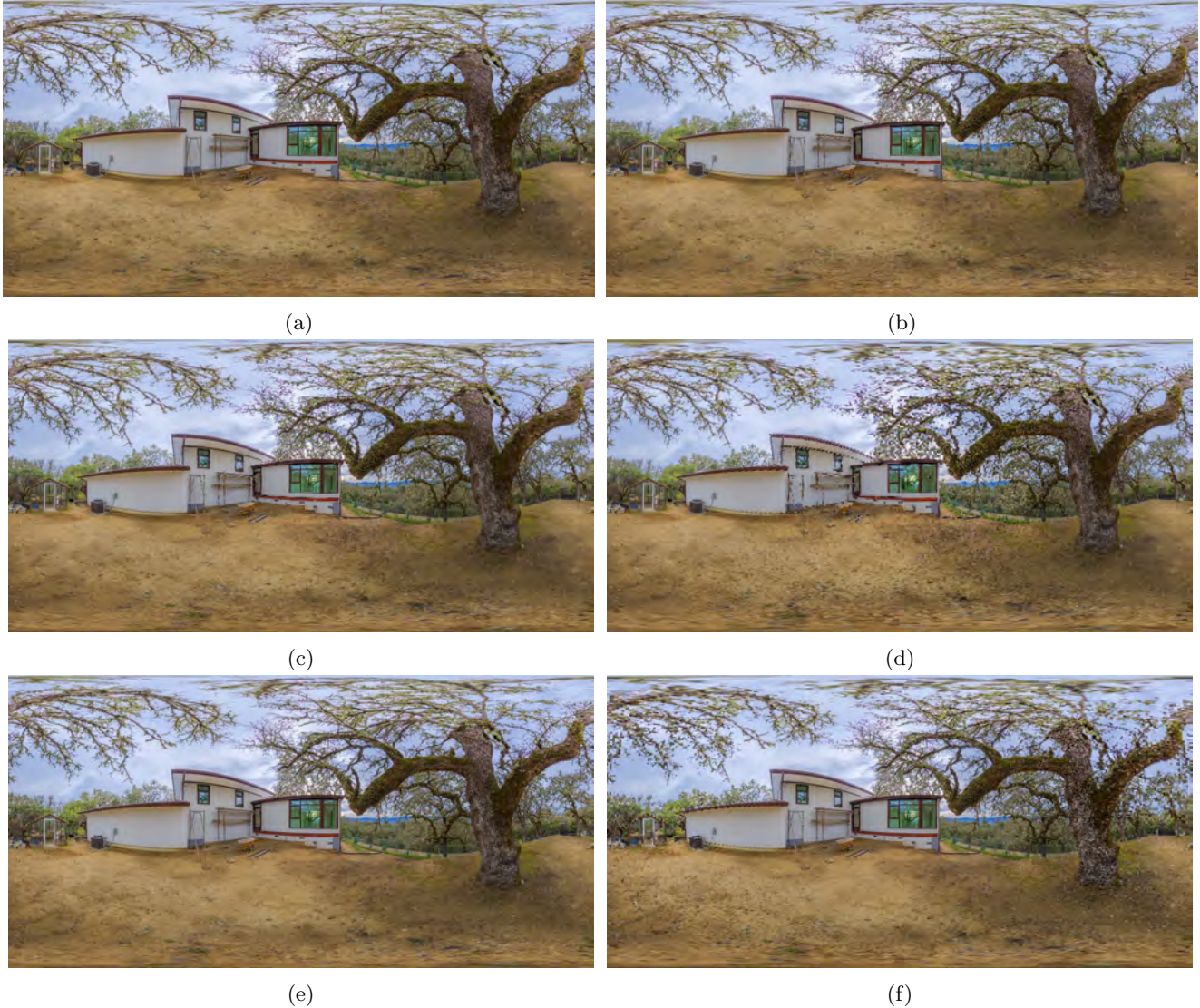


Figure 18: (18a) Original Image; (18b) Contraction by $k_1 = 0.5$; (18b) Contraction by $k_2 = 0.3$; (18c) Contraction by $k_3 = 0.15$; (18d) Expansion by $k_4 = 2.7$; (18e) Contraction by $k_5 = 5.0$.

In the upper half of the interface, we have the button “Input Image” that loads the initial image to be worked. The button “Load Log” loads a .dat text file that stores the transformations. This button allows the user to continue the project at another time, without needing to set all the transformations again. The red, green and blue input points spaces for coordinates allows the user to write x and y coordinates in the equirectangular domain for every point. The buttons “Up”, “Down”, “Left”, Right add a 0.02 (or -0.02) increment to the coordinates, allowing fine adjusts. The selection box “Show points” allows hiding and showing the input points in the initial image, for a better visualization of the transformation.

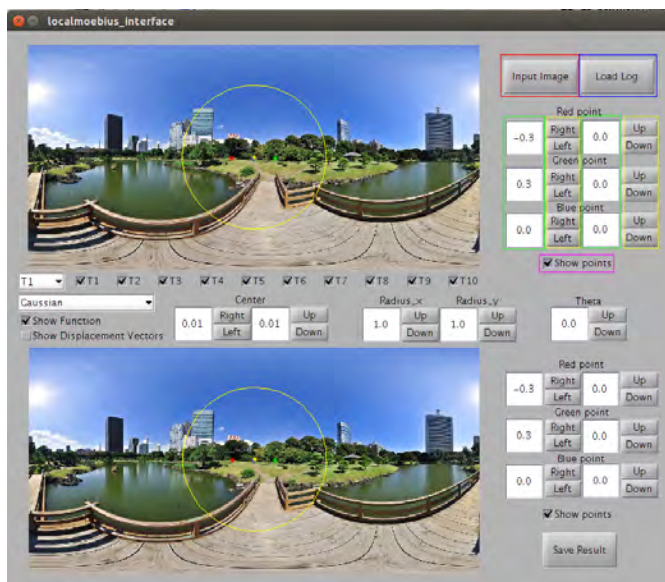


Figure 19: in red: Input image button; in blue: load log button; in green: spaces for x and y coordinates of input points; in yellow: Right, Left, Up and Down buttons; in pink: Show points box.

In the middle of the interface we have the pop-up button that allows changing the current transformation, with a maximum of ten transformations. It updates the upper and the lower images as well, where the upper image is the result of the previous transformation and the lower image is the result of the current transformation. Right below, we have another pop-up button that allows changing the smoothing function. In our prototype we have only the Gaussian function as smoothing function, other functions will be added in the future. The select box “Show function” allows hiding and showing the smoothing function in the initial and the result image, for a better visualization of the transformation. The select box “show displacement vectors” fills the result image with vectors showing how the pixels moved with the transformation (as you can see in Figure 20). The ten select boxes T_i allow the user choose what transformations will be applied over the image. The transformations of this method are not commutative, so if a transformation affects the same region of a previous transformation, unchecking the previous transformation may cause undesired results. We have the space for defining the coordinates of the center of the smoothing function, as well as the radius of it and the angle of rotation. The “Up”, “Down”, “Left” and “Right” buttons works in the same way as the previous buttons.

In the lower half of the interface, we have the red, green and blue output points spaces for coordinates, allowing the user to write u and v coordinates for every point. The buttons “Up”, “Down”, “Left” and “Right” works just like the others. The select box “Show points” allows hiding and showing the output points in the result image. The last button “Save result” saves the current result image and a .dat file with all the transformations parameters.



Figure 20: in red: current transformation pop-up button; in blue: smoothing function pop-up button; in green: Show function box; in yellow: show displacement vectors box; in pink: T_i boxes; in orange: Center coordinates of smoothing function, R_x, R_y and θ , respectively; in cyan: Right, Left, Up and Down buttons.



Figure 21: in red: spaces for u and v coordinates of output points; in blue : Right, Left, Up and Down buttons; in green: Show points box; in yellow: Save result button.

8 Conclusion

The proposed method allows a high degree of freedom for processing omnidirectional images. We have been able to provide a way to improve such images, which, along with the idea of combining transformations in an intuitive way, leaves their use in a very accessible manner for people who are not in this area of study. The idea of a smoothing function limiting the reaching of the transformation allows multiple transformations over the image making possible other ways of improving the image. Line rectification and face or object recognition for security footage are only some possibilities of the implementation and show the flexibility of the method. Our implementation resulted in a intuitive and interactive interface that can compute the transformations almost in real time, even being a prototype that can be ported to more efficient languages (as C/C++).

A natural extension of this work is to apply our method to omnidirectional videos. A future application area of this project is virtual reality, since the omnidirectional images and videos represent more visible information than usual cameras. Another possibility is to process images in a dome.

We also consider to extend our method to find a Moebius transformation that satisfies more than 3 points in a least-squares sense.

Acknowledgements

First of all, we would like to thank PICME and CNPq for funding this scientific initiation project in Mathematics. We thank the coordination of the Mathematics course from UFSC and IMPA for the structural support for this project. And of course, we thank friends and family who supported us in pursuit of this goal. For releasing their images under a Creative Common license for common use, we thank the following Flickr users: Luca Biada, figure 2; Peter Leth, figures 6a, 7c, 7a, 8a, 9a, 12a, 13a, 14a and 15a; rui_ke, figure 10a; Masakazu Matsumoto, figure 16a and 17a; Bob Dass, figure 18a; Masato OHTA, figure 19.

References

- [1] Robert Carroll, Maneesh Agrawal, and Aseem Agarwala. “Optimizing Content-preserving Projections for Wide-angle Images”. In: *ACM SIGGRAPH 2009 Papers*. SIGGRAPH '09. New York, NY, USA: ACM, 2009, 43:1–43:9.
- [2] John B. Conway. *Functions of One Complex Variable*. Ed. by P. R. Halmos, F. W. Gehring, and C. C. Moore. Second Edition. Springer-Verlag, 1978.
- [3] Johannes Kopf et al. “Locally Adapted Projections to Reduce Panorama Distortions”. In: *Computer Graphics Forum* (2009).
- [4] Luis Penaranda, Luiz Velho, and Leonardo Sacht. “Real-time Correction of Panoramic Images using Hyperbolic Möbius Transformations”. In: *Journal of Real-Time Image Processing* (May 2015), p. 19.
- [5] Leonardo Koller Sacht. “Content-Based Projections for Panoramic Images and Videos”. MA thesis. Instituto Nacional de Matematica Pura e Aplicada, Apr. 2010.
- [6] Saul Schleimer and Henry Segerman. “Squares that Look Round: Transforming Spherical Images”. In: *CoRR* abs/1605.01396 (2016). URL: <http://arxiv.org/abs/1605.01396>.
- [7] L. Zelnik-Manor, G. Peters, and P. Perona. “Squaring the Circle in Panoramas”. In: *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*. ICCV '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 1292–1299.

- [8] 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. New York, NY, USA: ACM, 1995, pp. 257–264.