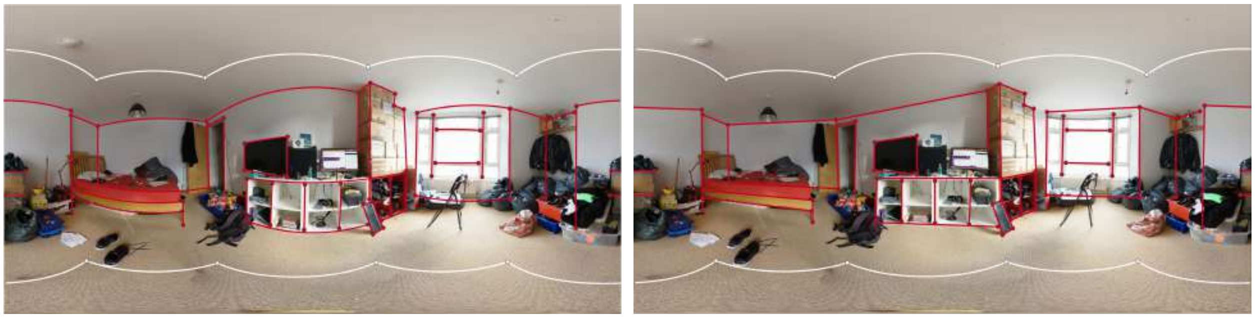


Input image

Result

Figure 1: The user marks in red the lines that (s)he wants to be rectified in the final result. Each individual line defines a Möbius transformation that makes points on it colinear. Linear blend skinning with bounded biharmonic weights combine the multiple transformations into the final result. In this example, the user also specifies spherical cages (in white) to avoid the transformations to affect the poles of the sphere. Both input image and result contain the full spherical field of view around a viewpoint.

Bounded biharmonic blending of Möbius transformations for flexible omnidirectional image rectification

Leonardo Souto Ferreira[1,2] and Leonardo Sacht[1]

[1]UFSC - Universidade Federal de Santa Catarina - Campus Universitário Trindade, Florianópolis 8804-900, Brazil
[2]UNESP - Universidade Estadual Paulista Júlio de Mesquita Filho, IFT - Instituto de Física Teórica, Rua Dr. Bento Teobaldo Ferraz, 271 - Bloco II - Várzea da Barra Funda, São Paulo 01140-070, Brazil

November 21, 2020

Abstract

We propose a new method to correct distortions in equirectangular images, i.e., images that use the longitude/latitude representation to describe the full spherical field of view around a given viewpoint. We show that Möbius transformations are the correct mathematical tool to deal with the conflicting distortions in this setting: they are conformal, are able to rectify lines, perform rotations, translations and scales on the sphere and are bijective.

Multiple transformations are specified through points, lines and cages on the sphere. We associate three points that uniquely define a Möbius transformation to each one of these geometric handles. Linear blend skinning with bounded biharmonic weights combine the different transformations into a single full spherical modified image.

We present a collection of results in challenging settings and show that our method is more flexible and produces higher quality results when compared to previous methods.

Keywords Wide-angle images, Möbius transformations, linear blend skinning

1 Introduction

Depicting scenes with minimally distorted straight lines and objects has been a challenge faced by painters, architects and computer scientists over the last centuries. The commonly used perspective projection generates results with perfectly straight lines but also with stretched objects as the field of view becomes wider. This observation motivated Renaissance painters to combine multiple local perspective projections in the same image to correct objects with high distortion. One famous example of the use of this technique is the School of Athens by Raphael.

More recently the advance of computer vision techniques brought new life to the problem. Omnidirectional images (Figure 1, left), which represent the complete color information of the visible sphere around a viewpoint, have more distortions than other previous representations of a scene. The longitude/latitude parameterization (a.k.a. equirectangular format) bends straight lines and introduces stretching on objects. On the other hand, it presents the advantages of not missing any information of the scene visible from the viewpoint and of being widely known and adopted by both amateur and professional photographers.

Due to these distortions and the much wider field of view, the combination of perspective projections is no longer effective to correct all the artifacts in these images. Instead, we propose to combine *Möbius transformations*, which are the only conformal bijective transformations on the sphere. In our context, this means that they stretch objects equally in all directions and produce results that contain the full visible information from the input.

The well known stereographic projection allows a bijective correspondence between the sphere and the complex plane with an additional point at infinity. Using this correspondence, the Möbius transformations defined as

$$\begin{aligned} M : \mathbb{C} \cup \{\infty\} &\to \mathbb{C} \cup \{\infty\} \\ z &\mapsto \frac{az+b}{cz+d}, \quad ac - bd \neq 0, \end{aligned} \tag{1}$$

induce mappings on the sphere. Some basic transformations such as scalings, rotations and translations are obtained by properly setting the complex coefficients in (1). Other effects can be achieved by specifying new positions to three points (see Section 3.1 for more details). In particular, three points on a bent line can be colinearized to achieve line rectification (Figure 2 (c)). This is effective for rectifying *one* single line, but an omnidirectional image usually has lots of bent lines such as the ones marked in red in Figure 1 (left).

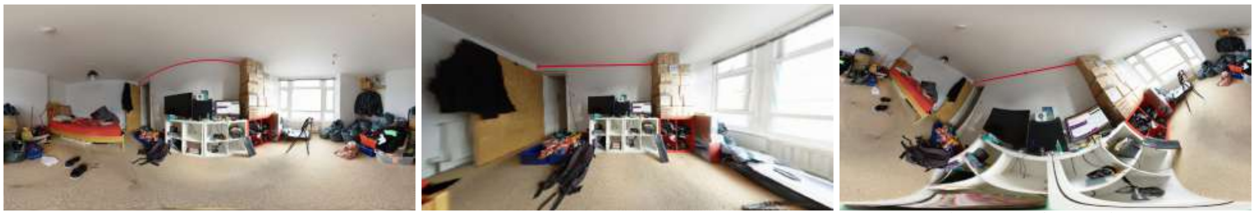
To apply multiple transformations to the same image, we use linear blend skinning [1] with Möbius transformations instead of the usual rotations, translations and scalings in Euclidean space. By using the bounded biharmonic weights [2] as the weighting scheme, we achieve results with different transformations being applied to different points, lines and bidimensional regions on the sphere. In Figure 1 for example, each red line in the image on the left had an associated transformation that rectified it in the image on the right, and they all appear straight in the final result due to the use of these weights. In the same result the user also specifies two spherical cages in white to avoid the transformations to affect areas near the poles, where the higher distortions of the equirectangular format would lead to bad results. Our method is designed to work on the full spherical field of view, while most of the previous methods work with subsets of the sphere.

Our main contribution is a new method that is able to apply Möbius transformations locally only to regions close to intuitive handles. This leads to results that have low conformal distortion, rectified straight lines and can be adjusted in an interface. Due to the bijection between the sphere and the extended complex plane (the stereographic projection), our results are naturally computed over the entire spherical field of view. We also propose a methodology to minimally change conflicting constraints determined by the user (for example, when two handles are set to occupy the same positions in the final result). To the best of our knowledge, no previous method in the context of wide-angle image rectification presents all these features.

The paper is organized as follows: in section 2 we discuss how the previous works are related to ours; in section 3 we present the general idea of our method, review the necessary background and provide details of our method; in section 4 we show how we implemented it; in section 5 we provide results, comparisons with previous methods and discuss limitations; and in section 6 we present directions for future work.

2 Related work

Omnidirectional images represent all the visible information around viewpoints and are well known and appreciated by the large public: the development of matching and stitching techniques in the previous decade [3, 4] made it possible to produce them in devices such as hand-held cameras and cell phones.



(a) Input image with one marked line (b) Perspective projection, FOV = 130° (c) Möbius transformation, FOV = 360°

Figure 2: The user marks one line in the input image to be rectified. The perspective projection is able to rectify this and the other lines, but causes too much stretching when the field of view (FOV) is wider than 90° and is not even defined when the FOV is greater than 180°. The Möbius tranformation uniquely determined by colinearizing three points on the line rectifies it but does not rectify the other lines. On the other hand, it is conformal and defined over the full sphere.

The representation of the visible information as a sphere turns the problem of visualizing it on a planar device into finding a projection from the sphere to the plane, the same problem faced by cartographers through the centuries [5]. Although projections developed by cartographers had to satisfy different properties, both applications shared a common requirement: conformality [6], i.e., the projections had to map circles on the sphere to circles on the plane.

The goal of approximating what the human visual system perceives brings an additional requirement to wide-angle imaging: straight lines in the world have to appear straight in the final result. The well-known perspective projection satisfies this requirement but is not conformal. In fact, there is no projection from the sphere to the plane that maps all possible straight lines to straight lines and all possible circles to circles [7].

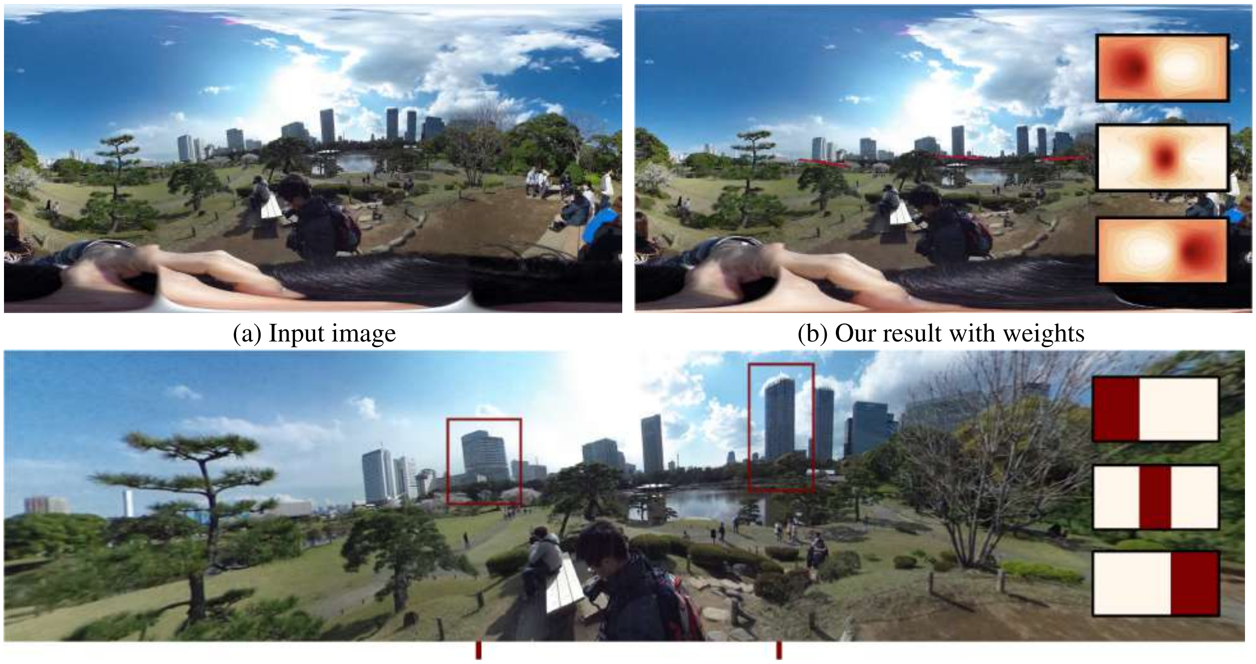
The adoption by Renaissance painters of local perspective projections to avoid the stretching caused by one single perspective inspired the authors of [8] to generate similar depictions in the context of computer generated images. The method in [9] also used narrower field of view projections but to rectify mainly indoor scenes. Both these techniques are limited by how wide the field of view of each projection is. Although we use a combination of projections that resembles the one proposed in [9], our method overcomes this field of view limitation by using Möbius transformations instead of perspective projections.

The method in [10] approached the problem of generating wide-angle images by minimizing energies that quantify deviation from conformality and straightness of lines. This method was able to produce very good results at the cost of having to solve an optimization problem over a very dense mesh. On the other hand, we solve our minimization problem on an adaptive mesh with less vertices.

Another method [11] uses less variables and is able to rectify lines by deforming a cylindrical projection to approximate the perspective projection in specified regions. Although their methodology produces high quality results, the use of a cylinder as the initial domain leads to results that miss regions around the poles of the sphere. Our method produces results with the full spherical field of view and preserves the entire visual information around a given viewpoint. Additionally, our formulation is able to achieve other effects such as local rotations, translations and zoom. Although these effects could be incorporated to their method with additional constraints, this could lead to over-constrained optimization problems, since their formulation already constrains entire regions. Ours uses more localized constraints that lead to more flexibility.

More recent works [12, 13] proposed to apply Möbius transformations to ominidirectional images to achieve different effects. The method in [12] applies scales in the complex plane to map wide fields of view to narrow ones or vice-versa, achieving spherical zooming. The authors of [13] proposed to use specification of three points to achieve zoom, rotation and translation. They also proposed to combine different transformations in the same image, but their ad hoc choice of weights and lack of flexibility to change constraints lead to undesirable distortions when the areas affected by different transformations are too close. Our method, on the other hand, uses weights that are optimally computed to satisfy many different constraints, even when they are very close. We also slightly change constraints to be able to achieve better results.

The methodology of linearly combining local transformations with optimally computed sets of weights is known in the geometry processing literature as linear blend skinning[1]. Some of the best known weighting schemes include harmonic [14], mean value [15] and Green coordinates [16] and an investigation about the use of them in the complex plane was done in [17]. These schemes are mostly determined in terms of bidimensional regions (cages) in the domain and do not have the flexibility we need to deal with points, straight lines and polygons on the sphere.



(a) Input image

(b) Our result with weights

(c) Result by [9] with weights

Figure 3: The unaligned horizon in the input image is corrected by our method through the use of point handles (represented by arrows). Smoothness of weights ensure unnoticeable transitions between the different Möbius transformations. On the other hand, the result of the method in [9] with piecewise constant weights clearly shows the different projections on the buildings (highlighted in red). Ticks on the bottom indicate the vertical transitions between projections.

To achieve this flexibility we opt for using bounded biharmonic weights [2]. These weights guarantee that a transformation determined by one handle does not affect areas close to the other handles and also have the smoothness we need to guarantee non-abrupt transitions in areas affected by different transformations. They were used in other (non-spherical) image processing applications such as transfusive image manipulation [18] and colorization [19], but these applications deal with color processing instead of image warping.

3 Method

We propose a method that satisfies all the following requirements, which are not simultaneously satisfied by any of the previous works:

(i) **Rectification of straight lines:** all user specified lines in the scene should appear with minimal bending in the result.

(ii) **Conformality:** objects should have minimal anisotropic distortion in the final image.

(iii) **Full spherical field of view:** the full 360° by 180° visual information around the viewpoint should be displayed in the result.

(iv) **Interface:** there should be the possibility of editing the result through different effects, such as changing the size, orientation and relative positions of elements in the scene.

The solution that is widely adopted for (i) is to apply perspective transformation to the input equirectangular image. This indeed rectifies all lines in the scene, but introduces conformal distortions as the field of view is widened (Figure 2 (b)), thus not satisfying (ii) and (iii).

On the other hand, *Möbius transformations* [12] satisfy requirements (ii) and (iii): they are the only bijective transformations on the sphere that are conformal [20]. Though not being able to rectify all lines in a scene, the possibility of specifying new positions of three points (see more details in section 3.1) makes it possible to choose one line in the scene to be rectified (Figure 2 (c)). Unfortunately there is no way to guarantee that the other straight lines will be rectified as well and the use of a single Möbius transformation does not satisfy requirement (i).

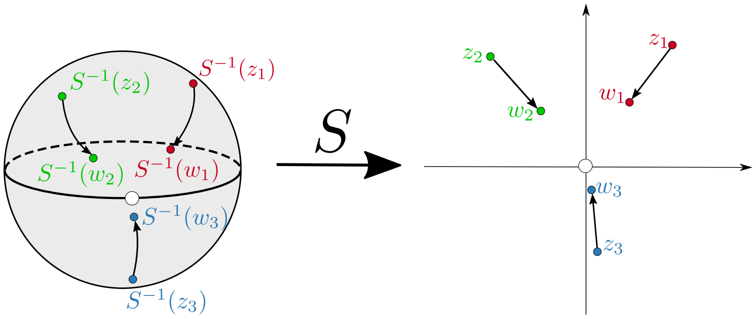


Figure 4: A unique Möbius transformation maps the points z_1, z_2 and z_3 on the complex plane (right) to new positions w_1, w_2 and w_3. Composing this Möbius transformation with the inverse of the stereographic projection defines a transformation on the sphere that maps the corresponding points $S^{-1}(z_1)$, $S^{-1}(z_2)$ and $S^{-1}(z_3)$ to $S^{-1}(w_1)$, $S^{-1}(w_2)$ and $S^{-1}(w_3)$.

To overcome this issue, we propose to adapt a methodology widely used in geometry processing to have different transformations being applied to different areas of a geometric object: linear blend skinning [1]. In our context the object is the sphere $\mathbb{S}^2$ and given a point (λ, ϕ) in the final image, we define its *inverse image* as

$$T^{-1}(\lambda, \phi) = \left(\sum_{j=1}^{m} W_j(\lambda, \phi) \cdot M_j^{-1}(\lambda, \phi) \right), \tag{2}$$

where $\{W_j\}_{j=1}^m$ is a set of weights and $\{M_j\}_{j=1}^m$ is a set of Möbius transformations on the sphere (i.e., the composition of the stereographic projection, with a Möbius transformation on the extended complex plane and the inverse of the stereographic projection). If the weights are local, i.e., if they have higher values only in regions that they are supposed to affect the most, we can overcome the limitation of the Möbius transformations rectifying only a single line (Figure 2 (c)) and combine them to rectify multiple lines (Figure 1, right), thus satisfying requirement (i). Notice also that (2) is defined for every (λ, ϕ) on the sphere, making our method able to deal with the full spherical field of view (requirement (iii)).

Möbius transformations also produce different effects such as rotations, spherical zoom (scaling) and translations. To explore this property and have a more complete interface (iv), we need to specify other handles such as points, which are more natural to specify over translations and rotations, and bidimensional regions, which is more natural over scaling.

Our definition of the inverse mapping (2) uses the weights $W_j(\lambda, \phi)$ in terms of the final positions (λ, ϕ). To properly set the constraints on the weights, we have to know in advance where the handles in the input image will be mapped to in the final image. The knowledge of what Möbius transformation will be applied to each handle will make it possible to know where the handles are mapped to. We elaborate on this point in section 3.3.

A crucial property the weights must have is smoothness: non-smooth weights can cause unnatural transitions in areas where the different transformations are being applied. The extreme case is the use of weights that assume only values 0 or 1, which has been exploited previously by the method in [9] (they apply perspective transformations instead of Möbius). Although they can produce good results in scenes where the discontinuities can be well hidden, the transitions are noticeable in most cases (Figure 3 (c)). The use of smooth weights solves this problem (Figure 3 (b)).

To summarize, the weights have to be local, smooth and work for different types of handles: *bounded biharmonic weights* [2], for which we present additional details in section 3.2, completely satisfy all these requirements. Combining them with Möbius transformations allows us to propose a method that satisfies requirements (i) through (iv) stated in the beginning of this section.

3.1 Möbius Transformations

Since the stereographic projection establishes a bijection between the unitary sphere $\mathbb{S}^2$ and the extended complex plane $\mathbb{C} \cup \{\infty\}$, complex transformations arise as a possibility to transform spherical images. Figure 4 illustrates this possibility: a transformation is applied to the complex plane (on the right) and its composition with the inverse of the stereographic projection defines a transformation on the sphere (on the left). When the mapping on the right is a Möbius transformation, we say that this composition is a Möbius transformation on the sphere. Möbius transformations constitute the set of conformal transformations of the unitary sphere into itself, meaning that these transformations

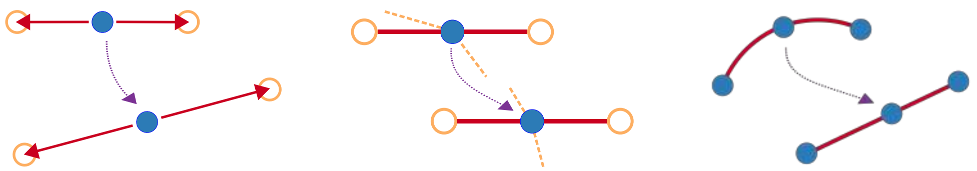


Figure 5: Left: point handle transformation. Center: curved cage handle vertex transformation. Right: curved bone handle transformation. This figure illustrates how each associated Möbius transformation is obtained by position specification. Filled circle represents control point, empty circles represent auxiliary control points. Dashed lines represent curved cage segments.

preserve angles, locally rotating and scaling by the same amount along all directions, therefore making them appropriate to shape preservation. In the extended complex plane, the Möbius transformations are defined by Equation (1).

Another noteworthy property of Möbius transformations is the possibility to define a unique transformation given three initial points and their respective final positions, enabling an intuitive way to define the transformation associated to each handle. This property allows a bigger set of transformations to be used without requiring previous knowledge, therefore satisfying the desired property (iv). The following theorem is crucial for the development of our method:

Theorem 1. *Given distinct* $z_1, z_2, z_3 \in \mathbb{C}$ *and distinct* $w_1, w_2, w_3 \in \mathbb{C}$ *there is a unique Möbius Transformation* $M(z) = \frac{az+b}{cz+d}$*, such that* $w_i = M(z_i)$ *for* $i = 1, 2, 3$*. Its coefficients are given by:*

$$a = det\begin{pmatrix} w_1z_1 & w_1 & 1 \\ w_2z_2 & w_2 & 1 \\ w_3z_3 & w_3 & 1 \end{pmatrix}, \quad b = det\begin{pmatrix} w_1z_1 & z_1 & w_1 \\ w_2z_2 & z_2 & w_2 \\ w_3z_3 & z_3 & w_3 \end{pmatrix},$$

$$c = det\begin{pmatrix} z_1 & w_1 & 1 \\ z_2 & w_2 & 1 \\ z_3 & w_3 & 1 \end{pmatrix}, \quad d = det\begin{pmatrix} w_1z_1 & z_1 & 1 \\ w_2z_2 & z_2 & 1 \\ w_3z_3 & z_3 & 1 \end{pmatrix}.$$

Proof. The proof can be found in [13]. □

This theorem allows us to define Möbius transformations on the sphere that map three points (λ_1, ϕ_1), (λ_2, ϕ_2) and (λ_3, ϕ_3) to new positions (Λ_1, Φ_1), (Λ_2, Φ_2) and (Λ_3, Φ_3). Let S be the stereographic projection, $z_1 = S(\lambda_1, \phi_1)$, $z_2 = S(\lambda_2, \phi_2)$, $z_3 = S(\lambda_3, \phi_3)$, $w_1 = S(\Lambda_1, \Phi_1)$, $w_2 = S(\Lambda_2, \Phi_2)$, and $w_3 = S(\Lambda_3, \Phi_3)$ (Figure 4, right). Using Theorem 1, we calculate the coefficients a, b, c and d that define the Möbius transformation $M(z) = \frac{az+b}{cz+d}$. The composition $S^{-1} \circ M \circ S$ is the Möbius transformation on the sphere that maps (λ_1, ϕ_1), (λ_2, ϕ_2) and (λ_3, ϕ_3) to (Λ_1, Φ_1), (Λ_2, Φ_2) and (Λ_3, Φ_3). The specification of these points is done using geometric handles (points, bones and cages) and is discussed in Section 3.3.

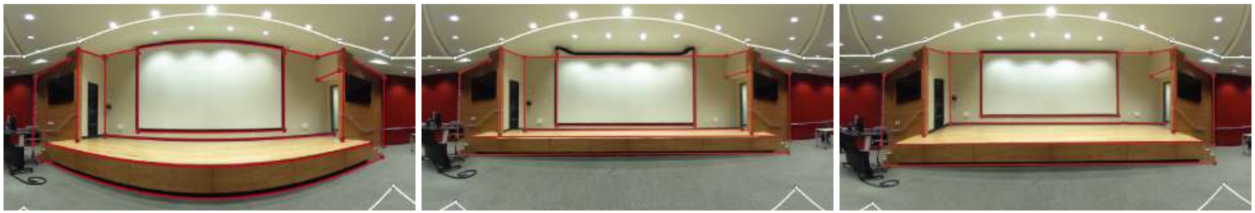
3.2 Bounded Biharmonic Weights

As shown in Figure 2 (c), a Möbius transformation affects the entire omnidirectional image, leading to unexpected distortions far from the region of interest. We thus associate to every Möbius transformation a set of weights to ensure the locality of the mapping. Bounded Biharmonic Weights (BBW) were formulated in [2] as a method for linear blend skinning (LBS) of 2D and 3D shapes. This method has the advantage of offering support to all common types of handles: points, bones and cages.

BBW are obtained by minimizing the Laplacian energy subject to constraints of non-negativity, range between 0 and 1, and global maxima over the handles. Since the minimization of the Laplacian Energy is equivalent to solving the related Euler-Lagrange equations ($\Delta^2 W_j = 0$), the BBW are $\mathcal{C}^1$ over the handles and $\mathcal{C}^\infty$ everywhere else, provided that the boundary conditions are sufficiently smooth; they thus satisfy the desired smoothness property.

The constraint of global maxima combined with non-negativity allows us to achieve locality of transformations, with points in the longitude/latitude domain being affected only by the closest handles. With this property, the number of possible rectified lines is greatly improved, thus satisfying property (i).

Finally, the implementation of BBW allows the use of the most common types of handles since they appear as boundary conditions in the minimization problem. In the original work, point handles are defined as points in the shape domain



(a) Cropped input image with handles

(b) Without relative position preservation

(c) With relative position preservation

Figure 6: Three very close handles are specified in the input image: the top of the screen and the left and right sides of the screen. Moving the middle point of the top of the screen downwards to rectify it makes it cross the other handles (b). We optimally calculate new positions for the handles to satisfy the user requirements as closely as possible while preserving their relative positions (c).

whose weight values must be 1, while bone handles requires that the entire shortest Euclidean segment connecting two points must have weight equal to 1. Cage handles behave as point handles except that they require to have weights varying linearly over the faces. This methodology increases flexibility and allows more intuitive transformation specification, therefore satisfying the property (iv). In our work, we redefine these boundary conditions to work better in the spherical domain.

3.3 Handles and Associated Transformations

Since the geometric object we are dealing with is a sphere, we have to adapt the Euclidean versions of handles (points, bones and cages) and transformations to work in this setting.

Point handles allow us to (locally) translate, scale or rotate regions of the sphere. To each point handle in the input equirectangular image we associate two auxiliary points on the left and on the right of it in the equirectangular domain (Figure 5, left). With these three points (called control points of the point handle) and their new positions in the final image we are able to define a unique Möbius transformation, according to what was explained in Section 3.1.

If the central point is moved by the user, the new positions of the other two points follow the same displacement in the equirectangular domain, leading to a local translation effect. The input/output positions of these three points are then mapped to the sphere and to the complex plane via stereographic projection, where the coefficients of the corresponding Möbius transformation is computed (see Theorem 1).

The auxiliary points can also be moved in the interface. They are represented by arrows emanating from the central point and, when this arrow is moved, the other arrow moves by the same amount in the opposite direction, while the central point is kept at the same place. The corresponding transformation rotates and/or scales the three points and areas where the associated weights have higher values.

We define spherical cages to be the curved analogue of cages: their vertices work as point handles where only the central point can move (Figure 5, center) and weights vary linearly over the spherical segments. The two auxiliary points move accordingly, just as in the point handle case, leading to a local translation effect. Cage handles are useful to restrict the scope of the transformations associated to the other handles since over the cage segments the only acting transformations are the ones defined by the corresponding vertices (Figure 10). These handles are also useful to modify an entire region of the image.

We also define a spherical bone as the analogue of the bone in the spherical metric, consisting of the geodesic connecting two points on the sphere. Given two points $P_1, P_3 \in \mathbb{S}^2$ its mathematical expression is the following:

$$\begin{array}{rcl} SB : [0,1] & \to & \mathbb{S}^2 \\ t & \mapsto & \frac{(1-t)P_1 + tP_3}{\|(1-t)P_1 + tP_3\|} \end{array} \tag{3}$$

Let $P_2 = SB(\frac{1}{2}) \in \mathbb{S}^2$ be the middle point of this geodesic. P_1, P_2 and P_3 and their final positions define a unique Möbius transformation to be applied to the bone.

These three points, which we call control points of the bone handle, can be moved freely or in a way to colinearize them in the final result. In the former case, the new positions are directly specified by the user. In the latter, the points at the end of the segment are fixed, while the middle point is mapped to the middle position on the straight line segment

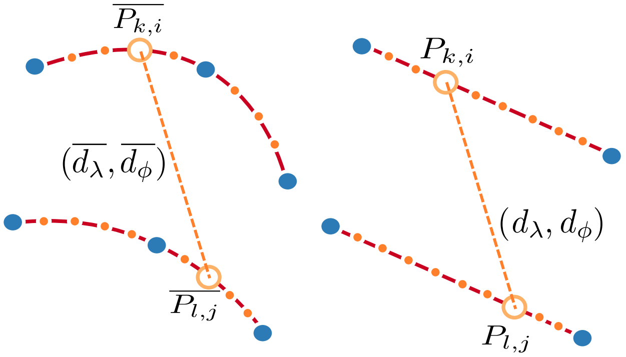


Figure 7: We propose an energy term to preserve relative distances between points on the input handles (left) and on the output rectified handles (right). This term is formulated to preserve distances on both λ and ϕ coordinates, i.e., $d\lambda \approx \overline{d\lambda}$ and $d\phi \approx \overline{d\phi}$.

(Figure 5, right). In other words, the Möbius rectification transformation is determined by

$$M : [P_1, P_2, P_3] \mapsto [P_1, \frac{1}{2}P_1 + \frac{1}{2}P_3, P_3]. \tag{4}$$

3.4 Optimal handle positions

To help the user achieve line rectification we make available in our interface a button that rectifies all lines at once. When dealing with multiple lines, the mappings defined in (4) can lead to artifacts in the result, since considering each handle individually can cause the handles to intersect.

Figure 6 (b) illustrates this problem. Trying to rectify the horizontal line of the top of the screen using (4) maps its middle control point downwards, because it colinearizes the three points with the two endpoints fixed. On the other hand, the endpoints of the vertical segments stay fixed, leading to an artificial intersection. One possibility would be to show an error message and ask the user to redefine the handles when the problem is detected, but this would be very restricting, since these cases happen often.

To avoid this problem, we propose to optimize the positions of the bone handles in the final image to preserve their relative positions in the input image. After this optimization is performed and new control points positions are found for the handles, we associate to each handle the transformation in (4), replacing the final positions of P_1 and P_3 and the middle point of the bone by the positions obtained by the optimization. Figure 6 (c) shows that the result of this methodology (to be described in the rest of this section) is effective at avoiding the intersection in (b).

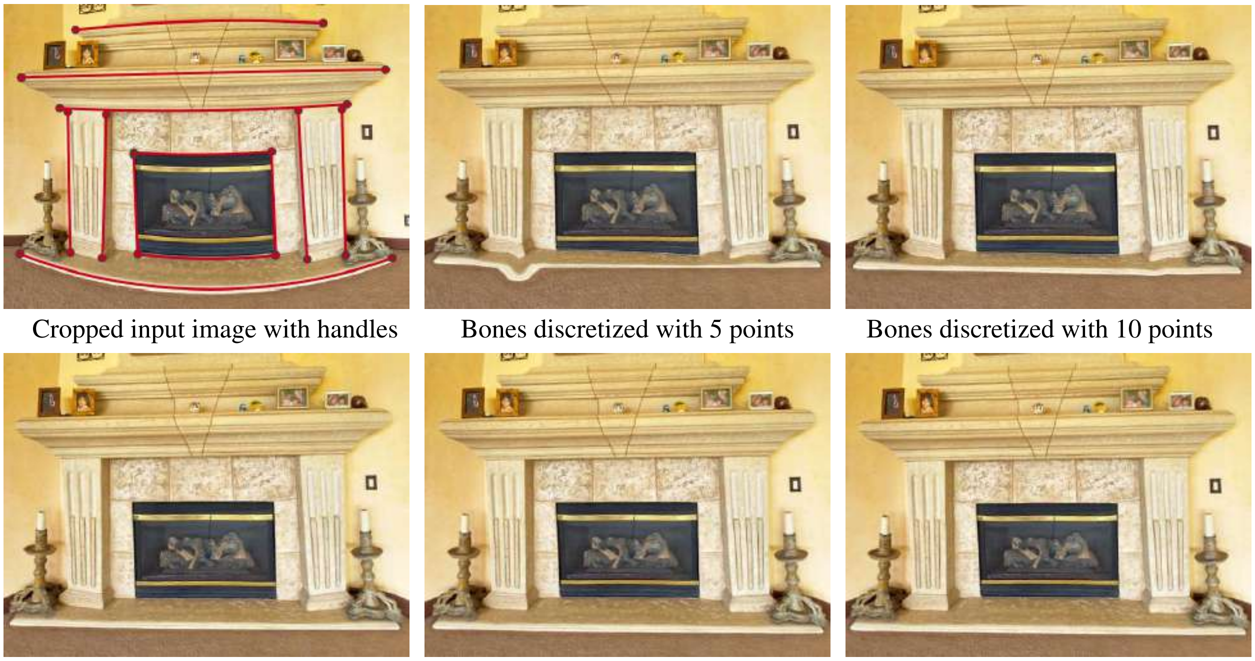
Let the original positions of the endpoints of the k-th handle be $\overline{P_{1,k}} = \left(\overline{P^\lambda_{1,k}}, \overline{P^\phi_{1,k}}\right)$ and $\overline{P_{3,k}} = \left(\overline{P^\lambda_{3,k}}, \overline{P^\phi_{3,k}}\right)$, where the superscripts λ and ϕ represent the longitude and latitude coordinates in the equirectangular domain. We sample 50 points equally spaced in the parameter domain of the spherical bone connecting $\overline{P_{1,k}}$ to $\overline{P_{3,k}}$ (given by equation (3)). We denote these points by $\overline{P_{k,i}}$ and their respective parameters by $t_{k,i}$ $(i = 1, \ldots, 50)$. Given the l-th handle $(l \neq k)$ with respective samples $\overline{P_{l,j}}$ and parameter values $t_{l,j}$ $(j = 1, \ldots, 50)$, the segment connecting the output endpoints of the k-th handle and the output endpoints of the l-th handle will preserve relative positions if

$$\begin{aligned}\left(P^\lambda_{1,k} \cdot (1 - t_{k,i}) + P^\lambda_{3,k} \cdot t_{k,i}\right) - \left(P^\lambda_{1,l} \cdot (1 - t_{l,j}) + P^\lambda_{3,l} \cdot t_{l,j}\right)\\ \approx \left(\overline{P^\lambda_{k,i}} - \overline{P^\lambda_{l,j}}\right)\end{aligned} \tag{5}$$

and

$$\begin{aligned}\left(P^\phi_{1,k} \cdot (1 - t_{k,i}) + P^\phi_{3,k} \cdot t_{k,i}\right) - \left(P^\phi_{1,l} \cdot (1 - t_{l,j}) + P^\phi_{3,l} \cdot t_{l,j}\right)\\ \approx \left(\overline{P^\phi_{k,i}} - \overline{P^\phi_{l,j}}\right),\end{aligned} \tag{6}$$

where $(P^\lambda_{1,k}, P^\phi_{1,k})$ and $(P^\lambda_{3,k}, P^\phi_{3,k})$ are the final optimal positions of the endpoints of the k-th handle and $(P^\lambda_{1,l}, P^\phi_{1,l})$ and $(P^\lambda_{3,l}, P^\phi_{3,l})$ are the final optimal positions of the endpoints of the l-th handle. Figure 7 illustrates the desired behavior described in (5) and (6).



Cropped input image with handles

Bones discretized with 5 points

Bones discretized with 10 points

Bones discretized with 25 points

Bones discretized with 50 points

Bones discretized with 100 points

Figure 8: For the input image with close bone handles (top-left), we show the final results of our method using 5, 10, 25, 50 and 100 points to discretize the bones in the relative position energy.

In a least squares sense, these conditions can be reformulated as the minimization of the following energy:

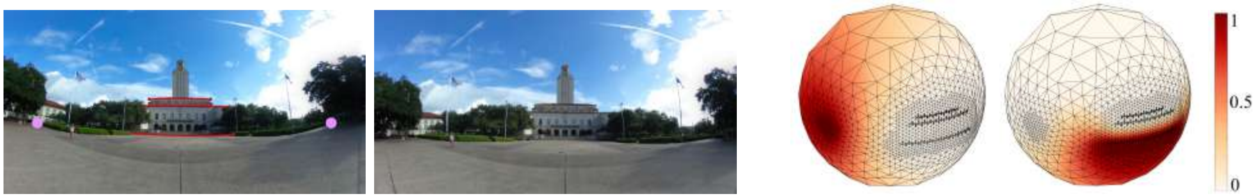
$$
\begin{aligned}
E_{rp,i,j,k,l} = w_{rp}(i,j,k,l)\Big[& \Big(P^{\lambda}_{1,k} \cdot (1 - t_{k,i}) + P^{\lambda}_{3,k} \cdot t_{k,i}\Big) - \\
& - \Big(P^{\lambda}_{1,l} \cdot (1 - t_{l,j}) + P^{\lambda}_{3,l} \cdot t_{l,j}\Big) - \\
& - \Big(\overline{P^{\lambda}_{k,i}} - \overline{P^{\lambda}_{l,j}}\Big) \Big]^2 + \\
+ w_{rp}(i,j,k,l)\Big[& \Big(P^{\phi}_{1,k} \cdot (1 - t_{k,i}) + P^{\phi}_{3,k} \cdot t_{k,i}\Big) - \\
& - \Big(P^{\phi}_{1,l} \cdot (1 - t_{l,j}) + P^{\phi}_{3,l} \cdot t_{l,j}\Big) - \\
& - \Big(\overline{P^{\phi}_{k,i}} - \overline{P^{\phi}_{l,j}}\Big) \Big]^2 .
\end{aligned}
\tag{7}
$$

The weights are given by

$$
w_{rp}(i,j,k,l) = \begin{cases} \frac{1}{20}(1 - 4r/S)^4, & \text{if } 0 \le r \le S/4, \\ 0, & \text{otherwise} \end{cases} \tag{8}
$$

where $r = r(i,j,k,l)$ is the Euclidean distance between $\overline{P_{k,i}}$ and $\overline{P_{l,j}}$ and $S = S(k)$ is the Euclidean distance between $\overline{P_{1,k}}$ and $\overline{P_{3,k}}$. The goal of these weights is to prioritize relative position preservation for closer points.

We opted for 50 points to discretize each bone because the energy term in eq. (8) decays fast with respect to distance and we needed a number of points high enough to guarantee coverage of bones of arbitrary size and orientation. Figure 8 presents different results varying the number of points to discretize each handle. With 5 points the energy does not prevent crossing of the bones. With 10 points this problem is prevented but the weak relative position preservation leads to distortions. Results with 25, 50 and 100 have less distortions and small improvements as the number of points grows. We chose the default to be 50 due to the timing to set up the final linear system: in this example, using 25 points took around 0.07 seconds, 50 points took 0.11 seconds and 100 points took 0.26 seconds (please refer to Section 5 for more timing details of the whole method). On the other hand, adjusting eq. (8) to decay slower would spoil the locality of the energy, affecting handles that are too far. Finally, the 4 dividing the distance between endpoints in (8) is a good compromise to avoid overlapping of handles without affecting handles that are too far.



(a) Cropped input with handles

(b) Cropped output

(c) Weights associated to a point handle

(d) Weights associated to a bone handle

Figure 9: The user defines two point handles (magenta) and three bone handles (red) over the input image (a). Point handles are rotated and translated and bone handles are rectified, obtaining the result in (b). The adaptive mesh is denser in areas close to the handles (c,d).

The minimization of E_{rp} can find new positions that are far from the original positions and do not rectify the handles. To ensure bone rectification, we associate to the i-th handle the energy

$$E_{br,i} = 100 \left\| P_{2,i} - \left(\frac{1}{2} P_{1,i} + \frac{1}{2} P_{3,i} \right) \right\|^2 . \tag{9}$$

This energy term tries to set the new position of the middle point of the respective bone to be colinear with the endpoints, according to (4).

To preserve the location of the handles in the final result, each point that is not a middle point of a bone handle is enforced to keep its initial position. We then add an energy term related to the j-th control point of the i-th handle:

$$E_{po,i,j} = 0.3 \| P_{i,j} - \overline{P_{i,j}} \|^2 . \tag{10}$$

Finally, due to joints in bone handles, we also add an energy term to preserve the relative position of control points that are part of the m-th joint, given by

$$E_{jo,m} = 1000 \sum_{j} \sum_{k>j} \| P_j - P_k - (\overline{P_j} - \overline{P_k}) \|^2 . \tag{11}$$

We define the total energy as

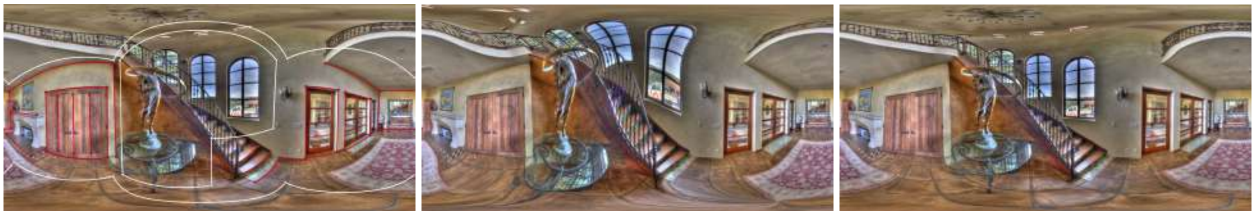
$$E = \sum_{i} E_{br,i} + \sum_{k,l} \sum_{j,i} E_{rp,k,l,j,i} + \sum_{m} E_{jo,m} + \sum_{i,j} E_{po,i,j} . \tag{12}$$

The fixed parameters in equations (8), (9), (10) and (11) were used to generate all results in this paper. Although they were empirically determined, we can discuss their relative importances. The energy term E_{jo} (eq. (11)) has the highest weight (1000) to guarantee that bone joints do not split, what would cause tears in the final image. The next highest parameter is 100 in E_{br} (eq. (9)) and it enforces rectification of bones, a fundamental feature of our method. The term E_{po} (eq. (10)) has a lower weight (0.3) because assigning it a higher weight would make it conflict with the other energies. Preserving the original position of the handles in the final result is less important than rectifying bones, preserving relative distances between handles and guaranteeing that bone joints do not separate. The parameter $\frac{1}{20}$ in w_{rp} (eq. (8)) is lower but it is summed over the many discretization points, leading to the desired relative distance preservation.

Since every energy term is a quadratic function with respect to control point positions, we can minimize it by solving the linear system given by $\frac{\partial E}{\partial P_{i,j}} = 0$. The solution of this linear system provides the final positions of each control point. Then the respective Möbius transformations are calculated and linearly blended into the result (Figure 6, (c)).

3.5 Adaptive Mesh

To calculate the Bounded Biharmonic Weights (Section 3.2) we have to discretize the sphere. To improve efficiency, we propose an adaptive mesh that is more refined near handles, where the weights are likely to vary more, and contains vertices that belong to the handles, over which the boundary conditions are imposed ($W_k = 1$ over the k-th handle, $W_k = 0$ over the others and weights varying linearly over cage edges).



(a) Input with marked lines and cages (b) Result using only lines (c) Result using lines and cages

Figure 10: The user marks in red lines to be rectified and in white spherical cages to prevent the points inside it to move. Using only the specified lines rectifies them but distorts the windows, central sculpture and table. Using lines and cages keeps lines straight and does not distort the objects inside the cages.

The first step for the mesh generation is to define an icosahedral mesh, which is a coarse discretization of the sphere with well shaped triangles. We then calculate the minimum distance from the barycenter of each triangle of this icosahedron to the set of handles. Triangles with barycenter distance smaller than a given threshold are subdivided into four triangles using the usual 1-to-4 split operator and the newly generated vertices (edge midpoints) are projected to the sphere. This process is repeated five times for the resulting refined meshes, with the threshold being decreased at every round. This guarantees that the closer a region of the sphere is to the set of handles, the denser the mesh is over that region.

We add to these mesh vertices that belong to the transformed handles. It is important to emphasize that we calculate inverse warpings (see equation (2)), so the *final* position of the handles are the points over which the boundary conditions are defined. We add to the mesh 50 points that result from sampling each bone and each cage edge. Point handles are added to the mesh as well. We then calculate the convex hull of the vertices of the previous refined mesh and these new vertices to obtain a triangle mesh.

Figure 9 shows an example of adaptive mesh generated by our method. The user marks over the input image (a) three bone handles and two point handles. Only a part of the input is shown, since the missing parts of the images do not contain any handle. The user then sets the bone handles to be rectified and the point handles to be translated and rotated in order get a better alignment of the image. The result is shown in (b). The adaptive mesh is shown in (c) and (d) along with weights associated to a point handle (c) and a bone handle (d). We can observe that the region around the north pole has a very coarse discretization, since this region has no handles. On the other hand, areas near the handles are more refined.

4 Implementation

Our methodology was implemented in MATLAB and the interface was developed with the GUIDE Toolbox. The program is divided in two windows (see an example in the accompanying video).

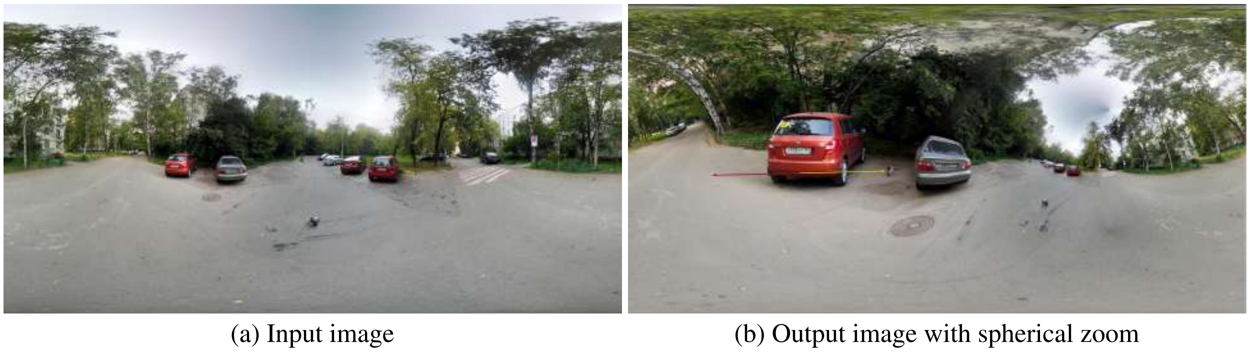
The first window is where the user defines the handles that will act as boundary conditions in the BBW computation (section 3.3). To deal with bone handle intersections we simply break them at the point of intersection, generating four bones (for each intersection) instead of two. We also use a clustering algorithm [21] to couple spherical bones if the endpoints are below some distance threshold.

In the second window, the user defines the Möbius transformation associated to each handle by moving the control points over the image. We also provide a button that rectifies all lines at once, which solves the linear system discussed in section 3.4. If it is not desired to rectify all of them, we provide another button that rectifies only the segment selected by the user.

With the boundary conditions defined as described in Section 3.3, we compute the BBW using its Matlab implementation [22] integrated with Mosek [23] to perform the optimization. As mentioned in [2], dropping the partition of unity condition does not perceptually affect the generated results. We then disable this option making the computation run significantly faster, with time being reduced between 70% and 95% in our tests.

The linear blending is then performed over the sphere using the Möbius transformations generated by control point displacement and the result of our method replaces the input image in the interface.

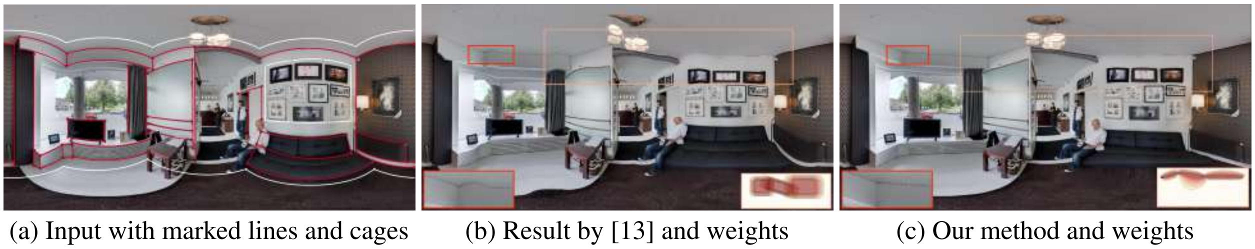
The source code, a video showing an example of our user interface and a complete description of all the functionalities of our interface are made available in the supplementary material accompanying this work. In case of acceptance of this paper, the source code will be publicly available in an online repository.



(a) Input image

(b) Output image with spherical zoom

Figure 11: A point handle performs zoom in the license plate making the letters and numbers on it more clear. Applying common zoom to achieve the same task would miss part of the visible information of the scene, while spherical zoom maintains all the information in the image.



(a) Input with marked lines and cages

(b) Result by [13] and weights

(c) Our method and weights

Figure 12: The weights used in [13] affect overlapping regions, leading to artifacts on lines on the ceiling (zoomed in insets on the bottom left) and also distorting the man in the center. We calculate weights solving a constrained optimization problem that ensure their locality and avoid these distortions.

5 Results

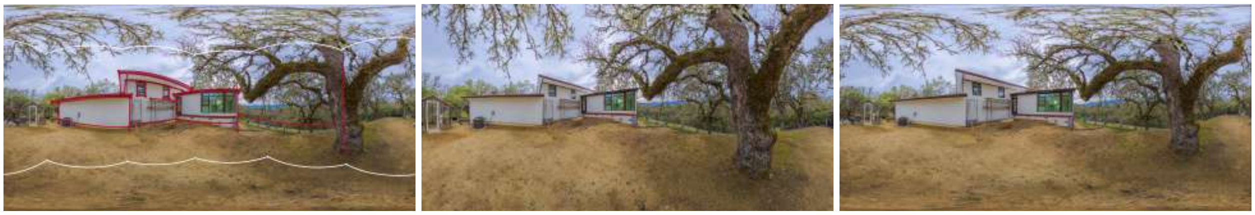
In this section we present a collection of results illustrating the different possibilities of our method and provide comparisons to current state-of-the art algorithms to correct distortions in wide-angle images. A video with additional results and full resolution versions of all input and output images in this paper can be found in the supplementary materials. All our results were generated using the full spherical field of view, while other methods such as [9], [10] and [11] have field of view limitations.

The user can constrain the transformations more and impose that some regions do not move at all when equation (2) is applied, i.e., regions where the final transformation should be the identity transformation. This is achieved by enclosing the regions by spherical cages and not moving their vertices in the final result. In Figure 10 the generation of a result using only the specified lines in (a) leads to distortions in the windows and in the central sculpture (b). On the other hand, the result using lines and cages specified in (a) preserves these regions (c). This feature was also used in Figures 1, 12 and 13.

Figure 11 illustrates how to achieve spherical zoom in an equirectangular image: the user marks a point handle in the area they want to zoom in, and scales its two arrows. In this example, the letters and numbers on the license plate became much more clear after the zoom. Since the associated Möbius transformation is a bijection on the sphere, no information in the scene is lost performing this operation. This is particularly useful for surveillance applications.

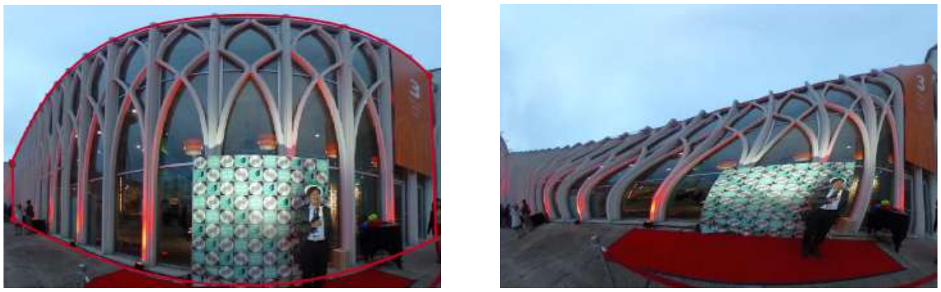
We present in Figure 12 a comparison that shows the effectiveness of the bounded biharmonic weights to deal with very close handles. The weights proposed by [13] affect overlapping regions (inset of Figure 12 (b)) and leads to artifacts when combined with the corresponding Möbius transformations. BBW (Figure 12 (c) and inset) are more local and lead to a better result.

We provide in Figure 13 a comparison to the method proposed in [10]. We first observe that none of the results in their paper use the full spherical field of view, while our method is designed specifically for this task. Using their method in this context leads to irregular borders and the commonly employed cropping leads to loss of information (Figure 13 (b) and supplementary materials for the uncropped version). Our method (c) also rectifies all the specified lines (a), produces a result with regular borders (it has the same size of the input image). In addition, our algorithm has



(a) Input with lines and cages (b) Result by [10] (cropped) (c) Our result

Figure 13: The user specifies lines to be rectified in the scene (a). The method in [10] (b) does rectify all these lines, but leads to irregular borders and loss of content if the result is cropped to produce a rectangular output. Our result (c) has regular borders and the same resolution as the input image. In this example, the same lines were provided to both methods.



(a) Input with marked lines (b) Result (limitation)

Figure 14: Our method is not able to rectify lines that occupy a too wide field of view, such as the line on the ceiling of the building in the input image (a). It also depends on complete specification of lines, making the missed lines such as the vertical ones of the building distorted.

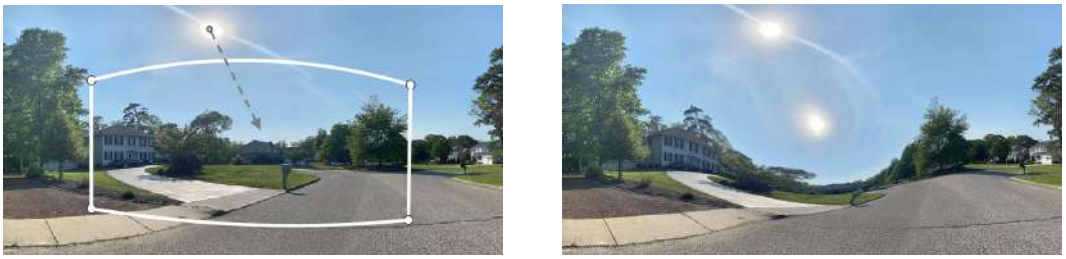
other features such as local zoom (Figure 11), local translations and rotations (Figures 3 and 9) and cage specification (Figure 10).

Our method takes in average 0.4 seconds per handle on an Intel Core i5 @ 2.3GHz Quad-Core Macbook with 8 GB RAM. The optimization to find the weights and the interpolation of these weights from the mesh resolution to the final image resolution are the most time consuming parts of our code, the former taking approximately 70% of the total computation time and the latter 23%. This is due to less efficient implementations of these tools in Matlab. Our method would highly benefit from a C++ implementation with GPU texture interpolation, and we plan to approach this task as a future work. The standard resolution of the output images (2048×1024) makes the weight interpolation step slower. Other less time consuming parts of our code include optimal handle positions computation (section 3.4, with approximately 3%), adaptive mesh construction (section 3.5, $\approx 3\%$) and final image calculation (eq. (2), $\approx 1\%$). The computation of the coefficients a, b, c and d from Theorem 1 to determine each handle transformation takes negligible time (less than 0.1%).

Limitations: Figure 14 shows a limitation of our method: its inability to rectify too wide lines. This is because there is no guarantee that making three points colinear with a Möbius transformation rectifies the whole line. This limitation was also faced by previous approaches; for example, see Figure 7 in [11] and Figure 9 in [10]. We think that the solution to this problem may be in higher dimensional complex space where more constraints can be prescribed.

Additionally, our method depends on precise specification of lines and missing them will likely cause them to be bent in the final result (vertical lines of the building in Figure 14). This was also a limitation of previous methods [10, 11].

Although a single Möbius transformation is a bijection on the sphere, the blending of multiple Möbius transformations (eq. (2)) is not guaranteed to be. Figure 15 shows an example where the user tries to move a point handle to inside a fixed cage, leading to conflicting constraints. On one hand, the final color of the point inside the cage is given by the input color of the point handle. On the other hand, points outside the cage are affected by the identity transformation and stay fixed. This leads to a duplication of the sun in the final image. We emphasize that this is an artificial user specification, and we have not seen this behavior in practical examples. We also mention that the lack of bijectivity also happens to other methods: in [10] the irregular borders make parts of the spherical field of view to be missed when



(a) Input with handles

(b) Result

Figure 15: The user maps the point handle to inside the cage, while maintaining the cage fixed. This (unusual) conflicting specification lead to a duplication of the sun.

cropped to a rectangular format (Figure 13) . The method in [11] misses areas near the poles and their final result is not a bijection on the sphere.

6 Conclusion and Future Work

In this work we presented a method to correct conflicting geometric distortions in the challenging scenario of fully spherical images. The use of Möbius transformations defined by handles in the images and the blending of them with bounded biharmonic weights showed to be effective and generated results with low conformal distortion, rectified lines and re-positioning, re-scaling and re-orientation of objects in the scene.

The popularization of omnidirectional video presents promising directions for future work. Spherical video warpings represent an even more challenging problem due to additional temporal coherence constraints and only a few works have investigated them [24, 25].

A related hot topic is the visualization and interaction with spherical videos in head-mounted displays [26] and we plan to exploit Möbius transformations to achieve real-time zoom, rotations and translations in this virtual reality context. We believe that other more sophisticated effects can be proposed within the same framework.

Acknowledgments

The first author thanks CNPq and CAPES and the second author thanks CNPq for the financial support. The authors also thank Sarah Kushner for a draft review and the following Flickr users, who made available their equirectangular images: Dominic Alves, Kuruman, Moody College of Communication, Bob Dass, Artem Svetlov, Janne-Alias, Dave Jimison and Georgie Dortone.

References

[1] A. Jacobson, Z. Deng, L. Kavan, and J. P. Lewis, "Skinning: real-time shape deformation," in *ACM SIGGRAPH 2014 Courses on - SIGGRAPH '14*, (New York, New York, USA), pp. 1–1, ACM Press, 2014.

[2] A. Jacobson, I. Baran, J. Popović, and O. Sorkine, "Bounded biharmonic weights for real-time deformation," *ACM Transactions on Graphics*, vol. 30, p. 1, 7 2011.

[3] R. Szeliski, "Image Alignment and Stitching: A Tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.

[4] M. Brown and D. G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features," *International Journal of Computer Vision*, vol. 74, pp. 59–73, 4 2007.

[5] J. Snyder, *Map projections: A working manual*. Washington, D.C.: U.S. Government Printing Office, 1987.

[6] L. Sacht, *Content-based projections for panoramic images and videos*. PhD thesis, IMPA, 2010.

[7] D. Zorin, *Correction of geometric perceptual distortions in pictures*. PhD thesis, California Institute of Technology, 1995.

[8] M. Agrawala, D. Zorin, and T. Munzner, "Artistic Multiprojection Rendering," in *Rendering Techniques 2000*, pp. 125–136, Eurographics, 2000.

[9] L. Zelnik-Manor, G. Peters, and P. Perona, “Squaring the circle in panoramas,” in *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, pp. 1292–1299, IEEE, 2005.

[10] R. Carroll, M. Agrawal, and A. Agarwala, “Optimizing content-preserving projections for wide-angle images,” *ACM Transactions on Graphics*, vol. 28, p. 1, 7 2009.

[11] J. Kopf, D. Lischinski, O. Deussen, D. Cohen-Or, and M. Cohen, “Locally Adapted Projections to Reduce Panorama Distortions,” *Computer Graphics Forum*, vol. 28, pp. 1083–1089, 6 2009.

[12] L. Peñaranda, L. Velho, and L. Sacht, “Real-time correction of panoramic images using hyperbolic Möbius transformations,” *Journal of Real-Time Image Processing*, vol. 15, pp. 725–738, 5 2015.

[13] L. Souto Ferreira, L. Sacht, and L. Velho, “Local Moebius transformations applied to omnidirectional images,” *Computers & Graphics*, vol. 68, pp. 77–83, 11 2017.

[14] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki, “Harmonic coordinates for character articulation,” *ACM Transactions on Graphics*, vol. 26, p. 71, 7 2007.

[15] Y. Lipman, J. Kopf, D. Cohen-Or, and D. Levin, “GPU-assisted Positive Mean Value Coordinates for Mesh Deformations,” in *SGP07: Eurographics Symposium on Geometry Processing* (A. Belyaev and G. Michael, eds.), pp. 117–123, 2007.

[16] Y. Lipman, D. Levin, and D. Cohen-Or, “Green Coordinates,” in *ACM SIGGRAPH 2008 papers on - SIGGRAPH ’08*, (New York, New York, USA), p. 1, ACM Press, 2008.

[17] O. Weber, M. Ben-Chen, C. Gotsman, and K. Hormann, “A Complex View of Barycentric Mappings,” *Computer Graphics Forum*, vol. 30, pp. 1533–1542, 8 2011.

[18] K. Yücer, A. Jacobson, A. Hornung, and O. Sorkine, “Transfusive image manipulation,” *ACM Transactions on Graphics*, vol. 31, p. 1, 11 2012.

[19] A. Jacobson and O. Sorkine, “A cotangent Laplacian for Images as Surfaces,” tech. rep., ETH Zurich, Zurich, 2012.

[20] J. Olsen, “The Geometry of Möbius Transformations,” 2010.

[21] Y. Marcon, “clusterXYpoints,” 2016.

[22] A. Jacobson and others, “{gptoolbox}: Geometry Processing Toolbox,” 2018.

[23] “Mosek ApS,” 2020.

[24] L. Sacht, L. Velho, D. Nehab, and M. Cicconet, “Scalable motion-aware panoramic videos,” in *SIGGRAPH Asia 2011 Sketches on - SA ’11*, (New York, New York, USA), p. 1, ACM Press, 2011.

[25] J. Kopf, “360° video stabilization,” *ACM Transactions on Graphics*, vol. 35, pp. 1–9, 11 2016.

[26] A. Serrano, V. Sitzmann, J. Ruiz-Borau, G. Wetzstein, D. Gutierrez, and B. Masia, “Movie editing and cognitive event segmentation in virtual reality video,” *ACM Transactions on Graphics*, vol. 36, pp. 1–12, 7 2017.