Scalable Motion-Aware Panoramic Videos

Leonardo Sacht Luiz Velho
Diego Nehab Marcelo Cicconet
IMPA, Rio de Janeiro, Brazil

September 16, 2011

Abstract

The work presents a method for obtaining perceptually natural panoramic videos, that is, videos composed of wide-angle frames, where straight lines and object shapes are preserved. The shape preservation for moving objects has a special treatment in order to avoid unpleasant temporal artifacts. The problem is geometrically modeled for the case of a fixed omnidirectional camera, and a solution by optimization is presented. Our optimization works per-frame, which makes the proposed method scalable and practical for arbitrarily long scenes. Results show the potential of this technique as a way to provide more information in a movie, and reveals new possibilities for the art of film making.

1 Introduction

We present a method for obtaining perceptually natural panoramic videos. By panoramic videos we mean videos where each frame is a wide-angle image. The notion of naturalness is represented by a set of energy terms describing perceptual properties such as shape and straight-line preservation. Shape preservation of moving objects across time is especially treated.

Since their invention, video cameras evolved a lot in terms of image quality, image representation, zoom and portability. On the other hand, the language of shooting videos remains nearly the same. Thus, movie making techniques consist mainly of changing the position of the camera, the direction which it points to and the zoom of the lens (in some narrow field of view range). Even artistic explorations (e.g., the *Hitchcock zoom*) do not depart very much from these settings.

With the recent developments in computational photography, omnidirectional video cameras appeared. They are now being widely used for navigation and immersion purposes (e.g., Google Street View). However, the way in which they are applied simulates traditional cameras. The only difference is that now the user decides the view direction and zoom angle.

We consider a new use for the spherical videos provided by these cameras: videos with wide-angle frames. We improve on recent developments in the understanding of the distortions involved in wide-angle images [Zorin and Barr 1995; Zelnik-Manor et al. 2005; Kopf et al. 2007, 2009; Carroll et al. 2009], and study the implications of the introduction of the time dimension.

We consider the case in which the viewpoint and the field of view are fixed, which allows a variety of applications such as sport broadcasting, film making, and surveillance.

Since the only difference between our problem and the one of computing static panoramic images are the moving objects, one could think of segmenting the moving objects and computing separate projections for the background and the foreground. This approach presents some problems: first, a precise segmentation (at subpixel level) would be required. Second, a method for accurately combining images from the different projections would also be necessary. Finally, the most difficult problem would be the imposition of spatial and temporal relations between the different projections for background and foreground.

Another possibility would be to solve the problem for both background and foreground at the same time, considering all the video as a space-time volumetric mesh. Shape preservation for the moving objects in different frames could be imposed in the same way methods for other problems already did (for example, [Wang et al. 2009] for video resizing). This solution suffers from some drawbacks: first, the related optimization problem can easily reach the size of millions of variables. Solving for smaller parts of the video can alleviate this problem but compromises temporal coherence. But the most important issue is that imposing coherence for moving objects can affect the background, and even if the background changes smoothly this would be noticeable. For example, it would be unacceptable for the floor or walls of a room to move during a video.

Our solution is a hybrid of the two possibilities mentioned above. We first compute a minimum energy projection for the background, using novel energy terms which substantially improve previous ones proposed in the panoramic image literature. We then use this background projection as a reference to compute an optimizing projection for each frame, based on energy terms designed specifically to avoid distorting moving objects and to consider temporal variations coming

from neighboring frames. Finally, we perform another optimization to correct the background areas that were affected by the intermediate step.

Since our method solves the problem separately for each frame, it is scalable and practical for arbitraitly long films. The linear systems that have to be solved for each frame are well conditioned, which enables the method to quickly converge to an accurate solution. Moreover, the method is simple to be implemented and leads to high quality results.

The text is structured as follows. We review the literature and discuss previous work in Section 2. The pipeline of the proposed method is described in Section 3. Section 4 details the spatial constraints and the first optimization step of our method. Section 5 approaches the temporal requirements and the other two optimizations that are performed for each frame. Section 6 gives some implementation details and presents some results. The text concludes with remarks about limitations and future work in Section 7.

2 Related Work

As the availability of point-and-shoot cameras capable of producing panoramic images increases, the problem of preserving straight lines and object shapes in the resulting projections has gained renewed practical relevance.

Early work in this area [Zorin and Barr 1995] formalized desirable properties for wide-angle images and showed that the preservation of object shapes and straight lines cannot be satisfied simultaneously. The alternative the authors proposed was to employ a family of transformations that can achieve a compromise between these two constraints according to a parameter and minimize the overall distortion of the final picture.

Follow-up work in this area started to use the scene content. Agrawala et al. [2000] use local projections to introduce shape distortions of individual objects, allowing for artistic control. Zelnik-Manor et al. [2005] use local projections for the foreground objects and multi-plane perspective projections for the background. Discontinuities appearing in the intersection between projection planes are handled by choosing these planes in a way that fits the geometry of the scene.

The method by Kopf et al. [2009] starts with a projection cylinder and deforms it in order to rectify user specified planar regions in the scene. The work of Carroll et al. [2009] formulates energies that measure how a panoramic image creates distortions such as line bending and shape stretching and find a least distorted

projection via an optimization process.

Between all the works that deal with panoramic images, our work is most closely related to [Carroll et al. 2009]. The spatial constraints we use to obtain a good projection for the background are similar to theirs. But we have made some significant improvements to their formulation: (1) Our optimization process has a proof of convergence; (2) Our line constraints are more uniformly distributed along the lines. Their approach imposed considerably more constraints on the endpoints; (3) Our extra energy term to make the final linear system nonsingular is more natural, since it fixes scale and general position of the projection. In contrast, they add a term that penalizes the projection from deviating from the stereographic projection.

The most straightforward approach to generating a panoramic video would be to directly use well-established panoramic image methods in a frame by frame manner. However, since these methods do not consider temporal variations in the scene, undesirable distortions in the final result would appear. For instance, applying the work of Zelnik-Manor et al. [2005] would cause objects to have strong variation in orientation when they pass over the intersection of projection planes. A similar behavior would happen if one applied the approach by Kopf et al. [2009] to scenes with intersecting or close planar regions. Temporal incoherences of scale and orientation for the moving objects would also happen when using the method of Carroll et al. [2009], especially near line segments.

Temporal incoherences as the ones mentioned above were already considered in other contexts, such as video resizing. Between all the works in this subject, the ones closest to ours are [Wang et al. 2009], [Wang et al. 2010] and [Wang et al. 2011]. The preservation of moving objects that we impose are related to the ones proposed by Wang et al. [2009] and our concern of making the method scalable has connections with [Wang et al. 2011]. But we explore these ideas in a different way and in a very different context.

Although the problem of generating videos composed of wide-angle frames has not yet been thoroughly investigated, a few additional related works are worthy of mention. In [Agarwala et al. 2005], for instance, the authors produce a video with wide-angle frames from a set of common photographs, by transferring textures between frames in a coherent way. This work, however, does not consider geometric distortions, and is restricted to scenes with a particular structure: those that allow the construction of video textures. For immersion purposes, works such as [Neumann et al. 2000], [Kimber et al. 2001], and [Uyttendaele et al. 2004] generate video in which each frame has a narrow FOV. Similar ideas are already available

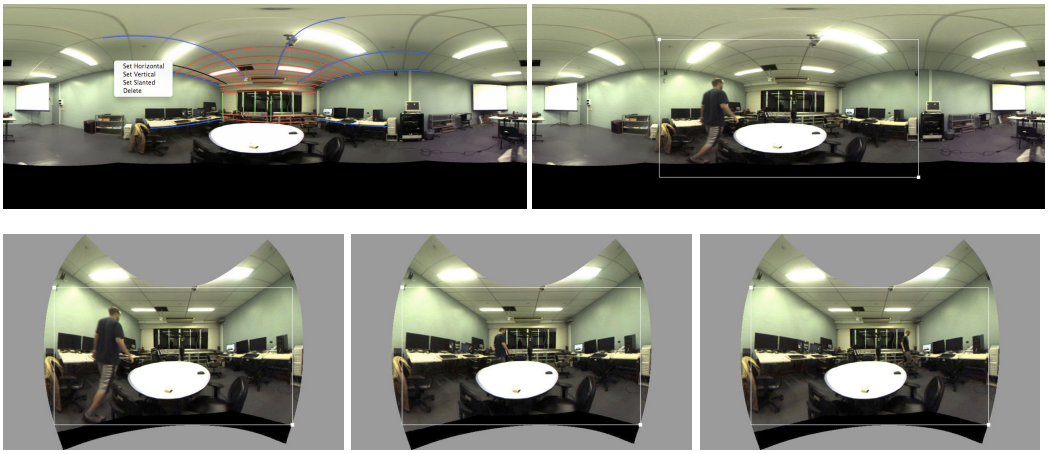


Figure 1: *Top: in the pre-processing phase, straight lines are marked on the background images (left) and the field of view is specified on the foreground images (right). Bottom: the greatest box contained in all frames is suggested as the crop area for the generation of the final video.*

on the web[1].

3 Overview

The pipeline for the panoramic video generation method that we propose is as follows.

Since we are dealing with the case of a fixed camera and moving objects, we treat background and foreground objects in different ways. Thus, the first step consists of capturing two panoramic videos: one with the background scene, and the other with the moving objects. The source videos are omnidirectional: a series of equirectangular frames, shot with a 5-lens spherical camera (Ladybug2, by Point Grey).

Then, three phases are performed: pre-processing, optimization and post-processing. We will detail here the phases performed with user interaction (pre- and post-processing). The optimization procedure will be described in the following sections.

In the *pre-processing* phase, the user loads the background and foregrounds images of the scene, in separate windows. Over the background images, straight

[1]See http://edition.cnn.com/interactive/2010/01/world/haiti.360/

lines are marked, in the same manner as was done in [Carroll et al. 2009] for panoramic images (Figure 1, top left). Lines are tagged as horizontal, vertical or slanted (of undetermined orientation). Foreground images are used to specify the area of the images that will be transformed (Figure 1, top right). While the user works on these operations, background subtraction is performed in order to detect the moving objects.

The *post-processing* phase consists of cropping an area of the distorted images to produce the final video. The solution of the optimization problem is a vector with the positions where the points of a grid superimposed on the original frames should go. These positions, along with the corresponding texture coordinates, are used to suggest to the user a possible region for cropping (Figure 1, bottom), which can be edited if desired.

4 Spatial Constraints

Our method consists of distorting the set of input equi-rectangular images in order to produce a temporally consistently deformed panoramic video. Thus, we want to find a projection

$$\begin{aligned} \mathbf{U} : [\lambda_o, \lambda_f] \times [\phi_o, \phi_f] \times [0, t_f] &\to \mathbb{R}^3 \\ (\lambda, \phi, t) &\mapsto (U(\lambda, \phi, t), V(\lambda, \phi, t), t) \end{aligned} \tag{1}$$

with desirable properties. Above, $[\lambda_o, \lambda_f] \times [\phi_o, \phi_f] \subseteq [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the FOV specified by the user and $[0, t_f]$ is a time interval. Given integers m and n we define the discretization

$$\phi_i = \phi_o + i\Delta\phi,\ \lambda_j = \lambda_o + j\Delta\lambda,\ t_k = k\Delta t, \tag{2}$$

where $i = 0, \ldots, m-1$, $j = 0, \ldots, n-1$, $k = 0, \ldots l-1$, $\Delta\phi = \frac{\phi_f - \phi_o}{m-1}$, $\Delta\lambda = \frac{\lambda_f - \lambda_o}{n-1}$ and l is the number of frames. With this setting, we replace the continuous problem by finding

$$\mathbf{U}_{ijk} := (U_{ijk}, V_{ijk}, t_k) = \mathbf{U}(\lambda_j, \phi_i, t_k). \tag{3}$$

Before computing a discretized projection for the whole video, we formulate in this section energies that measure how the discretized projection of the background deviates from well known requirements for panoramic images. By minimizing these energies, we obtain a good projection for the background which consists in the first step of our panoramic video generation system. The unknowns

U_{ij} and V_{ij} for the background (the index k is not necessary in this case) are put together in a solution vector denoted by $\mathbf{x}_{bg}$.

Conformality and smoothness: The energies that we use to model these properties are very similar to the ones proposed by Carroll et al. [2009], with the main differences being that we constrain the variation of both north and east differential vectors in the smoothness energy (their formulation constrains only the north vector, which can lead to additional temporal incoherences in panoramic videos), we do not use the proposed spatially-varying weights (we did not find them necessary in our framework) and we weight the constraints by the area of the quads, instaed of a factor of the area. We denote by $E_c(\mathbf{x}_{bg})$ the conformality energy and by $E_s(\mathbf{x}_{bg})$ the smoothness energy.

Straight lines: Let L be a user specified line segment in the spherical domain and $\mathbf{n}_L$ a normal vector associated to the projection of L in the final result. For example, if the user specifies L to be vertical in the final result, then $\mathbf{n}_L = \begin{pmatrix} 1 & 0 \end{pmatrix}^T$. If L is set to be horizontal, then $\mathbf{n}_L = \begin{pmatrix} 0 & 1 \end{pmatrix}^T$. If the user does not specify an orientation for L, then $\mathbf{n}_L$ is unknown.

Let also V_L be the set of edges of the discretization grid of the domain intersected by L. For each vertical edge $\mathbf{e}_w = \{(\lambda_{j_w}, \phi_{i_w}), (\lambda_{j_w}, \phi_{i_w+1})\} \in V_l$, we define a virtual output vertex

$$\mathbf{U}_w^L = a_w \mathbf{U}_{i_w, j_w} + b_w \mathbf{U}_{i_w+1, j_w}, \tag{4}$$

where the values a_w and b_w are obtained in the same way did by Carroll et al. [2009], with the only difference that we interpolate only between two points (the edge endpoints), while they interpolate between four. We define output vertices in an analogous way for horizontal edges.

The following energy is defined to express how the projection of L deviates from the normal direction to $\mathbf{n}_L$, for all line segments L:

$$E_l(\mathbf{x}_{bg}, \{\mathbf{n}_L\}) = \sum_{L \ line} \sum_{\mathbf{e}_w \in V_L} \left(\mathbf{n}_L^T \left(\ \mathbf{U}_{w+1}^L - \mathbf{U}_w^L \ \right) \right)^2. \tag{5}$$

Above, $\mathbf{U}_{w+1}^L$ is the virtual output vertex defined by $\mathbf{e}_{w+1}$, the edge intersected by L immediately after $\mathbf{e}_w$.

Assuming a solution $\mathbf{x}_{bg}$ is available we can fix $\mathbf{x}_{bg}$ in (5) and minimize E_l for the unknown normals $\mathbf{n}_L$. For each line L, the normal for the other lines do not affect its energy value, thus the minimization can be performed separately for each line L. It is natural that we impose $\| \ \mathbf{n}_L \ \|^2 = 1$. Then the problem becomes

$$\arg\min \sum_{\mathbf{e}_w \in V_L} \left(\mathbf{n}_L^T \left(\ \mathbf{U}_{w+1}^L - \mathbf{U}_w^L \ \right) \right)^2, s.t. \ \| \ \mathbf{n}_L \ \|^2 = 1. \tag{6}$$

By using the Lagrange optimality conditions, one obtains that $\mathbf{n}_L$ is the unitary eigenvector associated to the smallest eigenvalue of $P^T P$ where the lines of the matrix P are given by

$$P_{w*} = \left(\ \mathbf{U}_{w+1}^L - \mathbf{U}_w^L \ \right)^T, \ \forall \mathbf{e}_w \in V_L. \tag{7}$$

Our optimization process to obtain a projection for the background will alternate between optimizing for the normals $\mathbf{n}_L$ and for the discretized projection $\mathbf{x}_{bg}$. More details will be provided at the end of this section.

Fixing some positions: Up to now, all energies we presented are annihilated by constant projections, making these trivial projections global minimizers for any combination of them. To avoid this problem, we propose a new term that fixes the V coordinates of the corner points of the discretization of the domain $[\lambda_o, \lambda_f] \times [\phi_o, \phi_f]$:

$$E_a(\mathbf{x}_{bg}) = (V_{00} - 0)^2 + (V_{0,n-1} - 0)^2 \quad + (V_{m-1,0} - 1)^2 + (V_{m-1,n-1} - 1)^2. \tag{8}$$

This term can also be seen as a term that prevents arbitrary scales and translation of the points, behavior that was not prevented by the other energy terms.

We are now ready to describe the first step in our optimization. The other two steps are described in the next section.

Step 1 - Optimal background projection: The optimal projection for the background $\mathbf{x}_{bg}$ is obtained by solving the following minimization problem:

$$\arg\min \ E_{bg}(\mathbf{x}_{bg}, \{\mathbf{n}_L\}), \ s.t. \parallel \mathbf{n}_L \parallel^2 = 1, \tag{9}$$

for all line segments L with no specified orientation, where

$$E_{bg}(\mathbf{x}_{bg}, \{\mathbf{n}_L\}) = w_c^2 E_c(\mathbf{x}_{bg}) + w_s^2 E_s(\mathbf{x}_{bg}) \quad + w_l^2 E_l(\mathbf{x}_{bg}, \{\mathbf{n}_L\}) + w_a^2 E_a(\mathbf{x}_{bg}) \tag{10}$$

For all results in this paper, we have set $w_c = 1$, $w_s = 0.5$, $w_l = 3$ and $w_a = 0.01$.

We solve this problem as follows: first, we drop off from (10) the terms in E_l which depend on the normals $\{\mathbf{n}_L\}$ and optimize all the other terms to obtain $\mathbf{x}_{bg}^{(0)}$. Then we obtain a set of optimal normals $\{\mathbf{n}_L^{(0)}\}$ by solving (6) with $\mathbf{x}_{bg} = \mathbf{x}_{bg}^{(0)}$. The next step is to fix $\{\mathbf{n}_L\} = \{\mathbf{n}_L^{(0)}\}$ in (10) and solve the optimization problem

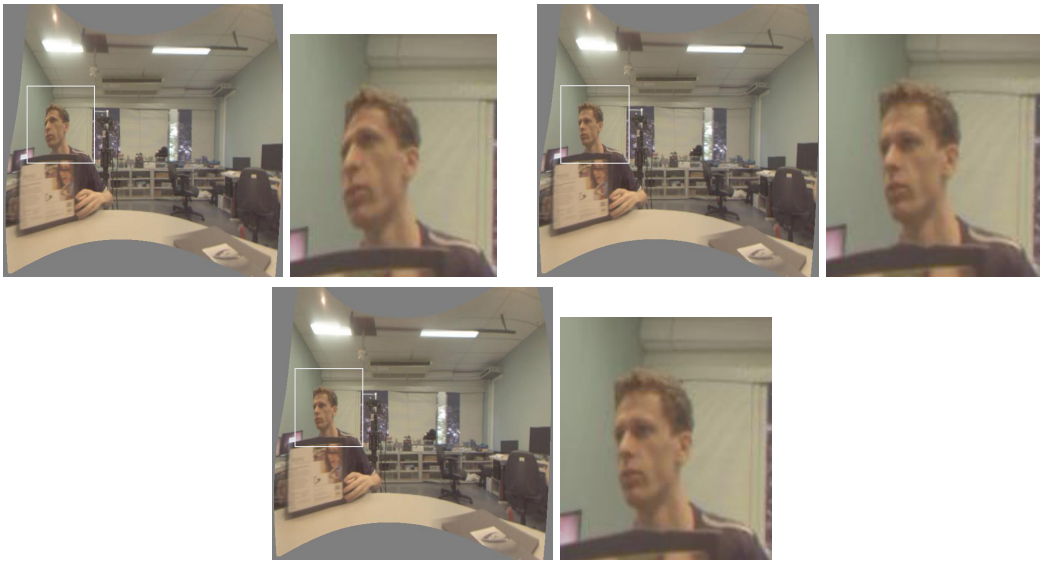


Figure 2: *Detail of one frame of the result video after each of the three steps of our method. Top-left: Result video of step 1. The man's face present unacceptable distortions. Top-right: Result video of step 2. The face is much less distorted, but now some lines near the man are bent. Bottom: Result video of step 3. The head's shape is preserved, and the lines are a little more correct.*

to obtain $\mathbf{x}_{bg}^{(1)}$. By continuing this process we obtain a decreasing sequence of energy values

$$E_{bg}(\mathbf{x}_{bg}^{(0)}, \{\mathbf{n}_L^{(0)}\}) \geqslant E_{bg}(\mathbf{x}_{bg}^{(1)}, \{\mathbf{n}_L^{(0)}\}) \geqslant E_{bg}(\mathbf{x}_{bg}^{(1)}, \{\mathbf{n}_L^{(1)}\}) \geqslant \dots \qquad (11)$$

and, since E_{bg} is bounded below by zero, we have that the sequence of energy values is convergent.

5 Temporal Constraints

Applying the background projection for all frames can result in temporal inconsistencies on the moving objects, such as the ones shown in Figure 2 (top-left). In this example, strong line constraints on the background severely distorted the man's head leading to undesirable artifacts.

In this section we design temporal constraints to avoid such inconsistencies and propose a solution for the resulting optimization problem. We denote by $\mathbf{x}_k$ the vector that contains the unknown positions U_{ijk} and V_{ijk}. The optimization is performed separately for each $\mathbf{x}_k$, which makes our method scalable to long films.

Shape preservation of moving objects: The inconsistencies observed in Figure 2 (top-left) are caused by strong variations of the differential north and east vectors, which are defined by

$$\mathbf{H}(\lambda, \phi, t) = \begin{pmatrix} \frac{\partial U}{\partial \phi}(\lambda, \phi, t) \\ \frac{\partial V}{\partial \phi}(\lambda, \phi, t) \end{pmatrix} \tag{12}$$

and

$$\mathbf{K}(\lambda, \phi, t) = \frac{1}{\cos(\phi)} \begin{pmatrix} \frac{\partial U}{\partial \lambda}(\lambda, \phi, t) \\ \frac{\partial V}{\partial \lambda}(\lambda, \phi, t) \end{pmatrix}, \tag{13}$$

in different areas of the background projection. When an object passes over these areas, these variations become more pronounced, leading to unpleasant effects.

To avoid these problems we restrict the projection to be smoother in moving object areas, by avoiding variation of the vectors $\mathbf{H}$ and $\mathbf{K}$. For all points (λ_j, ϕ_i, t_k) detected as belonging to an object, if $(\lambda_{j+1}, \phi_i, t_k)$ and $(\lambda_j, \phi_{i+1}, t_k)$ also belong to a moving object, we enforce

$$\begin{cases} \mathbf{H}(\lambda_j, \phi_i, t_k) = \mathbf{H}(\lambda_{j+1}, \phi_i, t_k) \\ \mathbf{K}(\lambda_j, \phi_i, t_k) = \mathbf{K}(\lambda_{j+1}, \phi_i, t_k) \end{cases} \tag{14}$$

and

$$\begin{cases} \mathbf{H}(\lambda_j, \phi_i, t_k) = \mathbf{H}(\lambda_j, \phi_{i+1}, t_k) \\ \mathbf{K}(\lambda_j, \phi_i, t_k) = \mathbf{K}(\lambda_j, \phi_{i+1}, t_k) \end{cases} \tag{15}$$

These requirements alone could still lead to temporal incoherences, since no information about the neighboring frames is being considered. For example, since we use a mesh which is coarser than the input image resolution, inconsistencies at object borders could appear. Also, abrupt changes in the scene such as objects suddenly coming in and out of the frame could lead to changes in the resulting video. Finally, the use of the segmentation in individual frames would make any errors in the detected foreground object immediately apparent.

To consider the information coming from neighboring frames, we impose (14) and (15) to points that belong to objects in adjacent past and future instants, as illustrated in Figure 3. For all points $(\lambda_j, \phi_i, t_{k+l})$ detected as object at time t_{k+l}, if $(\lambda_{j+1}, \phi_i, t_{k+l})$ $(\lambda_j, \phi_{i+1}, t_{k+l})$ belong to an object, we ask (14) and (15) to be satisfied. Above, $l \in \{-\frac{w}{2}+1, -\frac{w}{2}+2, \ldots, -1, 0, 1, \ldots, \frac{w}{2}-2, \frac{w}{2}-1\}$, where w corresponds to a chosen window size. Observe that we are only constraining

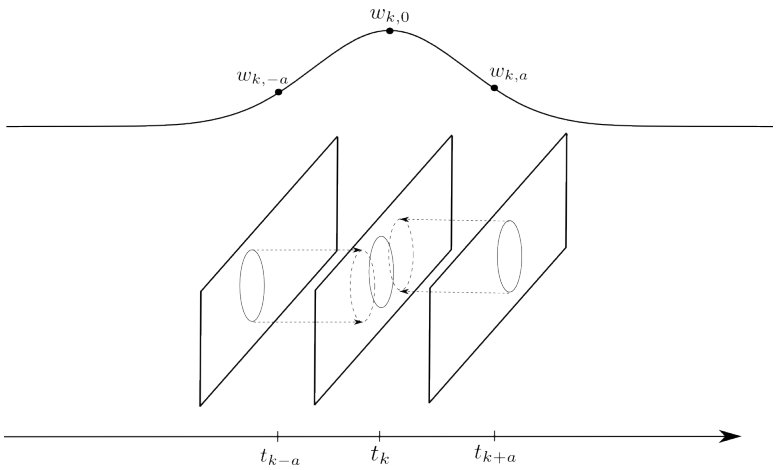


Figure 3: *We consider constraints coming from close past and future frames. These constraints are multiplied by gaussian weights.*

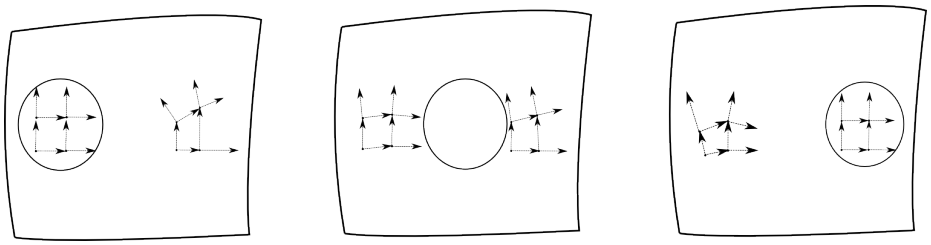


Figure 4: *Illustration of the result of our temporal constraints. The object is consistently deformed across time.*

points at time t_k, which makes the problem of obtaining $\mathbf{x}_k$ independent of the other frames. We multiply this constraints by the gaussian weights

$$w_{k,l} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{l^2}{2\sigma^2}}, \tag{16}$$

which makes constraints coming from closer frames stronger than the ones coming from more distant frames. All results we present in this paper were generated using $w = 32$ and $\sigma = \frac{w}{2}$.

We illustrate the result of our temporal requirements in Figure 4. An object moves from the left of the projection to the right. In the first frame, the vectors $\mathbf{H}$ and $\mathbf{K}$ at the area occupied by the object are consistent. As the object moves away (second frame in Figure 4) these vectors start to smoothly change, tending to be the vectors of the projection of the background. On the other side, the vectors of the area that the object will occupy in the future start to be more consistent. In

third frame, the vectors on the right are equal and the vectors on the left are the background ones.

Discretizing (14) and (15) using finite differences at each $(\lambda_j, \phi_i, t_k) \in Ob^{k+l}$, where

$$\begin{aligned} Ob^{k+l} =&\{(\lambda_j, \phi_i, t_k)|(\lambda_j, \phi_i, t_{k+l}), (\lambda_{j+1}, \phi_i, t_{k+l}), \\ &(\lambda_j, \phi_{i+1}, t_{k+l}) \text{ belong to an object}\}, \end{aligned} \tag{17}$$

multiplying the equations by $w_{k,l}$ and by $\cos(\phi_i)$ to compensate the spherical distortions, we obtain the following energy for moving object shape preservation:

$$E_{sh}(\mathbf{x}_k) = E_{sh,0} + E_{sh,1} + E_{sh,2} + E_{sh,3} \quad + E_{sh,4} + E_{sh,5} + E_{sh,6} + E_{sh,7} \tag{18}$$

In the energy above, four terms correspond to the discretization of (14) and the other four correspond to the discretization of (15). For example, the first equation in (14) lead to the term

$$\begin{aligned} E_{sh,0}(\mathbf{x}_k) = \sum_{l=-\frac{w}{2}+1}^{\frac{w}{2}-1} \sum_{(\lambda_j,\phi_i,t_k)\in Ob^{k+l}} &\big(w_{k,l}\cos(\phi_i)\tfrac{U_{i+1,j,k}-U_{ijk}}{\Delta\phi} \\ &- w_{k,l}\cos(\phi_i)\tfrac{U_{i+1,j+1,k}-U_{i,j+1,k}}{\Delta\phi}\big)^2, \end{aligned} \tag{19}$$

The other terms are analogous and are omitted for conciseness.

Step 2 - Optimization of the shape preservation energy: The intermediate step of our method consists of minimizing the energy just proposed and restricting the points that do not belong to moving objects throughout the entire time window to be projected as in the solution of step 1. This is achieved by minimizing

$$E_{ob}(\mathbf{x}_k) = \gamma^2 E_{sh}(\mathbf{x}_k) + \parallel \mathbf{x}_k - \mathbf{x}_{bg} \parallel^2, \tag{20}$$

where $\mathbf{x}_{bg}$ is the solution calculated for the background in step 1. For all the results in this paper, we have set $\gamma = 2.5$.

Step 3 - Optimization to combine foreground and background: The solution obtained in step 2 for $\mathbf{x}_k$ is not satisfactory yet. As can be seen in Figure 2 (top-right), the extra constraints for the man rectified his shape but distorted the region around him (the line on the right of his head is not as straight as desired, for example).

We fix this problem by re-optimizing the spatial energies plus a term that considers how much the projection deviates from the one obtained in step 2 (say $\mathbf{x}_k^{ob}$). Thus, we minimize

$$E_{final}(\mathbf{x}_k) = E_{bg}(\mathbf{x}_k, \{\mathbf{n}_L\}) + \gamma^2 \parallel \mathbf{x}_k - \mathbf{x}_k^{ob} \parallel^2, \tag{21}$$

where we choose the set of normals $\{\mathbf{n}_L\}$ for lines with no specified orientation as the last one obtained in step 1. The result for the example we are considering is shown in Figure 2 (bottom).

We observe that, although we have not proposed any energy term for temporal coherence of the background, the preservation of the background along time is a consequence of our formulation. After step 2, most of the points (except the moving objects in the time window) have the same projection as the one calculated for the background in step 1. After step 3, since the final energy is almost the same as in step 1, the result tends to be the same for background points as the one obtained in step 1.

6 Implementation and Results

To solve the linear systems associated to minimize (10), (20) and (21), we used PETSc[2], a toolkit that implements many Krylov methods for solving sparse linear systems. Between these methods, we chose the Conjugate Gradient method with SOR pre-conditioner (details can be found in [Saad 2003]).

The calculation of the optimal normals in (6) is done by using explicit formulas for the matrix $P^T P$ and its eigenvectors. This is possible since these problems are two dimensional problems. By alternating three times between minimizing the background energy for $\{\mathbf{n}_L\}$ and for $\mathbf{x}_{bg}$ we usually obtain satisfactory convergence.

Once the solutions for all frames are computed, a set of meshes is generated and passed to OpenGL for texture mapping (textures are, of course, the corresponding input equirectangular images). The user then determines, over the distorted images, the area that will be cropped to generate the final video (as explained in Figure 1).

For background subtraction we used OpenCV [Bradski and Kaehler 2008] and for MOV video generation we used Quicktime. Some image handling operations were performed using Netpbm[3].

We now present some results of our method. All these results are presented in the accompanying video of this paper.

We typically used for each frame of each video a mesh with resolution around 70×70 vertices. The results did not vary too much for finer meshes, so we

[2]http://www.mcs.anl.gov/petsc/

[3]http://netpbm.sourceforge.net/

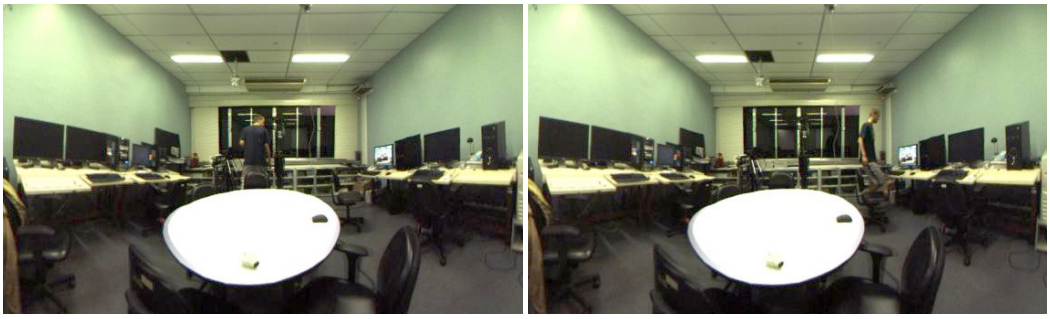


Figure 5: *Two frames frames of a result of our method.*

decided to use the minimum density that produces acceptable results.

All the results were generated using a desktop PC with an Intel Xeon Quad Core 2.13GHz and 12 GB of RAM. Each minimization for $\mathbf{x}_{bg}$ in Step 1 took about 3 seconds and 2,000 iterations of Conjugate Gradient method to converge. Solving steps 2 and 3 for each frame was much faster, taking in average 0.05 seconds and 100 iterations for step 2 and 0.005 seconds and 10 iterations for step 3. We conclude that calculating the projection for the background takes much longer, but this is not a problem, since we only compute it once.

The first result is shown in Figures 5 and 1. This video comprehends a field of view of 170 degrees longitude by 110 degrees latitude. It has 170 frames.

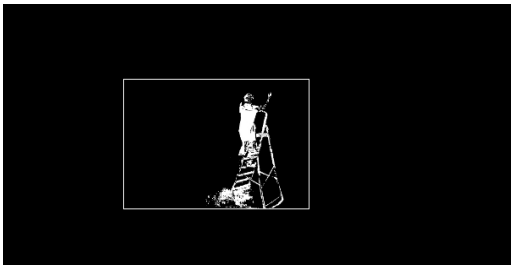
Figures 6 and 7 show input data for our method and some frames of the results. In figure 6 we show the lines marked by the user and a frame of the foreground/background segmentation (the FOV is also indicated). The result consists of a 120×90 FOV and has 60 frames. In figure 7 we show the lines and the specified FOV, which is 215×120 degrees. This video has 100 frames.

Finally, we compare a result of our method with well known standard projections in Figure 8. We can see that our result is the only one that has the property of preserving straight lines and both moving and still objects in the scene. This result also evidences the scalability of our method. It has 800 frames and is one minute long, and the the computational time to solve the optimization for each frame was about 0.05 seconds. Each frame comprehends 135×100 degrees.

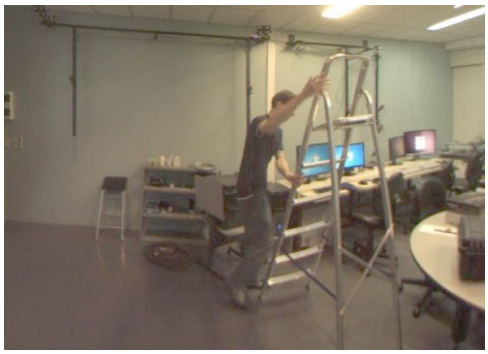
7 Limitations and Future Work

The result in Figure 7 of the previous section reveals a weakness of our method. Objects too close to the camera cannot appear uniformly conformal in the final

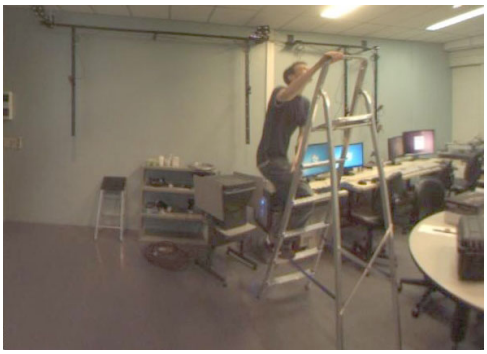
(a) User specified lines.



(b) Segmentation and specified FOV.

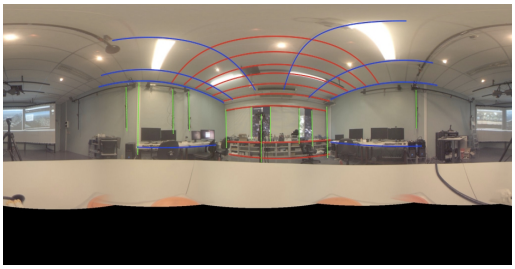


(c) 4th frame.



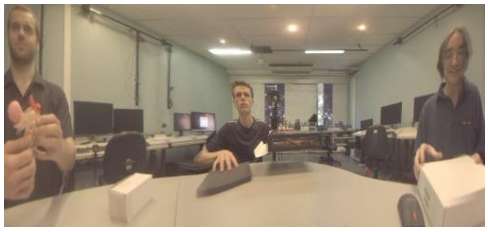
(d) 25th frame.

Figure 6: *(a), (b): Input data for our method. (c), (d): Frames of the result video.*



(a) User specified lines.

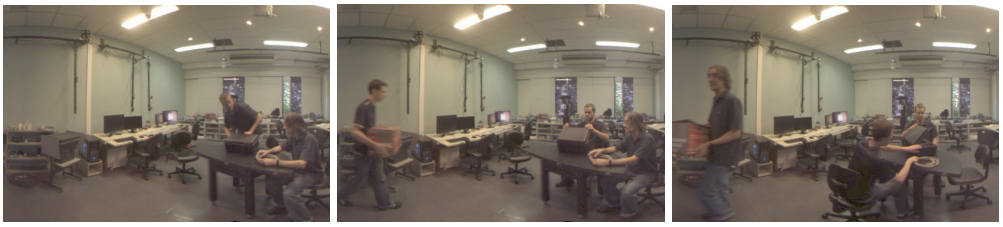
(b) FOV specified in the interface.



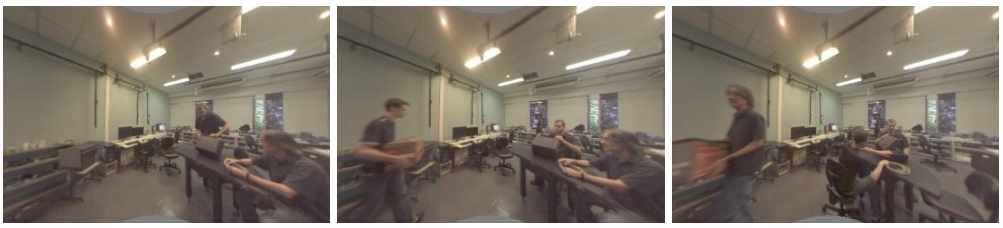
(c) 57th frame.

(d) 94th frame.

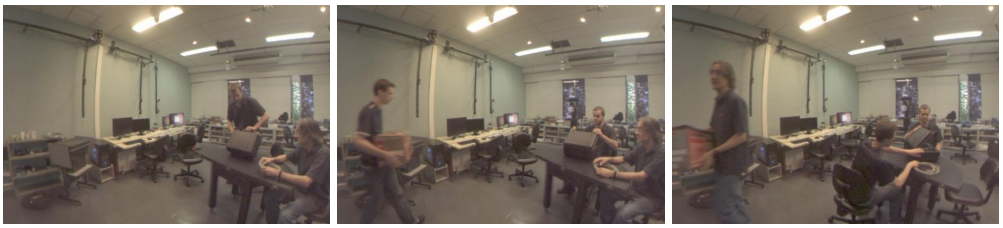
Figure 7: *(a), (b): Input data for our method. (c), (d): Frames of the result video.*



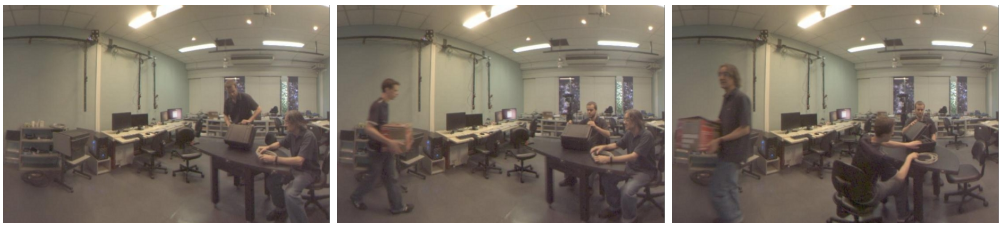
(a) Equi-rectangular projection.



(b) Perspective projection.



(c) Stereographic projection.



(d) Mercator projection.

(e) Our optimized projection.

Figure 8: *Comparison of our result with standard projections. As can be seen, equi-rectangular (a), stereographic(c) and Mercator (d) projections do not preserve straight lines. On the other hand, perspective projection (b) preserves lines but distorts objects. Our result (e) conciliates both straight line and object preservation.*

result. This property, which we enforced on the moving objects, is not satisfied by the table in Figures 7(c) and 7(d). Even the man on the right, which should be more consistently deformed, appears a little distorted in Figure 7(c). This could be fixed by prescribing an additional restriction to correct for such distortions.

In our method, we used only line segments marked over the background without considering lines over the moving objects. Line bending is usually a problem only for long line segments which occupy a wide FOV. Since moving objects usually occupy a narrow FOV, line bending over moving objects did not turn out to be a severe problem in our examples.

The need for background images could also be considered a limitation of our work, but the background images are only important to determine whether or not a region belongs to a moving object. Other possibilities for determining moving object areas, such as user marking or optical flow computation, can be used instead.

As mentioned in the introduction, our method is restricted to the case where both the field of view and viewpoint do not change across time. We think simple extensions of what was proposed in this work can handle the case of a moving FOV, such as introducing time-varying position constraints for some points on the varying FOV. For the more general case, where both FOV and viewpoint are moving across time, we could use the optical flow of the input video to transport geometrical properties of corresponding points in a temporally coherent manner.

Our minimization based approach offers the possibility of adding extra energy terms to control the distortion of scene features. For example, moving features could be preserved in the same way Wang et al. [2011] did for the video resizing problem. Artistic perspective control could be included in our formulation with energy terms similar to the ones proposed by Carroll et al. [2010].

In conclusion, we have seen that the method presented in this work allows the introduction of content in movies in a realistic manner. It would be interesting to provide this tool to film makers for exploring scenes and stories that would be told differently when using a camera with a narrower field of view. In a small room, for instance, the director has to perform a cut in the scene to follow the dialog of two actors in opposite ends of a wall, but a panoramic camera with the proper angle of view could allow both actors to appear in the same view simultaneously. In fact we are already in conversation with a film professional regarding the exploration of possibilities like this.

References

AGARWALA, A., ZHENG, K. C., PAL, C., AGRAWALA, M., COHEN, M., CURLESS, B., SALESIN, D., and SZELISKI, R. 2005. Panoramic video textures. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2005)*, pages 821–827.

AGRAWALA, M., ZORIN, D., and MUNZNER, T. 2000. Artistic multiprojection rendering. In *Proc. of the Eurographics Workshop on Rendering Techniques*, Springer-Verlag, pages 125–136.

BRADSKI, D. G. R. and KAEHLER, A. 2008. *Learning OpenCV*. O'Reilly Media, Inc., 1st edition.

CARROLL, R., AGARWALA, A., and AGRAWALA, M. 2010. Image warps for artistic perspective manipulation. *ACM Transactions on Graphics*, 29(4):127.

CARROLL, R., AGRAWAL, M., and AGARWALA, A. 2009. Optimizing content-preserving projections for wide-angle images. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2009)*, 28(3):43.

KIMBER, D., FOOTE, J., and LERTSITHICHAI, S. 2001. FlyAbout: Spatially indexed panoramic video. In *Proc. Multimedia 2001*, ACM, pages 339–347.

KOPF, J., LISCHINSKI, D., DEUSSEN, O., COHEN-OR, D., and COHEN, M. F. 2009. Locally adapted projections to reduce panorama distortions. *CGF (Proc. of EGSR'09)*, 28(4):1083–1089.

KOPF, J., UYTTENDAELE, M., DEUSSEN, O., and COHEN, M. F. 2007. Capturing and viewing gigapixel images. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2007)*, 26:93.

NEUMANN, U., PINTARIC, T., and RIZZO, A. 2000. Immersive panoramic video. In *Proc. of Multimedia 2000*, ACM, pages 493–494.

SAAD, Y. 2003. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition.

UYTTENDAELE, M., CRIMINISI, A., KANG, S. B., WINDER, S., SZELISKI, R., and HARTLEY, R. 2004. Image-based interactive exploration of real-world environments. *IEEE CG&A*, 24(3):52–63.

WANG, Y.-S., FU, H., SORKINE, O., LEE, T.-Y., and SEIDEL, H.-P. 2009. Motion-aware temporal coherence for video resizing. *ACM Trans. Graph. (Proceedings of ACM SIGGRAPH ASIA)*, 28(5).

WANG, Y.-S., HSIAO, J.-H., SORKINE, O., and LEE, T.-Y. 2011. Scalable

and coherent video resizing with per-frame optimization. *ACM Trans. Graph. (Proceedings of ACM SIGGRAPH*, 30(4).

WANG, Y.-S., LIN, H.-C., SORKINE, O., and LEE, T.-Y. 2010. Motion-based video retargeting with optimized crop-and-warp. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*, 29(4):article no. 90.

ZELNIK-MANOR, L., PETERS, G., and PERONA, P. 2005. Squaring the circles in panoramas. In *Proc. of ICCV*, IEEE Computer Society, pages 1292–1299.

ZORIN, D. and BARR, A. H. 1995. Correction of geometric perceptual distortions in pictures. In *Proceedings of ACM SIGGRAPH 1995*, pages 257–264.