STAG: Smart Tools and Apps for Graphcis (2014), pp. 1–10
Andrea Giachetti (Editors)

Stereo-browsing from Calibrated Cameras

Abstract

Modern Structure-from-Motion (SfM) methods enable the registration of a set of cameras and the reconstruction of the corresponding sparse point cloud of the object/location depicted in the input images. Despite the quality of such techniques they often fail where the input point cloud has a very low density thus decreasing the final user experience. On the other side, modern image-based rendering (IBR) techniques try to avoid a full reconstruction of the geometry by empowering the user with interfaces for the smooth navigation of the acquired images. In such methods, images viewed from viewpoints in-between the actual cameras are generated in some way, for example by using a textured proxy or by warping properly the input images. Usual navigation interfaces, however, neglect to use the inherent nature of such set of cameras which, despite having a wide-baseline, are often well spatially organized as they usually maintain a good overlap between images, varying smoothly both the position and the orientation of the camera. Given such a set of registered cameras, we present a framework for the stereoscopic exploration of the object/location depicted using any type of stereoscopic devices. In the proposed system, the users can have a full tridimensional experience without the need of a complete 3D reconstruction. Our method starts by building a graph where each node is associated to a calibrated camera that represents a virtual eye. Two virtual eyes give a stereo pair. Along each edge of this graph we can instantiate a novel virtual camera using simple linear interpolation of the extrinsic parameters and we can generate its corresponding novel view by using known IBR techniques. This, in practice, extends the domain of the possible views from the discrete set of acquired cameras to a continuous domain given by our graph. Combining any couple of cameras that we can pick on this graph we obtain the set of all possible stereo pairs, that is the codomain of our graph. We give a formal definition of this space, that we called StereoSpace. *Built on this, we designed our prototype system for the stereoscopic exploration of photo collections.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Nowadays, photo collections of objects or locations of interest are very common. These collections can be retrieved from the Internet or generated ad-hoc for several purposes, for example, to obtain a rough 3D reconstruction of the location (or specific object). Due to the explosion of multi-view data, several systems for the exploitation, navigation and exploration of photo collections have been realized in the last few years. Microsoft PhotoSynth (a work derived from PhotoTourism [SSS06]) and PhotoCloud [BBT*13] are two examples. PhotoSynth builds its service on an underlying automatic sparse 3D reconstruction which works on collections of photographs provided by the user. It allows a practical and effective navigation of these image sets arranged spatially according to their relative calibration data. PhotoCloud is somewhat similar but it can visualize both high-quality triangle meshes and points clouds and the photographs registered on this geometric data. Hence, it allows a joint 2D-3D navigation of this data. Another popular example is Google Street View [ADF*10] which works mainly on panoramic images integrated with additional 3D content. This increasing interest has also boosted image-based rendering (IBR) research field which continuously proposes new solutions for the seamless transition between photographs taken at different viewpoints, like [GAF*10], or free viewpoint solution like [CDSHD13]. All these systems provide the possibility to explore in 3D the location/object of interest without the necessity of a full 3D reconstruction. This is a viable alternative for certain applications, as, despite the increased quality of the latest reconstruction techniques, they continue to fail where the input point clouds recovered by SfM methods have very low density thus decreasing the final user experience. Here, we propose a visualization system of this type, which allows the stereoscopic navigation of user photo col-

lections. The proposed system requires a set of calibrated images, as from the output of any SFM algorithm. As a matter of fact, usual navigation interfaces neglect to use the inherent nature of such set of cameras which, despite having a wide-baseline between neighboring cameras, are often well organized. In particular, photographs taken for 3D reconstruction tend to follow circular or semi-circular arcs around the subject, and tend to be taken from an approximately constant height and varying the orientation of the camera smoothly to guarantee a good overlap between photographs. Thus it is quite likely that, among a given collection, we can find certain number of pairs of cameras that could be rectified and used as a stereo pair. Our approach consists in extending this discrete domain of stereo views to a continuous one by employing known IBR techniques to generate novel views. We will call this domain *StereoSpace* and we will define how it can be intuitively browsed using a simple pan/zoom interface to provide a seamless stereoscopic navigation experience. The contribution of this paper is twofold:

- A formal definition of the space of the stereo pairs which can be generated interpolating the actual viewpoints given a set of calibrated cameras called *StereoSpace*.
- The prototype of a visualization system for the stereoscopic exploration of photo collections using stereoscopic devices of any type (e.g. shutter glasses, anaglyphs, etc.) that provides a full tridimensional experience without the need of a complete 3D reconstruction of the location/object of interest.

2. Related Work

In the following we concisely review the literature most closely related to our approach.

Image Based Rendering The literature on IBR is vast. Seminal works such as light fields [LH96] and unstructured lumigraphs [BBM*01] laid the basis foundations for this field. In recent years, two main approaches have been proposed to this problem: proxy-based techniques and, more recently, variational warping methods.

Proxy-based approaches start by reconstructing a proxy geometry, that is a coarse approximation of the real underlying geometry of the acquired object. Given a novel view to be generated, they select a certain number of cameras whose images are projected onto the proxy and then re-projected the proxy onto the the novel view. Usually, four or three of the nearest actual acquired cameras are used and thus the projected images in the novel view needs to be blended. Even in most recent works blending rules are derived by the recipes found in [BBM*01]. Eventually, misalignment due to the coarseness of the proxy or to even small errors in cameras calibration can produce ghosting artifacts that can be corrected by warping the projected images using optical flow [EDDM*08]. In general, reconstructed proxies may miss entire regions of the corresponding images thus leading to poor rendering quality. A workaround to this problem is represented by the Ambient point clouds [GAF*10] which use non photo-realistic rendering to render the transition between images in poorly reconstructed or missing regions.

The most recent approaches to IBR are based on variational warping methods. In these approaches, image warping is based on the sparse correspondences given by the projection of corresponding points generated by multi-view stereo methods on the input images. In particular [ZJM13] used this method for 3D video stabilization while [CSD11] used the same approach for wide-baseline IBR. The essence of these methods is to lay down a regular grid or triangulation over each image. Each vertex of this grid becomes a 2D unknown to be computed in its warped version in the final image. Two energy terms are then imposed: a data term [ZJM13] and a similarity term. In the data term, SFM points are projected onto the grid and their coordinates are expressed by barycentric coordinates, for triangular grids [CSD11], or by the bilinear interpolation of their four surrounding vertices, in the grid case [ZJM13]. Then, the squared distance between the interpolated grid position in the output grid and the projected SFM point in the novel image is minimized. On other proposals, the similarity warp prior is included to minimize local shape distortions of each mesh triangle that essentially can undergo a transformation as close as possible to a similarity transformation. In the wide-baseline case [CSD11], attention must be taken to depth discontinuities which lead to unnatural silhouettes distortions. In [CSD11], this problem is overcome by requiring manual silhouettes annotation and including an ad-hoc energy component which affects only the edges along silhouettes. In a follow up paper Chaurasia et al. [CDSHD13], improve on these limitations by employing a local warping approach which essentially applies the warp to a super-pixel segmentation of the images [ASS*12], warping each individual super-pixel separately.

2D-to-3D conversion IN a way, our work can be classified among the 2D-to-3D conversions technologies used in movies post-productions. The vast majority of these works, however, concern stereoscopic view generation with small baseline (e.g video) while, to our knowledge, no one has developed a system based on wide baselines. In [KKS08], a system for the production of a 3D stereoscopic video from a monocular one is presented and essentially based on their previous stereo view synthesis algorithm developed in [KS07]. In practice, they first use a structure from motion system to recover both camera motion and a sparse point cloud of the filmed scene. Then, after selecting a camera of the sequence they instantiate a virtual camera applying an horizontal offset to it, so creating a virtual stereo rig. Then, the 3D points are projected onto the novel view and to nearby cameras. 2D correspondences between the novel view and the other images are then used to calculate perspective homographies. Each neighboring view is then warped into the virtual stereo frame and blended together to form the final

image. This approach, however, makes the assumption of having a small baseline between cameras and, more radically, it approximates the filmed scene to a planar scene. An alternative technique, frequently used in movies post-production, is to create a dense depth map for each monocular image. The depth map creation process can be either the accurate but manual process of an artist or the automatic creation of "surrogate depth maps" using simple 2D features such as luminance intensity, color distribution or edges [DPPC13]. These methods do not recover the real depth of the scene but they only try to create a depth which is only approximatively consistent, Anyway, such depth maps provide a comfortable user experience. Once a dense depth map is somehow created, it is mapped to a bi-dimensional disparity for each pixel of the original image which is then warped horizontally according to the disparity calculated obtaining the novel view, see, for example, [Kon99] for an in-depth treatment of this method which is usually addressed as "Depth-image-based rendering" (DIBR).

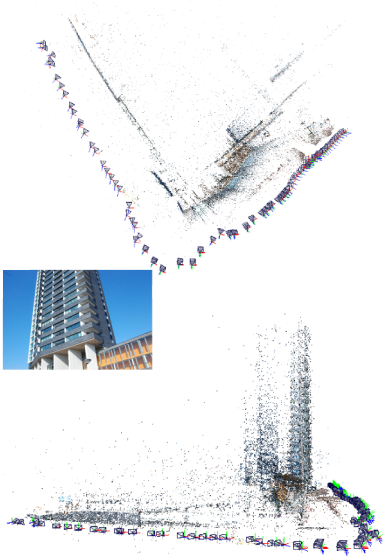


Figure 1: *A typical acquisition pattern for a building.*

3. Our Approach at a glance

The key observation behind our system is that a set of photographs that were taken to make 3D reconstruction is generally well suited to create stereo views. This is because a few typical characteristics of these cameras' position and orientation:

- The cameras tend to focus on the same region.
- Their positions are arranged on arcs surrounding the object.
- The height of each camera is approximatively the same (that is, the height of the person who is holding the camera).

Figure 1 shows these characteristics in a practical example.

Therefore, there is a high number of camera pairs that can be rectified so to create a stereo pair. Our approach consists of extending the domain of point of views from the original set of cameras to a continuous region so that the domain of stereo pairs is also continuous. We call this domain *stereospace* and define how it can be used to provide a seamless stereoscopic browsing experience.

In the following we will illustrate how we generate novel views and then stereoscopic pairs starting from the available wide-baseline views. Then, we formalize better the StereoSpace and how to map it on the proposed navigation system.

3.1. Camera Interpolation

First of all, we use a SfM algorithm [SCD*06] to calibrate the cameras and generate a sparse 3D point cloud of the scene. Then, given a novel view to be generated, we select four of its neighboring cameras. For each of these cameras, we reproject their 3D correspondences onto the novel camera and we warp each original image so that its 2D correspondences are made coincident to the ones onto the novel view. In our current implementation we use [SMW06] for image warping; the algorithm version which applies a similarity deformation. Referring to [SMW06], we can say that given an input image we use its 3D correspondences projected onto its image plane as deformation handles, The warping is obtained simultaneously offset each handle of an amount corresponding to the vector distance between the positions of the 3D correspondences projected onto the input camera image plane and the positions obtained by projecting the same points onto the image plane of novel camera as shown in reffig:warp.

The warped images are then blended using the principles which can be found in [BBM*01], in particular, in this prototype, we use only angular and field-of-view penalties, as shown in Figure 3. More precisely, given an input camera C_i and a novel camera C_n whose respective positions are $\mathbf{C_i}$ and $\mathbf{C_n}$ and given a point $\mathbf{P}$ on the proxy geometry which projects on the input camera image plane I_i at the point $\mathbf{p}(x,y)$, our penalty scheme is the following:

$$P_{ANG}(C_i,x,y) = \arccos((\mathbf{C_n}-\mathbf{P})\cdot(\mathbf{C_i}-\mathbf{P}))$$

$$P_{FOV}(C_i,x,y) = \begin{cases} \infty & if\ \mathbf{p}(x,y)\ lies\ outside\ I_i \\ 0 & otherwise \end{cases}$$

And, finally, the complete penalty is:

$$P = P_{ANG} + P_{FOV}$$

The geometry used to calculate the blending weights is ob-

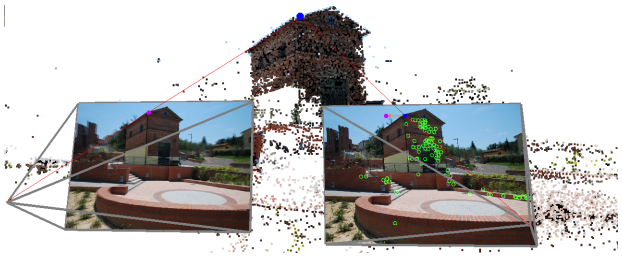


Figure 2: *Reprojection of 3D correspondences(only a subset of all correspondences are shown to improve the readability) from an input camera (right) to the novel view (left). The 2d projections are used as handles for the deformation algorithm. On right camera is shown in blue an handle and the corresponding offset shown as an orange arrow. The handle offset is calculated as the vector distance between the coordinates of same 3D point on the two different image planes.*

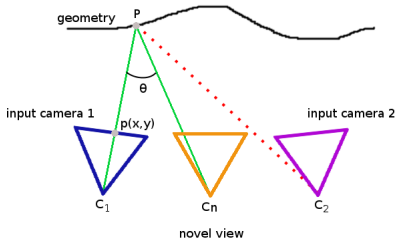


Figure 3: *Angle and field of view penalties employed in the final blending algorithm. The red ray coming from the second input camera is excluded from the final blending as it does not project onto the camera image plane (i.e it is out of its field of view). Instead the green ray is taken into account and its contribution weighted according to the angle theta.*

tained running poisson surface reconstruction [KBH06] on a densified point cloud calculated with PMVS [FP10]. Note that, as also underlined in [CSD11], geometry inaccuracies are well tolerated as only used to compute the blending weights and not for image reprojection as in proxy-based techniques. Our method, is somehow similar to [CSD11] even if we do not take into account the silhouette problem. This component of our prototype can be easily improved replacing it with a more robust IBR technique such as [CDSHD13].

3.2. Camera Rectification

Once two novel views are generated we have to rectify them in order to simulate an horizontal stereo rig. To accomplish this task, we used the rectification algorithm [FTV00] which is a simple and straightforward linear rectification method. In summary, this method uses a standard pinhole camera modeled by its center C and its image plane I located at a distance f from C , where f is the *focal length* of the camera.

Cameras are already calibrated thus we have full knowledge of their *perspective projection camera* [FTV00] which can factorized in:

$$\mathbf{M} = \mathbf{A}[\mathbf{R}|\mathbf{t}]$$

where $\mathbf{A}$ is the matrix of intrinsic parameters defined as:

$$\mathbf{A} = \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

. α_u and α_v are respectively the horizontal and vertical focal lengths in pixels, u_0 and v_0 are the coordinates of *principal point* (i.e the image center) and γ is the skew factor, which we assume to be zero. The camera extrinsic parameters are instead expressed in terms of the 3×3 rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$. Given such a representation, let $\mathbf{P} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ be the homogeneous coordinates of a point defined in the world reference frame, its projection $\mathbf{p} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ onto the camera image plane is obtained simply as:

$$\mathbf{p} = \mathbf{MP}$$

Now, given two cameras whose centers are $\mathbf{C_0}$ and $\mathbf{C_1}$ in homogeneous coordinates, and whose camera matrices are $\mathbf{M_0}$ and $\mathbf{M_1}$, the idea behind the rectification process is to define two novel matrices $\mathbf{M_0^1}$ and $\mathbf{M_1^1}$ which preserve their previous points of view but they are rotated to make the two image planes coplanar and parallel to the baseline C_0C_1 thus ensuring that also the epipolar lines are both parallel and horizontal. Moreover, forcing the cameras intrinsic parameters to be equal, conjugate points lay on the same line in both images having the same vertical coordinates. The new camera matrices $\mathbf{M_0^1}$ and $\mathbf{M_1^1}$ can be written down in terms of their factorization:

$$\mathbf{M_0^1} = \mathbf{A}[\mathbf{R}|-\mathbf{R}\tilde{\mathbf{C_0}}], \mathbf{M_1^1} = \mathbf{A}[\mathbf{R}|-\mathbf{R}\tilde{\mathbf{C_1}}]$$

where $\tilde{\mathbf{C_0}}$ and $\tilde{\mathbf{C_1}}$ are the cartesian components of the centers coordinates and where $\mathbf{R}$ is:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r_1^T} \\ \mathbf{r_2^T} \\ \mathbf{r_3^T} \end{bmatrix}$$

and $\mathbf{r_1^T}, \mathbf{r_2^T}, \mathbf{r_3^T}$ are the new X, Y and Z axes of the camera reference frame calculated as:

1. $\mathbf{r_1} = (\tilde{\mathbf{C_1}} - \tilde{\mathbf{C_0}}) / \|\tilde{\mathbf{C_1}} - \tilde{\mathbf{C_0}}\|$;
2. $\mathbf{r_2} = \mathbf{k} \times \mathbf{r_1}$;
3. $\mathbf{r_3} = \mathbf{r_1} \times \mathbf{r_2}$;

where $\mathbf{k}$ is a unit vector which we take equal to the old Z axis of the left camera. Let's now calculate the mapping between the image plane of the old and new (rectified) left camera, whose matrices are respectively $\mathbf{M_0}$ and $\mathbf{M_0^1}$. For each 3D point $\mathbf{P}$ we have:

$$\begin{cases} \mathbf{p_0} = \mathbf{M_0 P} \\ \mathbf{p_1} = \mathbf{M_0^1 P} \end{cases}$$

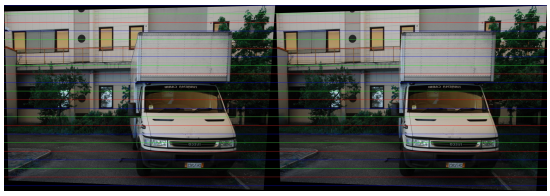


Figure 4: *An example of rectified stereo pair. Some epipolar lines are drawn on top of the two images in different colors.*

As the rectification process does not change the point of view of the camera we can write down the optical rays equations as:

$$\begin{cases} \tilde{\mathbf{P}} = \tilde{\mathbf{C}} + \lambda_0 \mathbf{Q_0}^{-1} \mathbf{p_0} & \lambda_0 \in \mathbb{R} \\ \tilde{\mathbf{P}} = \tilde{\mathbf{C}} + \lambda_1 \mathbf{Q_1}^{-1} \mathbf{p_1} & \lambda_1 \in \mathbb{R} \end{cases}$$

where, for example, $\mathbf{Q_0}$ is the 3x3 sub-matrix of $\mathbf{M_0}$, and in the end we have:

$$\mathbf{p_1} = \mathbf{Q_1}\mathbf{Q_0}^{-1}\mathbf{p_0}$$

where $\mathbf{H} = \mathbf{Q_1}\mathbf{Q_0}^{-1}$ is the mapping we were looking for, that is the homography transformation between the two image planes. A more in-depth treatment of these topics can also be found in [Fus08]. See Figure 4 for an example of the output of the rectification algorithm.

4. StereoSpace: the domain of stereo views

A stereo pair is defined by a *view position*, that is the mid point between the camera pair, an *interpupillary distance*, that is the distance between the camera pair, and a *view orientation*, which is obtained as shown in the previous section.

Since here we want to define a browsing space where the user can change the point of view and the interpupillary distance, we can identify a stereo pair just from the positions of the camera pair. Therefore each camera pair map to a stereo view as:

$$s(a,b) = ((a+b)/2, \|a-b\|)$$

where $a,b \in \mathbf{R}^3$ are the camera positions and $s \in \mathbf{R}^4$ is the position of the stereo camera enriched with the interpupillary distance. Let be $\mathcal{C}$ the set of camera positions, we define the *SteroSpace* as the codomain of function s:

$$S(\mathcal{C}) = \{s(a,b) | a,b \in \mathcal{C}\} \quad (1)$$

In this simplest setting the resulting stereospace is a set of points. Figure 5 shows a simple example where 4 cameras (in black) are paired (connected by a segment in the figure) to make 6 stereo cameras (in red). In this case the only browsing modality for the user would be to jump from one fixed stereo view to another.

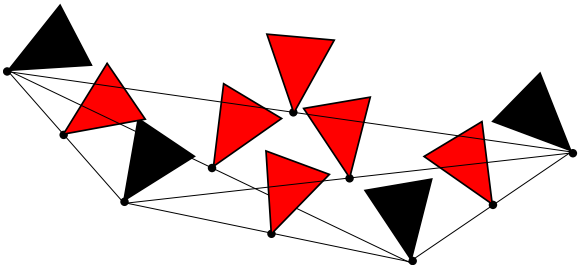


Figure 5: *Four cameras (in black) and the corresponding 6 stereo pairs (in red).*

By applying camera interpolation as described in Section 3.1, the space of cameras is extended from a set of point to a set of segments $=\{s_0,\ldots,s_m\}$ where each segment $s = \overline{ab}$ connects two cameras of the original dataset (we can include in $\mathcal{C}$ the original camera positions by considering degenerate segments $s = \overline{aa}$). In this case the corresponding stereospace become piecewise countinuous. Let us start by considering the stereospace corresponding to a pair of segments s_i and s_j (please refer to Figure 6). As a simplification assumption (that can be removed later) and for illustration purposes let us assume that the original cameras are in a common plane, say XZ, so that we can scale the dimension by one and map the interpupillary distance on the Z axis. The position of the stereo camera is mapped on the XY plane (because it is the halfway point between two points in that plane) and the interpupillary distance on the Z axis. The resulting stereospace is the union of three continous regions:

$$\begin{aligned} S(\{s_i,s_j\}) = &\{s(a,b) | a \in s_i, b \in s_j\} \cup \\ &\{s(a,b) | a,b \in s_i\} \cup \\ &\{s(a,b) | a,b \in s_j\} \end{aligned}$$

The portion of stereospace generated by pair of points belonging to different segments is a rhomboid-like shape, while if they belong to same segment the regions are triangular (which are actually folded rhombi). Note that the projection on the XY plane of these regions is not other the the Minkowski sum of the segments obtained by halving the coordinates of s_i and s_j.
Figure 7 shows the stereospace for a set of segments connected to form a sequence of polylines. This is a typical situation we have for cameras following a reconstruction-driven pattern. For the sake of illustration, we used different colors for subregions generated by different segments within the same polyline.

4.1. Stereo Browsing with the Stereospace

One way to move inside the stereospace would be just to visualize its geometric representation in a separate window, let the user click on a point and the the view to the corresponding stereo camera. However we can provide a much

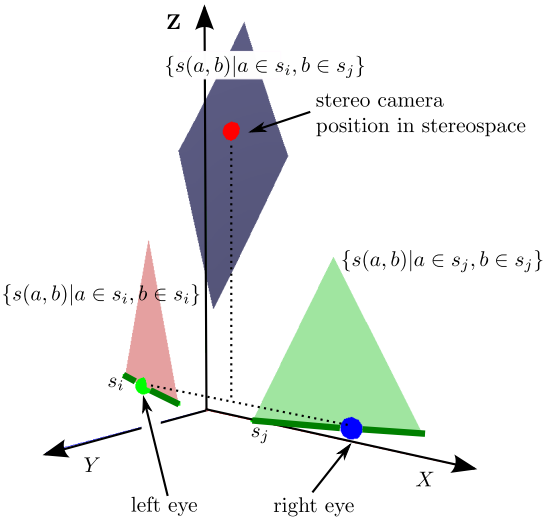


Figure 6: *Stereospace for two segments.*

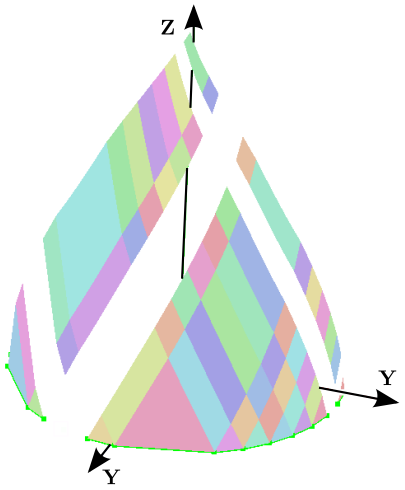


Figure 7: *Stereospace for polylines.*

more natural interaction. Let p be the current viewer position in the stereospace. It is clear by definition that we move from point p to a point with a greater z component we will increase the interpupillary distance, which corresponds to scaling down the model and bringing it closer to the viewer, something that we improperly call zoom-in in our interface. Conversely, moving to a point with lower z-value means to decrease the interpupillary distance, that is, scaling up and bringing the model farther away (zoom-out).
If we move to a point in the same XY plane (that is, leaving the z component unchanged) there will be no zooming involved. In the typical configuration of camera positions this horizontal movement will correspond to a left-right pan or to an horizontal orbit.

$\mathcal{C}$ can be easily parametrized with the index of the segment i and the linear interpolation coefficient beteen the endpoints λ_i:

$$c(i,\lambda_i) = s_{i0}\ (1-\lambda_i) + s_{i1}\ \lambda_i : i \in [0\ldots m], 0 \leq \lambda_i \leq 1$$

where $\overline{s_{i0}\ s_{i1}}$ is the segment i (we will just indicate (i,λ_i) with λ_i from now on).

A point in stereospace is then defined as:

$$s(\lambda_i,\lambda_j) = \begin{bmatrix} \frac{1}{2}(c(\lambda_i)+c(\lambda_j)) \\ \|c(\lambda_i)-c(\lambda_j)\| \end{bmatrix}$$

The gradient of $\bigtriangledown s(\lambda_i,\lambda_j)_z$ tells us the direction of maximum increment of p_z (for the zoom-int/zoom-out movement), while the tangential direction $T(\lambda_i,\lambda_j) : T \cdot \bigtriangledown s(\lambda_i,\lambda_j)_z = 0$ is the direction where no zoom takes place (see Figure 8(Top)).

Thus we can map the user commands to movements in parameteric space as:

$$zoom(v) \rightarrow [\lambda_i,\lambda_j]^T += v\ \bigtriangledown s(\lambda_i,\lambda_j)_z$$
$$pan(v) \rightarrow [\lambda_i,\lambda_j]^T += v\ T$$

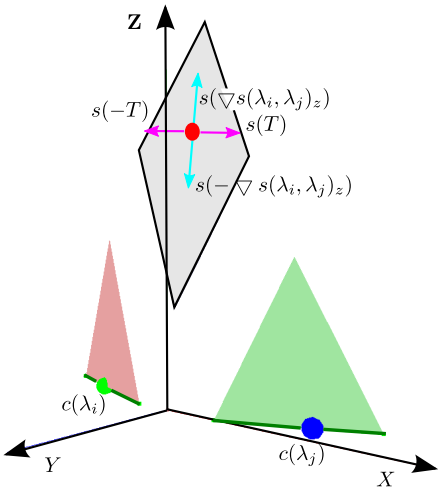
where v is the amount of movement (positive or negative). In this way the user can move with two degrees of freedom and the position is updated to the *best* position in stereospace. Whith this mapping we provided the way to smoothly change position and zoom within a continuous region of the stereospace. When the current position is on the border of a region of the stereospace we jump to a neighbor region in the direction of movement. In other words, if the user is zooming-in, that is increasing the z component in sterespace, we will look if there is a region of the stereo space above the current one and so on.

Please note that this time the direction is expressed in stereospace and it is the mapping of the moving direction in parametric space. that is:

$$d = s(\lambda_{dir})$$

We jump to the closer point on the stereospace which is in the cone with apex in the current position, oriented as d and with angle 40°. In our implementation this is done by creating tessellated surfaces for each region and inserting them in a search data structure.

When the position in stereospace is update to a specific point we are left with the problem of findding its projection in parametric space, that is, the λs. This only happens because we have to jump from one point to another of the stereospace directly, instead of changing the position in parametric space.



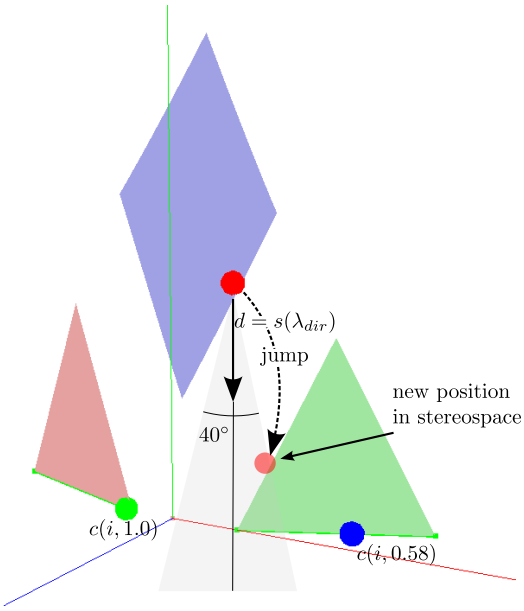


Figure 8: *Moving in the stereospace.*

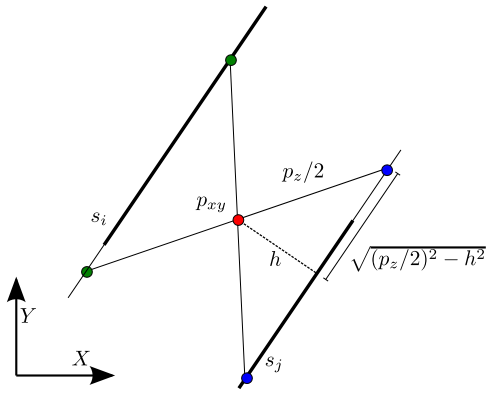


Figure 9: *Mapping to C for parallel segments.*

4.2. From Stereospace to parametric space

The function $s : \mathcal{C} \times \mathcal{C} \rightarrow S$ is not injective so there is not s^{-1}. However we can still find a mapping from stereospace to parametric space.

There are two cases in which a point in stereospace may correspond to more than one point in parametric space.
The first case is because two regions, corresponding to difference couples of segmements, intersect each other. When we jump from a position to a point which lies in this intersection, we simply choose to map in the first region we find. This is easily done by storing in the geometric representation of the region, two references to the segments that generate it.
The second case is when a region if generated by two parallel segments. This case happens for all the triangular regions generated by a single segment (that is, by two istances of it).

Let us consider the point p in the region region generated by s_i and s_j:

$$p_{xy}(\lambda_i,\lambda_j) = s_{i0_{xy}}\,(1-\lambda_i) + s_{i1_{xy}}\,\lambda_i + s_{j0_{xy}}\,(1-\lambda_j) + s_{j1_{xy}}\,\lambda_j$$

solving for λ_i, λ_j leads to a simple system:

$$\begin{bmatrix} s_{i1_{xy}} - s_{i0_{xy}} & | & s_{j1_{xy}} - s_{j0_{xy}} \end{bmatrix} \begin{bmatrix} \lambda_i \\ \lambda_j \end{bmatrix} = p_{xy} - \begin{bmatrix} s_{i0_{xy}} + s_{j0_{xy}} \end{bmatrix}$$

If the matrix is non singular we can solve the system and have the λs, otherwise we are in the case where the two segments are parallel. In this case we consider the projection of p on the plane XY and then look for points on the segments at a distance $p_z/2$ from p_{xy} (see Figure 4.2), which can be easily done by computing the distance from each segment and using the Pythagorean theorem. Note that we may have two solutions from which we will choose the more coherent with the rectified camera orientation.

4.3. The proposed system

We now illustrate our prototype system in particular its user interface. In Figure 10 the main window of the system is shown along with an automatically generated navigation map which can be activated/deactivated on the top-left of the main window. The user can simply navigate the acquired location by panning and zooming with the mouse interface, as described in Section 4.1. This map is provided to improve the sense of orientation of the user since there is not any direct user interaction with the map.

4.3.1. Navigation map

The map shown in Figure 11 represents the acquired scene with an image as an impostor for its real geometry. Then,

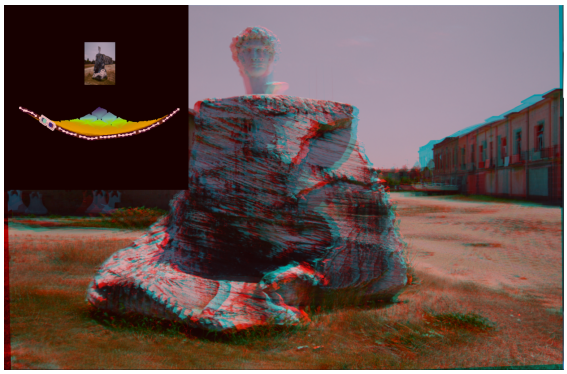


Figure 10: *An image of our prototype stereo browsing system*

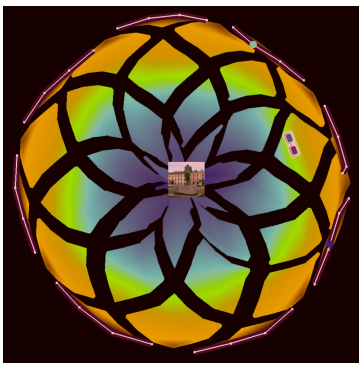


Figure 11: *An image of our automatically generated interface*

the polylines connecting the calibrated cameras are drawn to show the camera path around the subject/location of interest. The location of the stereo camera and of both left and right eyes are represented respectively with a small image of anaglyph glasses and with one colored sphere for each eye. In order to represent the StereoSpace, we render it with an orthogonal camera from above, with a color per vertex corresponding to the maximum quality for that point (see Section 4.3.2). When the user pan or zoom both the stereo camera icon and the eyes moves accordingly on the map. Figure 12, shows the interface while the user is panning, while in Figure 13 the case of a zoom interaction is shown. Finally, Figure 14 shows the case of a jump (while the user was zooming) inside the StereoSpace and its corresponding representation in our interface.

4.3.2. Quality of Stereo Views

Both vergence and stereopsis are binocular depth cues, e.g they depend on the relationship between the two eyes and between the two images formed onto both retinas (please refer to [How02] for a comprehensive treatment of these top-

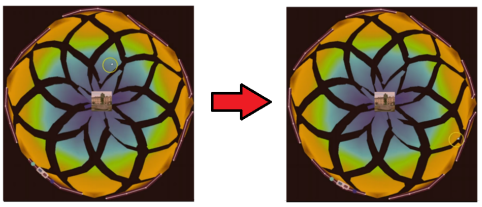


Figure 12: *Panning inside the StereoSpace as rendered in the navigation map.*

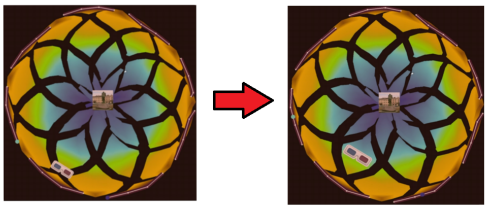


Figure 13: *Zooming inside the StereoSpace as rendered in the navigation map.*

ics). Human vergence is essentially a triangulation system in which the lines of sight of the two eyes intersecting at some environmental location calculates its depth basing on interpupillary distance and the angle between the two line of sights [TFCRS11]. On the other side, disparity concerns the analysis of the offset in position of corresponding points between the left and right eyes and is considered a relative depth cue. More specifically, disparity measures depth relative to the *horopter*, which can be defined as the locus of points which project onto the same positions in both eyes, that is, those points for which no disparity is perceived, and that, theoretically speaking, can be roughly seen as a circle passing through the centers of the lenses of the two eyes and the fixation-point. The human visual system has a characteristic range of retinal disparity around the horopter inside

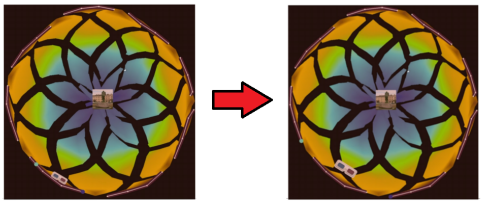


Figure 14: *Jumping inside the StereoSpace as rendered in the navigation map. Note that the anaglyph glasses icon is also scaled proportionally to the interpupillary distance.*

which the left and right image can be fused effectively. If disparity values go outside this range, called Panum's fusional area, double vision (diplopia) occurs. Moreover, not all disparity can be fused comfortably [LFHI09], thus common applications like 3DTV and 3D cinema try to limit disparities values inside the so called *Parceval's zone of comfort* (which is about a third of the whole fusible range), for example, also using also post-production methods like [LHW*10].

We defined the StereoSpace as the region of space from which we can build a stereo pair. However, not all stereo pairs are alike. The quality of a stereo pair is related to several factors: the quality of the interpolation between cameras, the quality of the rectification and, as just stated, the capability of the user to fuse the two views in a comfortable way. To account for these problems, we enrich the description of the StereoSpace with a *quality* value. This value can be computed in several ways from simple heuristics to more complex models. Once calculated, the quality can be stored in each vertex of the StereoSpace representation helping the user to avoid low quality regions of the StereoSpace. A pratical way of doing this would be to incorporate a system for cutting the StereoSpace with several clipping planes which would remove entire regions or parts of them where an uncomfortable stereoscopic vision would occur. We leave this improvement to our prototype system as a future work.

5. Results

We now show some examples on which we have tested our prototype system. From Figure 15 to Figure 18 a gallery of anaglyphs is shown. Note that each stereo pair is generated using our interpolation scheme, thus left and right eye are both interpolated images. As it is clearly visible, despite the simplicity of IBR approach employed, the resulting stereo pairs are quite effective and minimal artifacts, such as silhouettes deformations, are well tolerated, especially in this case of stereoscopic views, as also noted in [LHW*10]. See Figure 17 and Figure 18 for two examples of a panning and a zooming sequence.

6. Conclusion

In this paper we have presented a new prototype system for the stereoscopic navigation of photo collections, particularly effective to navigate images where calibration data are available or can be calculated. The system is able to provide a 3D navigation of the scene without requiring a full 3D surface reconstruction of it. A well-defined set of generated stereo pairs, called StereoSpace, allow us to limit the number of generated views such that a simple rectification of newly interpolated images is able to provide good results. Some aspects have to improved, for example a better in-between views generation and a better evaluation of the quality of the stereo pair, but, as shown, the current implementation is already effective.

Figure 15: *Puppets scene.*

Figure 16: *Church scene.*

References

[ADF*10] ANGUELOV D., DULONG C., FILIP D., FRUEH C., LAFON S., LYON R., OGALE A., VINCENT L., WEAVER J.: Google street view: Capturing the world at street level. *Computer 43*, 6 (2010), 32–38. 1

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SUSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 34*, 11 (2012), 2274–2282. 2

[BBM*01] BUEHLER C., BOSSE M., MCMILLAN L., GORTLER S., COHEN M.: Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 425–432. 2, 3

[BBT*13] BRIVIO P., BENEDETTI L., TARINI M., PONCHIO F., CIGNONI P., SCOPIGNO R.: Photocloud: Interactive remote exploration of joint 2d and 3d datasets. *Computer Graphics and Applications, IEEE 33*, 2 (2013), 86–96. 1

[CDSHD13] CHAURASIA G., DUCHÊNE S., SORKINE-HORNUNG O., DRETTAKIS G.: Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics 32*, 3 (2013), 30:1–30:12. 1, 2, 4

[CSD11] CHAURASIA G., SORKINE O., DRETTAKIS G.: Silhouette-aware warping for image-based rendering. *Computer Graphics Forum (Proceedings of the EUROGRAPHICS Symposium on Rendering) 30*, 4 (2011), 1223âĂŞ–1232. URL: http://www-sop.inria.fr/reves/Basilic/2011/CSD11. 2, 4

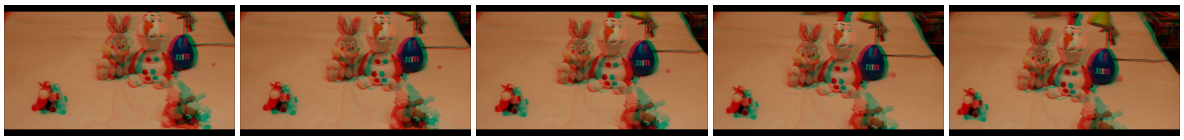


Figure 17: *A panning sequence of the Puppets scene.*

Figure 18: *A zooming sequence of the Camion scene. Note how the tree in the background starts from an almost flat appearance to a final one where is detached from the wall behind.*

[DPPC13] DUFAUX F., PESQUET-POPESCU B., CAGNAZZO M.: *Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering*. John Wiley & Sons, 2013. 3

[EDDM*08] EISEMANN M., DE DECKER B., MAGNOR M., BEKAERT P., DE AGUIAR E., AHMED N., THEOBALT C., SELLENT A.: Floating textures. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 409–418. 2

[FP10] FURUKAWA Y., PONCE J.: Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32*, 8 (2010), 1362–1376. 4

[FTV00] FUSIELLO A., TRUCCO E., VERRI A.: A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications 12*, 1 (2000), 16–22. 4

[Fus08] FUSIELLO A.: Visione computazionale. *Appunti delle lezioni. Pubblicato a cura dellâĂŹautore* (2008). 5

[GAF*10] GOESELE M., ACKERMANN J., FUHRMANN S., HAUBOLD C., KLOWSKY R.: Ambient point clouds for view interpolation. *ACM Transactions on Graphics (TOG) 29*, 4 (2010), 95. 1, 2

[How02] HOWARD I. P.: *Seeing in depth, Vol. 1: Basic mechanisms.* University of Toronto Press, 2002. 8

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing* (2006). 4

[KKS08] KNORR S., KUNTER M., SIKORA T.: Stereoscopic 3d from 2d video with super-resolution capability. *Signal Processing: Image Communication 23*, 9 (2008), 665–676. 2

[Kon99] KONRAD J.: View reconstruction for 3-d video entertainment: issues, algorithms and applications. In *Image Processing And Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465)* (1999), vol. 1, IET, pp. 8–12. 3

[KS07] KNORR S., SIKORA T.: An image-based rendering (ibr) approach for realistic stereo view synthesis of tv broadcast based on structure from motion. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on* (2007), vol. 6, IEEE, pp. VI–572. 2

[LFHI09] LAMBOOIJ M., FORTUIN M., HEYNDERICKX I., IJSSELSTEIJN W.: Visual discomfort and visual fatigue of stereoscopic displays: a review. *Journal of Imaging Science and Technology 53*, 3 (2009), 30201–1. 9

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 31–42. 2

[LHW*10] LANG M., HORNUNG A., WANG O., POULAKOS S., SMOLIC A., GROSS M.: Nonlinear disparity mapping for stereoscopic 3d. *ACM Transactions on Graphics (TOG) 29*, 4 (2010), 75. 9

[SCD*06] SEITZ S. M., CURLESS B., DIEBEL J., SCHARSTEIN D., SZELISKI R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 1, IEEE, pp. 519–528. 3

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. In *ACM Transactions on Graphics (TOG)* (2006), vol. 25, ACM, pp. 533–540. 3

[SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: exploring photo collections in 3d. *ACM transactions on graphics (TOG) 25*, 3 (2006), 835–846. 1

[TFCRS11] THOMPSON W., FLEMING R., CREEM-REGEHR S., STEFANUCCI J. K.: *Visual perception from a computer graphics perspective*. CRC Press, 2011. 8

[ZJM13] ZHOU Z., JIN H., MA Y.: Plane-based content preserving warps for video stabilization. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on* (2013), IEEE, pp. 2299–2306. 2