PhotoPath: Single Image Path Depictions from Multiple Photographs

Christopher Schwartz, Ruwen Schnabel, Patrick Degener, Reinhard Klein
Institute for Computer Science II, University of Bonn

ABSTRACT

In this paper we present PhotoPath, an image synthesis technique that allows the visualization of paths through real world environments in a single intuitive picture. Given a series of input photographs taken with a hand-held camera along the path, our method generates an image that creates the impression of looking along the path all the way to the destination regardless of any curves or corners in the route. Rendered from a pedestrian perspective, this visualization supports intuitive wayfinding and orientation while avoiding occlusion of path segments in the image.
Our approach intentionally avoids an involved and error-prone scene reconstruction. Instead we advocate the use of planar geometry proxies estimated from the sparse point-cloud obtained with Structure-from-Motion. The proxy geometry is spatially deformed in order to align the curved path with the straight viewing direction of the synthesized view. Finally, we propose a novel image composition algorithm accommodating for parallax and occlusion artifacts due to the approximation errors of the actual scene geometry.

Keywords: image-based-rendering, navigation, non-photorealistic-rendering, space-deformation

1 INTRODUCTION

To ease wayfinding in unknown environments, Degener et al. [7] recently proposed an efficient visualization for short paths as they are typically encountered public places like hotels, airports or museums. Their method intuitively conveys directions in a single 'warped' image of the path that gives the impression of looking along the path all the way to the destination. Unlike traditional floor plans, warped images show the path from the visitor's perspective and thus contain realistic portrayals of landmarks which enables intuitive orientation and wayfinding. In contrast to conventional photographs in which segments of the path that lie around a corner usually are occluded, the visibility of the entire path is ensured by a space deformation that aligns the route with the viewing direction. Degener et al. demonstrate the effectiveness of their visualization in a comparative user-study. An example of a warped image is shown in Figure 1. Such images are valuable and inexpensive navigation aids that can be handed out to visitors. However, the approach of Degener et al. suffers from a severe practical limitation: It supposes that a fully modeled, high quality textured 3D model of the scene is already given which, naturally, seldom is the case. To overcome this hurdle we want to use nothing

Figure 1: Given a series of input photographs and a sparse 3D point-cloud created by a Structure-from-Motion process our method generates a single image that summarizes the views along a user specified path (The path points are depicted in blue in the middle). In the synthesized image curves and corners in the path are straightened such that the impression is created of looking along the path all the way to the destination. Beside a use as navigational aid such images also possess a certain artistic appeal.

Plzen, Czech Republic.

but a series of photographs taken along the path with an ordinary hand-held camera.

Unfortunately, the extraction of accurate 3D geometry from photographs is a difficult task. Even though a set of photographs could in principle suffice for a full-scale multiview reconstruction [13, 20] this is relatively unreliable in general. In particular dense stereo matching is error prone if applied to images with a large baseline and many specular surfaces. However, recently approaches such as Photo Tourism [17] and [16] have demonstrated that Structure-from-Motion (SfM) can be used to robustly extract a sparse 3D point-cloud from a set of unordered photographs with relatively large baselines. The question now is if this sparse 3D information already suffices to generate warped images.

In this work we present PhotoPath that, rather than relying on a full-scale reconstruction, has at its heart a novel image-based rendering technique that is based on a sparse set of 3D-points reconstructed from image features in a SfM step.

Our approach is based on a coarse, not necessarily globally consistent, proxy geometry for the scene. As typical paths in urban or indoor environments are strongly dominated by planar geometry (facades, floors, walls or ceilings), our method simply fits a set of planar proxies to the sparse 3D point cloud. To avoid unmanageable occlusion issues, we introduce an innovative new technique for finding suitable boundaries to the potentially infinite planar proxies, based on a energy minimization in the space of the input images. Obviously however, an inaccurate proxy geometry as we advocate poses challenges for the subsequent image-based rendering due to frequently occurring incorrect occlusion and parallax effects. We demonstrate that these issues can be addressed effectively with our novel image stitching method that minimizes parallax and occlusion artifacts and at the same time is powerful enough to allow for the space deformation necessary for straightening the path. While our algorithm produces good results fully automatically – except for a few parameter settings – we also allow for interactive user intervention in all steps of the method to let the user further improve the final result.

2 RELATED WORK

Image Based Rendering (IBR) methods that create novel views from a set of input images have a long tradition in computer graphics. Individual approaches differ mostly in the richness of the underlying geometry representation. In general, to achieve a given rendering quality IBR methods have to make a trade-off between the required number of input images and the geometric detail of the employed proxy object. While the paper of Levoy and Hanrahan [11], proposing Light Field Rendering, was purely image based using no notion of geometry at all, this comes at the expense of a high number of required input views. The Lumigraph system [10] demonstrated that rendering quality drastically improves if a simple 3D proxy for the scene geometry is used. At the other extreme is view dependent texture mapping as used by Debevec et al. [6] in the Facade system: Starting from simple architectural models, Debevec et al. combine projections of images taken from different viewpoints into textures to increase realism. A generalization of these approaches was given by Buehler et al. [4], who presented an algorithm for unstructured lumigraph rendering that gives compelling renderings even for a sparse set of input images if a detailed proxy surface mesh approximating the scene is given. If depth maps are available for each image, it is also possible to apply classical image warping techniques [5, 12] to obtain novel views. If the density of ray sampling or the accuracy of the geometric proxy is insufficient, as in our case, all of the above IBR methods are prone to ghosting artifacts. As shown by Eisemann et al. [9] these ghosting artifacts can be reduced by warping projected images on the proxy. However, they assume that visibility is only affected by small geometry errors, whereas in our case, no reliable visibility function can be derived at all.

Our method can also be seen as an image-based rendering with a non-standard multi-perspective camera. In this area, the work of Agarwala et al. [1] [2] is related to our approach in two respects. In [2] they introduced user controllable Markov Random Field (MRF) optimization in the context of image stitching and in [1] they used a single proxy plane for composing a series of photographs into a multi-viewpoint panorama that can capture e.g. the side of a street in a single image. Our method is based on the same optimization principle and allows for user intervention in a similar manner but uses an objective function specifically designed for our scenario.

For an in-depth discussion of related work concerning Route Visualization for Navigation, we refer the interested reader to the paper of Degener et al. [7]. Let us here only shortly mention that for successful navigation a correspondence between map and environment has to be established by the observer. Humans perform this task by identifying and matching landmarks in a scene (e.g. walls, corners, doors) with their map representation [14]. This is greatly facilitated by our composite images since they contain real-world depictions of all these features and are therefore easily understood and memorized (the slight distortion in the image is usually not an issue for a human observer).

3 OVERVIEW

Our method requires an input set of photographs $\mathscr{I} = \{I_1, \ldots, I_N\}$ which have been registered by a SfM process such that every picture I_i is associated to a camera

C_i with respective orientation matrix R_i and projection center t_i. Moreover the SfM also supplies a sparse 3D point-cloud $\mathscr{P} = \{p_1, \ldots, p_M\}$ constructed from corresponding image features in the input images.

Given the user supplied path $\mathscr{X}$ defined by a sequence of points $\mathscr{X} = \{x_1, \ldots, x_Q\}$ it is the goal of our algorithm to synthesize a single image, which we call path-image, from the input photographs in which the entire path, i.e. from x_1 to x_Q, is visible and the surroundings remain easily recognizable.

In order to create the path-image, we use $\mathscr{P}$ to estimate a set of planes $\mathscr{A} = \{a_1, \ldots, a_S\}$ that serves as a rough approximation to the real-world geometry. The planes in $\mathscr{A}$ are in general unbounded, i.e. of infinite extent. If such unbounded planes were used as geometric proxies this would result in an unmanageable amount of occlusion errors during image based rendering since planes from very distant parts of the scene easily extent into the current view. It is therefore crucial that bounded proxies are generated in order to obtain approximate visibility information. To this end each input image I_i is partitioned into disjoint regions $I_i = \bigcup_j T_j^i$, where a region T_j^i is assigned as projective texture to a plane $a_j \in \mathscr{A}$ respectively. Thus, each such region results in a bounded, projectively textured 3D-plane $a_j^i \in \mathscr{B}$ which will be used for composing the final path depiction (see Fig. 2). This way occlusion errors are dramatically reduced and it is now feasible to resolve the remaining inaccuracies with our image stitching approach.

Before the path-image is composed, a non-linear space deformation ϕ is applied to the set of bounded proxy planes $\mathscr{B}$. Given the user specified path $\mathscr{X}$ defined by a sequence of points $\mathscr{X} = \{x_1, \ldots, x_Q\}$, ϕ is designed to align the points in $\mathscr{X}$ with the viewing direction, i.e. the Z-axis. The final image I is then composed from the images of the deformed proxies $P(\phi(\mathscr{B})) = \{P(\phi(a_j^i))\}$ under a standard perspective projection P. The composition is based on MRF optimization and selects for each pixel in I the most suitable proxy according to our novel objective function.

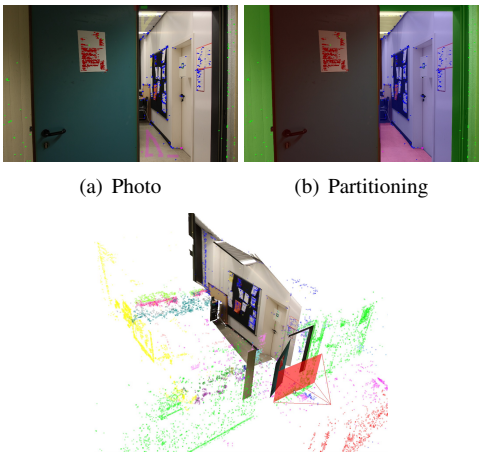
3.1 Preprocessing

We use the publicly available 'Bundler' SfM system [18] for computation of the sparse point-cloud $\mathscr{P}$ and the camera parameters and orientations.

3.2 Proxy Geometry

Given the sparse point-cloud $\mathscr{P}$ our aim is to find a set of planes $\mathscr{A}$ that roughly approximates the underlying geometry. Our algorithm is based on a RANSAC procedure that greedily extracts best fitting planes from $\mathscr{P}$ (see e.g. [15]). To this end the user has to define two parameters: (1) A tolerance value ε that defines the maximal point-to-plane distance allowed for a point to be considered fitted by the plane and (2) a minimum number of fitted points for a plane to be considered valid. Since the user has some knowledge about the scene, e.g. relative object dimensions, it is usually possible to find suitable parameter settings very quickly. The plane detection is fast enough (about a second) so that parameter settings can be evaluated and adjusted interactively. Due to the error tolerance of the following steps the planar approximation need not be very exact, in particular not every point in $\mathscr{P}$ has to be approximated by a planar proxy. Indeed, ignoring clusters of outliers or misaligned parts in $\mathscr{P}$ may often improve the final result.

Thus, after the plane detection $\mathscr{P}$ is partitioned into disjoint point sets $\mathscr{P}_{a_j} \subset \mathscr{P}$ corresponding to the plane $a_j \in \mathscr{A}$ respectively and a set $\mathscr{R}$ of remaining points, i.e. $\mathscr{P} = (\bigcup_{a_j \in \mathscr{A}} \mathscr{P}_{a_j}) \bigcup \mathscr{R}$.



(a) Photo (b) Partitioning

(c) Textured proxies

Figure 2: Semi-automatic partitioning of an input photograph. (a) The input photograph with seed pixels depicted as small colored pixels. Additionally the user has marked some regions to aide the partitioning process (shown as transparently colored pixels). (b) The resulting partitioning. (c) The partitioned imaged parts are projected onto the respective bounded proxies which are visible from the input image. Points in $\mathscr{P}$ corresponding to different planar proxies are rendered in different colors respectively. Please note, that user guidance is usually not required and the partitioning can run fully automatically.

4 INPUT IMAGE PARTITIONING

The proxy geometry $\mathscr{A}$ consists of a set of planes of which each has principally infinite extent. However, in order to generate the final image, the planes need to be bounded. In this work, rather than constructing a globally consistent set of bounded planes, we propose to perform a partitioning of each input image into (necessarily finite) parts corresponding to planes in $\mathscr{A}$. While our approach does neither guarantee a correct partitioning nor consistency between images, such errors are handled later in the image composition optimization. In fact, this simplicity is an important ingredient to the

practicability of our approach as usually quite involved algorithms (which in general nonetheless cannot guarantee a correct solution) are required in order to derive a globally consistent reconstruction.

Since, by principle, every $p \in \mathscr{P}$ originates from featurepoints in at least two different images and every image supports at least five points in $\mathscr{P}$, we can identify subsets $\mathscr{P}^i \subset \mathscr{P}$ of points which originate from featurepoints in image I_i respectively. To find a partitioning $I_i = \bigcup_j T_j^i$ for image I_i the points in $\mathscr{P}^i$ are projected into I_i. The projected locations of points in $\mathscr{P}_{a_j}^i = \mathscr{P}^i \bigcap \mathscr{P}_{a_j}$ serve as seeds for regions T_j^i in I_i corresponding to plane a_j. In order to find the regions we make use of a simple yet surprisingly effective heuristic: Since image edges often correlate with discontinuities in the scene geometry, we try to align the region borders with image edges and assert at the same time that the region T_j^i contains all the projected points from $P_i(\mathscr{P}_{a_j}^i)$. These conditions are easily captured in a MRF-like energy of the following form:

$$E(\mathscr{L}) = \sum_{x \in I} D_x(\mathscr{L}(x)) + \lambda \sum_{(x,y) \in \mathscr{N}} S_{xy}(\mathscr{L}(x), \mathscr{L}(y)) \tag{1}$$

where $\mathscr{L} : I \to \mathscr{A}$ is a labeling assigning planes to pixels, x and y are pixels in I and $\mathscr{N} \subset I \otimes I$ is the set of neighboring pixels in I. The term $D_x(\mathscr{L}(x))$ gives the cost of assigning plane $\mathscr{L}(x)$ to pixel x while the term $S_{xy}(\mathscr{L}(x), \mathscr{L}(y))$ describes the relation between neighboring pixels and gives the cost of simultaneously assigning $\mathscr{L}(x)$ to x and $\mathscr{L}(y)$ to y. The parameter λ introduces a weighting of the two energy-terms. Finding the regions T_j^i then amounts to finding a labeling $\mathscr{L}^*$ that minimizes (1).

For the partitioning D_x is used to introduce the seed locations for the regions, i.e. it encourages assignment of label a_j at projected locations of $\mathscr{P}_{a_j}^i$. Thus we define

$$D_x(a_j) = \begin{cases} 0 & x \notin \bigcup_{a_k \in \mathscr{A}} P(\mathscr{P}_{a_k}^i) \vee x \in P(\mathscr{P}_{a_j}^i) \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

In order to align region boundaries with image edges we set S_{xy} as follows:

$$S_{xy}(a_j, a_k) = \begin{cases} 0 & a_j = a_k \\ g(\|\nabla I\|) & a_j \neq a_k \end{cases} \tag{3}$$

where ∇I is the image gradient and $g(x)$ is a function approaching zero for larger values. In our experiments we use a four-connected neighborhood and a weighting factor of $\lambda = 50$. As in our case $\|\nabla I\|$ only takes values in $[0,1]$, the simple choice of $g(x) = 1.0 - x$ is sufficient.

To find $\mathscr{L}^*$ we employ standard MRF optimization techniques (see e.g. [19] and [3]). Since every image is treated independently the image partitioning can be performed in parallel straightforwardly such that we are able to process even a large number of images in a matter of a few minutes.

The effectiveness of our approach is best understood in conjunction with the nature of the SfM process. Since $\mathscr{P}$ is derived from image feature correspondences it naturally contains more points in textured areas or near image discontinuities. Thus, while our simple heuristic would generally fail in textured areas due to the abundance of image edges, the additional seed points constrain the solution in a sensible manner. Conversely, in uniform image regions the number of seed points is low, but so is the number of image edges and these edges are much more likely to coincide with actual discontinuities in the geometry. Additionally we give the user the opportunity to influence the partitioning by marking regions of images that should belong to a specified proxy plane a_j, see Fig. 2. The marked regions are simply interpreted as additional seeds by our system.

After the partitioning we create the final geometric proxies on a per-image basis. Each region T_j^i is triangulated in the image domain and projected onto a_j to give a bounded planar approximation $a_j^i \in \mathscr{B}$ which is projectively textured from I_i.

5 SPACE DEFORMATION

In this work we employ the space deformation introduced by Degener et al. [7] which is responsible for the straightening of the path while also ensuring visibility of the surrounding geometry.

The space deformation ϕ is derived from the user specified path-points $\mathscr{X} = \{x_1, \ldots, x_Q\}$ such that $\phi(\mathscr{X}) = \{\phi(x_i), \ldots, \phi(x_Q)\}$ lie along the negative z-axis, i.e. the viewing direction of the final image.

These conditions alone however do not yet suffice to specify all the important attributes of ϕ for creation of a single image path depiction. Therefore every path-point is also equipped with four auxiliary points which are used for specifying up- and right- direction as well as a scaling factor.

Finally a thin-plate spline [8] is used to define the globally smooth space deformation ϕ that adheres to the user specified constraints. In our implementation we derive an initial set of appropriate constraints automatically and give the user the opportunity to interactively adjust these in two ways:
First, the user has the possibility to change the spacing between transformed path points, to adjust the amount of image space a certain segment of the path occupies in the warped image. This can be used to reduce the space needed for long uninterrupted sections of the path in favor of complicated junctions. Furthermore, in many indoor situations, it might also be reasonable to adjust the scaling defined by the auxiliary points, e.g. to widen a narrow doorway along the path, providing a better view on the space behind.

6 IMAGE COMPOSITION

The final image I is composed from the deformed, bounded proxies $\phi(\mathscr{B}) = \{\phi(a_j^i) \mid a_j^i \in \mathscr{B}\}$. First, for each such proxy an image $I_j'^i = P(\phi(a_j^i))$ is rendered under the same projection P which is also used for the composed image I. It is these images which will be stitched in order to form the final path depiction.

The stitching is formulated as MRF energy minimization as in (1) [2][1]. In order to achieve visually appealing results we identify the following properties that the final image should possess [4]:

Visibility in the composed image should be consistent within in the image itself, i.e. a chair in front of a wall should not be partially occluded by the wall. Moreover visibility in the composed image should mimic visibility in the real world, i.e. if there is a chair in front of a wall in the real scene it should also appear in the composed image.

Angular deviation between the viewing ray of a pixel in the final image and the line of sight under which the photograph used for filling that pixel was taken should be minimal. A small angle reduces parallax artifacts and therefore increases visual quality of the stitched image.

Continuity between neighboring pixels in the composed image. That is neighboring pixels in the composed image should have similar colors as neighboring pixels in the original input images. This ensures recognizability of image features in the composed result.

Resolution sensitivity means that the image should be composed from those input photographs that possess the above properties and at the same time give the highest resolution when projected into the final composite.

All these aspects are captured in our MRF energy formulation that follows the pattern of (1). For the stitching, the labeling $\mathscr{L} : I \to P(\phi(\mathscr{B}))$ now assigns each pixel one of the projected proxies as color source. The terms D_x and S_{xy} have to be redefined to reflect the above desired properties. D_x will be responsible for visibility, angular deviation and resolution sensitivity while S_{xy} is used to encourage continuity. Thus we set D_x as

$$D_x(I_j'^i) = \alpha V_x(I_j'^i) + \beta A_x(a_j^i) + \gamma R_x(a_j^i) \qquad (4)$$

where $V_x(I_j'^i)$ is the depth value of $I_j'^i$ at pixel x and therefore rewards correct visibility as proxies closer to the eye receive lower cost. A_x is the term for controlling angular deviation of viewing rays from camera line of sight and R_x penalizes low resolution.

Compared to previous approaches such as [4] our setting differs in that the proxy geometry is subject to a space deformation ϕ. As a result, camera line of sights are effectively bent after application of the deformation. Therefore, in order to compare the direction under which a proxy is viewed after the deformation to the direction of the original camera line of sight we also need to apply ϕ to the direction of the line of sight (see Figure 3. Let $\vec{v}_x$, $\|\vec{v}_x\| = 1$ denote the direction of the viewing ray of pixel x which intersects $\phi(a_j^i)$ in p_x and let $s(t) = c_i + t\vec{r}$ be the line of sight under which $p_x' = \phi^{-1}(p_x)$ is observed by camera i. Then $\frac{d\phi}{dt}(s(\cdot)) = d\phi(\cdot)\vec{r}$ is the direction with which the bent line of sight intersects $\phi(a_j^i)$. Thus we let

$$A_x(a_j^i) = 1 - \alpha \max(0, \vec{v}_x' d\phi(p_x')\vec{r}) \qquad (5)$$

with normalizing factor $\alpha = 1/\|d\phi(p_x')\vec{r}\|$.

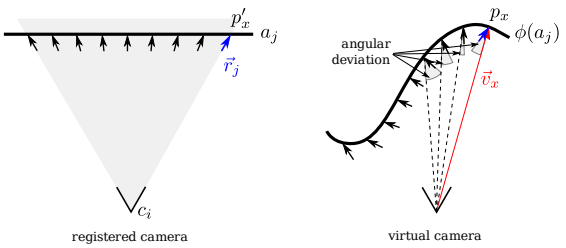


Figure 3: The angular deviation criterion under a space deformation ϕ. On the left hand side, the unbounded proxy plane a_j and the camera of image I_i are shown, as well as one example viewing ray $s(t) = c_i + t\vec{r}$ highlighted in blue. On the right hand side, the deformed proxy geometry $\phi(a_j)$ is depicted. One viewing ray of the virtual camera with view direction $\vec{v}_x$, used to project the deformed proxy, is highlighted in red. At the intersection point p_x between $\phi(a_j)$ and the viewing ray, the angular deviation is indicated as the angle between the viewing ray direction $\vec{v}_x$ and the deformed viewing direction $d\phi(p_x')\vec{r}$.

The last term R_x prefers proxies with a texture sampled in a resolution similar to that of the target image. Thus the sampling rate has to be determined in the input photograph as well as in the target image. Let x' be the pixel of I_i in which camera i observed $p_x' = \phi^{-1}(p_x)$ and let p_y' be the point of a_j^i which was observed in the neighboring pixel $y' \in I_i$. Then we define

$$R_x(a_j^i) = (\|p_x' - p_y'\| - \|p_x - p_y\|)^2 \qquad (6)$$

Finally the term S_{xy} is given as

$$S_{xy}(I_j'^i, I_l'^k) = \sum_{u \in \{\mathscr{N}_x \cap \mathscr{N}_y\}} \|I_j'^i(u) - I_l'^k(u)\|^2 \qquad (7)$$

where $\mathscr{N}_x$ is the neighborhood of pixel x, including x itself. S_{xy} computes the sum of squared distances between the corresponding neighbor pixel values in $I_j'^i$ and $I_l'^k$. This penalizes compositions with visible seams between different image patches.

User interaction

Using the D_x term it is possible to easily integrate user specified constraints into the optimization. In our system the user is able to encourage or discourage the use

of a given image in certain parts of the target image. This is realized by interactively marking the respective regions of the target image similarly to the partitioning interface.

6.1 Optimization

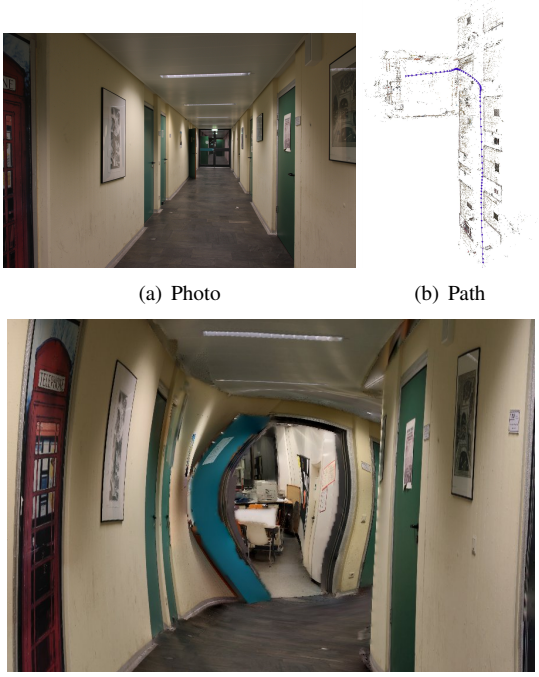
Two Step Stitching

To find a labeling $\mathscr{L}^*$ for the composition we need to minimize (1) where D_x and S_{xy} are defined as stated above. The algorithms we employed to this end in Sec. 4 are efficient if the number of labels is relatively small, i.e. a few hundred. However, in case of the image composition the number of labels can be very large, i.e. $|\mathscr{B}| = O(|\mathscr{I}||\mathscr{A}|)$. Therefore we propose the following two-step procedure to find the final image composite:
(1) We compose a single image I'_j for each proxy plane $a_j \in \mathscr{A}$ from all the I'^i_j using the energy described above.
(2) Then we compose these I'_j into the final result I, using the same energy formulation. The data-cost-terms for single bounded-planar-proxies a^i_j, however, are now replaced by a look-up using the composition computed in step 1.
This dramatically reduces the number of labels in each step. In the first stage there are only $O(|\mathscr{I}|)$ labels while there are exactly $|\mathscr{A}|$ labels in the second phase. Of course the final result is usually not the same as obtained when directly minimizing the full problem, but we found the results to be visually equivalent in our experiments. The reason for this is that the set of images I'^i_j usually is highly redundant due to the overlap between the input photographs I_i and therefore a large subset of the possible labelings in the full optimization actually produce very similar results. In our approach on the other hand this redundancy is eliminated in the first step so that the second step only has to resolve ambiguities between the different proxies.

Image Shifting

Since the scene geometry is only approximated by a coarse proxy geometry, parallax artifacts are to be expected. In order to deal with these artifacts we, first of all, identified the need to formulate an angular deviation constraint to minimize these artifacts, but we also extend the stitching algorithm to shift the input images $I \in \mathscr{I}$ by some pixels to further eliminate parallax effects. This extension can easily be implement by simply adding shifted versions $P(\phi(a^i_j))'_{x,y} = P(\phi(a^i_j))_{x+s_x,y+s_y}$ of the projected proxy geometry to the input of the stitching algorithm. The allowed shift $(s_x, s_y) \in \{(x,y) \in \mathbb{N}^2 : |x| \leq \sigma \wedge |y| \leq \sigma\}$ is specified by the parameter σ. Higher values of σ allow better compensation for bigger parallax artifacts. This, however, entails a significant increase in optimization times, since the number of labels in the optimization grows quadratically with σ.

Because only small values of σ can be used in practice, the angular deviation costs, which in theory vary for different shifted versions of a projection, do not need to be recomputed, as the changes in the view direction are negligibly small for differences of only a few pixels.



(a) Photo (b) Path

(c) Image

Figure 4: A path through a corridor into a kitchen. (a) A photograph taken from the starting location. (b) The user specifies the path in the sparse 3D point-cloud (path points depicted in blue). (c) The final image composition after the space deformation.

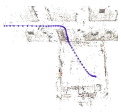
Gradient Domain Fusion

In order to further reduce visible discontinuities between different patches, which might occur for example due to differences in material appearance under varying viewing angles, we deploy a Poisson image editing technique on the final image, which in the context of image stitching is also referred to as *gradient domain fusion* by Agarwala et al.. For details on this optimization, please see [2].

7 RESULTS

We have applied our method to two different scenarios. In Fig. 1 we show a path through a supermarket. The input data consists of 378 images taken with a handheld camera in about 25 minutes. Care has been taken to ensure sufficient overlap between images as required for SfM. Apart from that no further acquisition planning was necessary. The reconstructed 3D points are

(a) Photo



(b) Path

(c) Automatic result

(d) User corrected image

Figure 5: A path from a kitchen into a corridor. (a) A photograph taken from the starting location. (b) The user specified path. (c) The result generated by our approach fully automatically. (d) The final result after a few user corrections.

(a) Photo

(b) Photo

(c) Image

Figure 6: An overview visualization of a museum's foyer. (a) The foyer is too large to fit into a single photograph. (b) Our rendering on the other hand shows all the exhibits and adjoining corridors in the room.

depicted in Fig. 1 middle together with the visualized path shown in blue. A total of 69 planar proxies were automatically extracted from these points. In the partitioning stage it was occasionally necessary to manually mark floor and ceiling due to lack of feature points in these low texture regions (roughly 20% of the input images). The final stitching result was obtained without further user interaction. As shown in the right of Fig. 1 the resulting image gives a good overview of the path clearly depicting features along the way which serve as landmarks during navigation. Please note that warped path-images are not intended to convey spatial survey knowledge but rather unfold their true power if the sequence of visible features matches the observer's immediate surroundings. A few inconsistencies due to geometric approximation errors become apparent under close examination. However, these minor artifacts are perceptually insignificant and do not compromise the visualization's purpose.

The corridor dataset depicted in Figure 4 and 5 was taken in an office environment which exhibits far less natural salient image features. Since features are fundamental for SfM, a slightly increased number of images was necessary. In total 400 images were taken with a hand held camera in approximately 30 minutes. From the sparse point cloud a total of 29 planar proxies were extracted. For the path emanating from the kitchen shown in Fig. 5 we applied a manual correction in the composition step to resolve occlusion problems: In this way the door was removed to give a better view into the corridor. Moreover a false occlusion due to the lack of seed points on the homogenous table surface in the front was corrected (see Figure 4 (c) and (d)).

Finally, the museum dataset (see Fig. 6) consists of 91 pictures and 13 proxy planes. This image's primary purpose is to give an overview of the museums foyer with special emphasis on the adjacent corridors. Again it was necessary to occasionally give some manual hints in the partitioning phase on the featureless white walls, but the respective planes were automatically extracted. Note that the smudgy brown area on the lower left in the image is due to unobserved areas in the input dataset.

In our approach most of the processing time is spent in the preprocessing step: The SfM step took roughly 5 hours for 400 images. Apart from preprocessing, the runtime of our method is dominated by the first step in the optimization described in Section 6.1: For all data sets it took about 15 minutes to compose textures on all proxies. Most time was spent on the floor and ceiling planes which are visible in many images (4.4 minutes on average). The final composition step took only 1 in the corridor and museum scenarios and 1.8 minutes for the supermarket dataset at an output resolution of 900×600 pixels. The times for input image partitioning (4 seconds on average) and warping are negligible.

8 CONCLUSION

This paper presents a novel approach to creating high-quality single image visualizations of short, possibly curved, routes from a series of input photographs. To accomplish this goal we introduce a novel image-based rendering technique able to deal with space deformations. It can be seen as a hybrid between unstructured lightfield rendering [4], panorama and image stitching [1, 2] as well as exploration of image collections [17]. We propose the use of simple proxy geometry estimated from a sparse set of 3D points obtained by SfM. Our proxies give only a rough approximation and are explicitly allowed to deviate from the actual scene geometry. A global energy minimization in the image domain stitches the deformed proxies' images and effectively handles parallax and occlusion artifacts and thus allows for very loose fitting proxies. The synthesized images are valuable and inexpensive navigation aids that can be handed out to visitors in public places. Moreover, we believe that beyond functional aspects, warped images also possess a high aesthetic value.

Our method is currently restricted to scenes predominated by piecewise planar geometry. While this is met by most indoor and urban environments, our method is not well suited if the scene geometry is more complex and cannot be reasonably approximated by planes, e.g. in a forest.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude towards the supermarket *Comet Verbrauchermarkt* and the museum *Rheinisches Landesmuseum* in Bonn, Germany.

REFERENCES

[1] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. In *SIGGRAPH*, pages 853–861, 2006.

[2] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *SIGGRAPH*, pages 294–302, 2004.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[4] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, pages 425–432, 2001.

[5] S. E. Chen and L. Williams. View interpolation for image synthesis. In *SIGGRAPH*, pages 279–288, 1993.

[6] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *SIGGRAPH*, pages 11–20, 1996.

[7] P. Degener, R. Schnabel, C. Schwartz, and R. Klein. Effective visualization of short routes. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1452–1458, 2008.

[8] J. Duchon. Spline minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables*, volume 571 of *Lecture Notes in Mathematics*, pages 85–100, 1977.

[9] M. Eisemann, B. D. Decker, M. Magnor, P. Bekaert, E. de Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. Floating Textures. *Proc. Eurographics*, 27(2):409–418, 4 2008.

[10] S. Gortler and R. Grzeszczuk. The lumigraph. *SIGGRAPH*, pages 43–54, 1996.

[11] M. Levoy and P. Hanrahan. Light Field Rendering. In *SIGGRAPH*, pages 31–42, 1996.

[12] L. McMillan and G. Bishop. Plenoptic modeling: an image-based rendering system. In *SIGGRAPH*, pages 39–46, 1995.

[13] M. Pollefeys, L. J. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[14] C. C. Presson. The development of map-reading skills. *Child Development*, 53(1):196–199, 1982.

[15] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.

[16] N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski. Finding paths through the world's photos. In *SIGGRAPH*, pages 1–11, 2008.

[17] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*, pages 835–846, 2006.

[18] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *Int. J. Comput. Vision*, 80(2):189–210, 2008.

[19] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1068–1080, 2008.

[20] M. Vergauwen and L. Gool. Web-based 3D reconstruction service. *Machine Vision and Applications*, 17(6):411–426, 2006.