EUROGRAPHICS 2012 / P. Cignoni, T. Ertl
(Guest Editors)

Volume 31 (2012), Number 2

Interactive Multi-perspective Imagery from Photos and Videos

Henrik Lieng[1,2] James Tompkin[1] Jan Kautz[1]

[1]University College London [2]University of Cambridge

Figure 1: Our simple multi-perspective interface makes it easy to improve the appearance and information conveyed in a photograph. *Left*: Original photograph. *Middle*: New perspectives change dark alleys to show destinations. *Right*: Recursive multi-perspective adds trees to the end of the left alley. Please watch our supplementary video to see the interactive multi-perspective exploration of this scene.

Abstract

Photographs usually show a scene from a single perspective. However, as commonly seen in art, scenes and objects can be visualized from multiple perspectives. Making such images manually is time consuming and tedious. We propose a novel system for designing multi-perspective images and videos. First, the images in the input sequence are aligned using structure from motion. This enables us to track feature points across the sequence. Second, the user chooses portal polygons in a target image into which different perspectives are to be embedded. The corresponding image regions from the other images are then copied into these portals. Due to the tracking feature and automatic warping, this approach is considerably faster than current tools. We explore a wide range of artistic applications using our system with image and video data, such as looking around corners and up and down stair cases, recursive multi-perspective imaging, cubism and panoramas.

Categories and Subject Descriptors (according to ACM CCS): I.3.m [Computer Graphics]: Miscellaneous—

1. Introduction

Photographs are usually captured from a single perspective, as we expect from a standard lens. However, sometimes we want to convey information in a single image which cannot be captured from a single perspective. This is solved by using multi-perspective images which merge multiple images taken from different perspectives into one single image.

Multi-perspective images have been used for centuries in art. An early example is traditional Indian miniature paintings where the perspective is distorted in order to see what is behind large objects such as walls. In the past century, multi-perspectivism has seen a sharp rise in the artistic community. The Cubism movement, pioneered by Pablo Picasso and Georges Braque in the early 20th century, represents objects as if all their faces are visible from one perspective. Synthetic cubism introduces the concept of merging together different surfaces and textures. Here, different regions are incorporated into one object. The term *collage* was coined by Picasso and Braque and derives from the French word ‘colle’ (glue). This has inspired countless artists to glue together pieces from newspaper clippings, ribbons, photographs, wood and other materials. The work of David Hockney is one example of combining photographs taken from slightly different perspectives into a collage (photo-collage). This was a replacement for wide-angle lenses which, according to Hockney, produce unwanted distortions.

Multi-perspective images can be composed into visualizations of impossible realities and objects. M. C. Escher produced many such paintings during the mid-20th century. These images look consistent locally but, when we 'zoom' out and look at the picture as a whole, there are inconsistencies globally. Hence, the paintings look seamless compared to the shattered style of cubism. Examples of related contemporary work include paintings by Rob Gonsalves and Clive Head. The former combines Escher's impossible constructions, Salvador Dali's surrealism and magic realism. On the other hand, the work of Head strives for realism even though the paintings represent multiple perspectives. Here, information from different views from a real world position is collected and combined into a realistic painting, capturing the experience of moving around the scene.

Research on multi-perspective images tends to focus on certain aspects. Usually, it is inspired by artistic work. For example, previously explored styles include cubism [CH03], Hockney style collages [NZN07], seamless panoramas [BL07] and impossible perspectives [PRAV09]. The goal of this work is to make a general system for designing multi-perspective images and videos. This approach is similar to synthetic cubism and collages where regions from other images are copied into a target image. This work is also associated with the work of Escher, Gonsalves and Head as we seek seamlessness (i.e., locally consistent images). However, we do not restrict ourselves to certain types of images but rather assume that the camera can be transformed by any kind of perspective transformation. This includes rotation such as in panoramas (Figure 6), and rotation with translation, by both panning (e.g., look around the corner, Figure 1) and tilting (e.g., look up or down a staircase, Figures 7 and 8) . Additionally, we can change the perspective projection by incorporating new images that have been taken further into the scene, such in Figure 9.

The proposed system copies image regions from multiple images into one single image (the target image). The image region in the target image that is about to be replaced is defined as a portal. This portal is user specified by selecting the vertices of the overlapping polygon which can be of any shape. These vertices are tracked through the image sequence using structure from motion (SfM). Then, the polygons defined by the tracked points are warped such that they overlap with the portal in the target image. The resulting image is produced by composition between the target image and the warped image region. To aid seamlessness, we blend the border between the two regions using feathering. Multiple portals and video are supported as well. Videos can also be computed from sequences of photographs, and in these cases we ensure consistent brightness and allow optional warping to smooth motions and temporal effects (Section 3). The workflow of our method is shown in Figure 2. Figure 3 shows an example of a target image, the defined portal and the resulting multi-perspective image. Additionally, we show other applications of this framework such as interactive multi-perspective images and object removal. Some of these kinds of images can be made manually in graphics software such as Adobe Photoshop; however, often this is a tedious process. We demonstrate that our system can generate visually comparable results in seconds after pre-processing, and provides an 8-10× time saving when dealing with even moderately sized image sets. We interviewed artists in-depth, and they responded well to the system and indicated that they would use it for designing such images (Section 5).

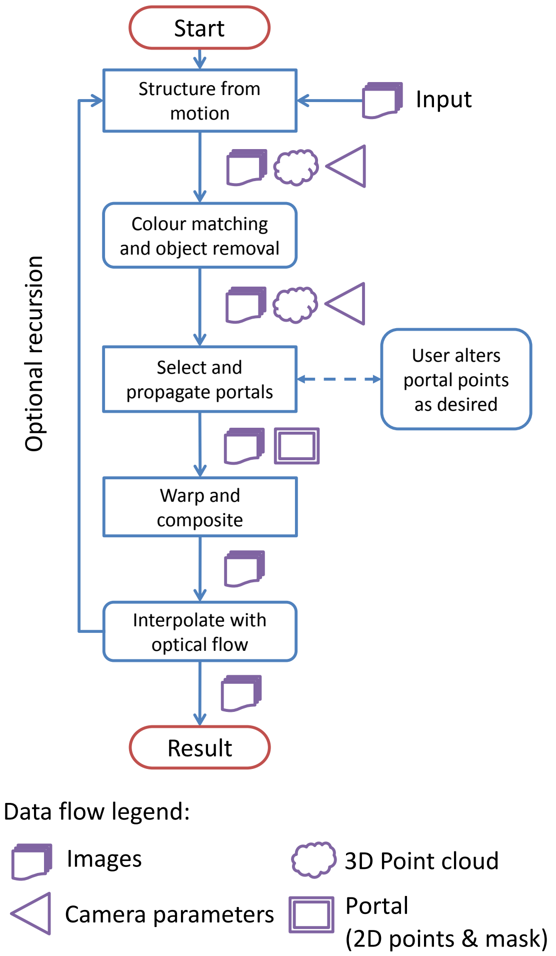


Figure 2: The workflow of our method. Optional processes have rounded corners.

2. Background

A common problem is to make one seamless image out of multiple images taken from slightly different perspectives (e.g., panoramas). Wood *et al.* produced one early example of this for 2D cel animation [WFH*97]. This research tried to simplify the process of making such animations by defining keyframes and letting the computer interpolate between keyframes. This work only considered synthetic scenes; therefore, the images are always correctly aligned. Hence, the mapping from 3D space to the space defined by the panorama image (usually a cylindrical canvas) is relative straightforward.

To automatically construct panoramas from real world images, camera parameters such as position and rotation must be estimated. Brown used Lowe's Scale Invariant Feature Transform (SIFT features) [Low04] in the matching process to estimate the necessary homography transform [BL07]. Additionally, the matching error is minimized using the Levenberg-Marquadt optimization strategy. To seamlessly blend the panorama, multi-band blending at different frequencies was used [BA83]. Examples of other types of panoramas are slit-scan panoramas [RGL04] and multi-viewpoint panoramas [AAC*06]. Even though these techniques can be used to create similar images to ours presented in this paper, they are only designed for certain types of multi-perspective images and therefore are not suited for general multi-perspective image editing. For instance, none of these techniques support user-defined region control and so this significantly limits the kinds of images that can be produced.

There are different approaches to image composition and blending, such as alpha blending, Laplacian pyramid blending and graph cuts [KSE*03]. Which technique is appropriate depends on the application and the desired result. Nomura *et al.* argue that visible seams in a collage provide intuitive representations of the structure of the scene [NZN07]. They display a collage with opaque layers, transparent layers or with blended image boundaries. This work was inspired by Hockney-style collages. In the cubism from photographs project the regions are opaque [CH03]. A post processing step creates a painterly effect on the images. When seamlessness is preferred blending (e.g., in panoramas [BL07]) or cutting (e.g., in push-broom images [AAC*06]) is usually used.

Multi-perspective images can also be used to convey more information about a single object or scene that is possible with a single perspective image. Rademacher and Bishop used a strip camera to create multiple-centre-of-projection images (MCOP) [RB98]. A strip camera consists of a moving strip of film behind a vertical slit. That is, we capture multiple images along a continuous curve or surface. Each image corresponds to at least one column in the resulting image. Popescu *et al.* propose a non-pinhole camera model, the graph camera, with rays that circumvent occluders and bend around corners [PRAV09]. Applications include 3D scene exploration, summarization and virtualization. We believe it would be possible to produce comparable results to ours using this framework as their portal-based constructor is similar. However, their system is impractical for real-world photographs and videos and, as the input data is different, does not attempt to solve the same problem. Previous work in geovisualization looked at interactive systems for multi-perspective images [MDWK08, PTD11]. However, these are limited to a small subset of multi-perspective views. For an overview of previous research on multi-perspective imaging we refer the reader to Yu et al. [YMS08].

3. Multi-perspective images

The system developed is a general editing and interaction system for multi-perspective images and videos. The results are created by copying image regions from other images into one target image or video. If the images are of the same scene, SfM is used to track the corner points defining the portal in the target image. This minimizes the amount of user interaction without limiting the capabilities of the system. Unlike related research, which usually restricts the movement of the camera to only certain transformations, we support any kind of perspective camera transformation.

3.1. Interface

We implemented a simple MATLAB user interface to assess the framework. The artist can load images and videos along with camera parameters and 3D information (i.e., a point cloud). Corner points are chosen by the user and this defines the necessary image warp. That is, the window inside the polygon defined by the warp points in all images is warped into the same window in the target image. As not all images in a working set will be similarly exposed or densely and regularly sampled, we include systems for motion synthesis and colour matching to provide smooth multi-perspective animations. We also provide masking tools to handle occlusions in the target image. For instance, often railings, street furniture and tree branches will occlude an otherwise good portal, and so we include functionality to mask and preserve their appearance in the target image while the multi-perspective illusion continues behind.

Compared to the user interface by Popescu *et al.* [PRAV09], we provide an intuitive direct manipulation interface for real-world photographs. Their graph camera is constructed by interactively drawing frustums. For photographs, this is difficult without some kind of 3D representation of the scene. The sparse point cloud that we use would not be sufficient for this task as it is more difficult for a user to interpret and manipulate than picking 2D points as in our system. Additionally, frustums are not well suited for all kinds of portal transformations, and we demonstrate results on non-rectangularly-shaped portals. Furthermore, a frustum has a well-defined shape, and our system allows polygon-shaped multi-perspective regions (see Section 4). Our system is more intuitive since the users directly clicks what they wish to include. Of course, it is left to the photographer to capture the required images for their desired transformation (e.g., panning, tilting shots away from the target image).

3.2. Making multi-perspective images

To make multi-perspective images with as little user interaction as possible we need to calibrate the images. That is, to find the 3D position of feature points and the camera poses. Here, SfM systems were used (Bundler for photographs [SSS06] and Voodoo camera tracker for videos [Tho06]). After the images are loaded and the calibration data is parsed, the cameras are optionally sorted as, when the user is browsing through the images, we wish to present an intuitive ordering. For example, if the user is making a 'look around the corner' image sequence, we wish to view the images with respect to the distance to that corner such that the user is moving closer to the corner as he progresses through the sequence. Since the feature points around the corner are visible in every image we receive a high density of points in this area from SfM. We then make the assumption that the centroid of the point cloud is always located in the area of the corner. This assumption held for all our tested scenes. We then sort the cameras with respect to the centroid distance such that the closest camera is the last image in the sequence.

Figure 3: *Left*: User defined portal mask. The user specifies the mask by selecting a polygon within the original image. The vertices are then automatically found in all other images in the working set. *Middle*: Original image. *Right*: Resulting multi-perspective image.

The user picks points with the mouse in the target image. Having picked the warp points, as shown in the accompanying video, we track them over the rest of the image sequence. First, we find the closest projected 3D corner points in the point cloud to the 2D points chosen by the user. Here, all points in the point cloud are projected onto the image and all points within a given image-space radius are gathered (e.g., 5 pixels). The closest point in terms of depth (i.e., smallest positive z-coordinate) is then chosen. The reason for this gathering approach is that we want to choose the closest point that is actually visible in the image. This way, we do not pick a 3D point on a wall behind a corner (for example). These 3D points are then projected to every other image in the sequence and define the portal.

As the distance spatially between the target and source image increases, the tracking error is likely to increase as well. Therefore, we let the user interactively change tracked corner points if necessary. Since neighbouring cameras are better aligned, we run the projection algorithm over again using the new corner points that the user has selected over neighbouring cameras. Suppose we have 20 sorted cameras where camera 1 is farthest away from the corner and camera 20 is closest. If the user changes the warp points in camera 10, we reproject the 3D points from these new points into cameras 6-20. If the user further changes the warp points in camera 15, cameras 13-20 are updated and so on. With this approach we are able to quickly and precisely track the corner points over the whole sequence without too much interaction.

The image region in the source image is copied into the portal in the target image by warping. The polynomial transformation is defined as:

$$\begin{aligned} x' &= a_0 + a_1 x + a_2 y + a_3 xy \\ y' &= b_0 + b_1 x + b_2 y + b_3 xy \end{aligned} \tag{1}$$

The warp is then computed using the n vertices in each portal:

$$\begin{bmatrix} x'_1 & y'_1 \\ \vdots & \vdots \\ x'_n & y'_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n y_n \end{bmatrix} \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix} \tag{2}$$

Finally, we use α-blending when compositing the result. α was computed by blurring a binary version of the portal with a Gaussian kernel. For the results presented in this paper, a 20x20px Gaussian kernel with a standard deviation of 20 was used.

When working with video, the user either selects the target as a video (i.e., video in a video) or a single image (i.e., video in an image). When the target is a single image, we use the same system as with photographs. The only difference is that we have more images to work with in a video sequence. For video targets we assume that the two videos are synchronized and have the same length. The target image then changes as we progress through the video. That is, at a given time step t we warp and composite from image t in the source video into image t in the target video.

3.3. Colour matching

If the difference in colour distributions between the warped image region and the target image is large, it is possible to see discontinuities on otherwise identical semantic image features (such as brightness variation along a wall). Simply blending the intersection between the target image and warped region will not work: commonly, photographs are captured with automatic settings, and we wish to provide a system that requires no special capture parameters. Factors affecting why overall intensity in photographs of the same scene varies include changes in aperture/exposure time, vignetting and radial distortion. Therefore, we optionally align the colour distributions using the technique of Reinhard *et al.* [RAGS01] to compensate for colour and brightness differences between images.

Figure 4 shows a result where the dark wall of the warped image region is correctly colour matched to the corresponding wall in the target image. With this approach, we must be careful not to include other parts of the image beyond the regions to be matched as this will influence the results proportionally. Therefore, we segment out the relevant portal edge parts of the two images and align only these.

Figure 4: *Left*: Original multi-perspective image. *Right*: Result after aligning the colour distributions on the wall with our system. Zoom regions outlined in red show mismatch on bricks without colour transfer.

3.4. Recursive multi-perspective images

Additionally, we support multiple recursive portals in our system. A new portal can be placed anywhere in the image, also on top of other portals, such that they recurse. When computing the result, the image regions from the inner portals are inserted subsequently. Recursive multi-perspective images can also be designed by computing the first recursion and then select the new portal in the resulting image. This system is very intuitive - there is no difference to the user when they create a recursive portal than to when they create a normal portal.

3.5. Warp interpolation

When attempting to look around a corner with a series of photographs, it is easy to produce jerky motion as we have only a sparse set of images. Therefore, motion synthesis was added such that we smoothly interpolate between two warps. First, we estimate the motion field between these two warps using SIFT Flow [LYT11]. Second, we synthesise n number of images using the motion field. For more detail on the two interpolation schemes supported by our method (linear and bidirectional), please see the additional material.

3.6. Object removal

Texture synthesis is the only reasonable way to remove medium and large occluders with data from only a single image. This can either be done fully manually (e.g., with a clone tool in image-editing software) or fully automatic with methods such as PatchMatch (e.g., Photoshop's 'content-aware fill' feature) [BSFG09]. However, these methods often fail for large occluders, and here a single image is often not enough to achieve a successful object removal. With multiple images, our framework can be used to remove objects in a semi-automatic manner. The user chooses corresponding points in both the target image and an image without the occlusion. First, the user selects the occluder (such as in Figure 5, a) and the image where the occluded area is visible. Next, the user picks out correspondence points surrounding the occluder (such as in Figure 5, b and c). The warp is then computed and composited using our framework. The accompanying video shows how this is performed in our system.

The main advantage of this approach is that the user is in full control when deciding what is behind the occluder. Letting the computer automatically compute an inpainting with acceptable results for structured areas is still an open problem for large occluders. For instance, we would not achieve acceptable results when removing the tree from Figure 5 using current automatic tools. We believe our system is also intuitive for artists since it is similar in principle to manual approaches of copying from other images, and this is found to be the case in Section 5.

3.7. Interactive multi-perspective images

We provide the option of visualizing results interactively with our system. For example, when wishing to look around a corner, the user clicks on the portal and drags left or right in order to bend space and see what is around the corner. Both multiple, separate portals and recursive portals are supported in this interactive multi-perspective mode. From the accompanying video, we see that the system is especially intuitive for scene exploration. We argue that this approach provided by our system is more prudent in many use cases than a single image which tries to convey all perspectives, such as in existing techniques, as visual content is squeezed into one image and the 3D relationship between objects is lost. We start with a coherent image and interactively transform it. This interaction connects the user's actions to the view, which may lead to better scene comprehension and understanding.

4. Results

Our system, along with source code and all our source data, is available at `http://www.cs.ucl.ac.uk/research/vr/Projects/InteractiveMultiPerspective/`.

The performance of our system depends on its three major components: SfM estimation, portal propagation and warping. We use Bundler to perform SfM estimation for photographs as a pre-processing step. Here, the performance depends heavily on the number of input images, and Snavely et al. provide a more detailed discussion of their performance [SSS06]. Alternatively, Photosynth or VisualSFM [Wu11] could be used, where VisualSFM exploits the GPU for faster performance. When propagating the portal points through the image sequence, we project all points onto each image. Here, the performance depends on the number of images and the number of points in the point cloud. For example, we used 26 photographs (1800x1200) for the scene in Figure 1. Bundler estimated approximately 10,000 points in 7m30s on a 2.4GHz CPU (Photosynth took 1m10s), and the portal propagation took approximately 5 seconds. Finally, we compute the result by warping. This is computed in real time or near real time depending on the resolution of the portal. If we perform interpolation with optical flow, the performance depends on the optical flow system used. SIFT flow is relatively slow but gives accurate results. In our experience, each estimation takes 1 minute. However, real-time GPU optical flow implementations do exists [PVH08] should speed be a priority.

Figure 6 shows a Hockney-style panorama. Here, the target image started out as a white, empty image. Using the concept of a strip camera, a strip from each image was copied onto the white target image. This approach is similar to how artists make collages; gluing pieces of paper, ribbons, wood, etc., onto an canvas.

Figures 1, 7 and 8 show various camera transformations such as panning and translating (looking around a corner), tilting and translating (looking down a stair case) and tilting along a curved camera path, respectively. The points defining the portal in the latter image intentionally do not match between the images. Instead, we pick similar points from completely different viewpoints. This recursive effect was created depth-first.

Figure 9 change information in the image by moving further into the scene, changing the perspective projection. This moves the effective vanishing point deeper into the scene. These changes look continuous, as in Figure 9, since the points defining the portals are the same. We simply redirect the lines where they enter the portal towards a new vanishing point.

Our two supplementary videos (*show reel* and *tour*) show many examples of image and video multi-perspectivity in many different styles. The SIFT flow interpolation produces visually pleasing results when the photographs in a set are taken close together. However, when the photographs are considerably different and the motion flow estimation fails we receive popping artefacts. In Section 5, we discovered to our surprise that our interviewed artists did not object to these artefacts. The video examples follow moving objects through corridors and corners. For video in photograph effects, the target image is the photograph at the point where the object enters the portal. The recursive example was created breadth-first. Note that any shakiness at portal edges is due to camera shake and the failure of our SfM algorithms to correctly track feature points in these difficult cases. We also support recursive video in video. Here, we need as many cameras as the depth of the recursion.

5. Evaluation

> It is self-evident that nothing concerning art is self-evident.
>
> Theodor Adorno, [AATHK04]

Quantitative evaluation of this work is difficult. Beyond the few minutes required to pre-process the working set of images for structure and motion (after which any number of multi-perspective images may be created), there is very little to grasp when using our system and very little time is spent generating images – this is a major benefit of our system. We asked 5 artists to reproduce our examples with the same datasets and software tools with which they were familiar. While their results were qualitatively similar, our system was 5× faster in creating a single image, and 8-10× faster when working with multiple images. This is a substantial speed up, and enables previously infeasible projects.

For instance, we took a dataset of 700 photographs of parts of London, with some shot with a wide-angle fisheye lens. Within two working days, an artist using our system generated over 200 multi-perspective images of many different kinds (panoramas, Hockney-style collages, doorway and staircase portals, etc.). The images were placed into a video such that the viewer travels around London though changing multi-perspective imagery. We believe this kind of mass editing and sequencing would not have been possible in such a short amount of time with existing systems. Our supplementary *tour* video shows these generated images.

Qualitative evaluation is similarly hard as there are no well-defined metrics for art; this is the philosopher's problem. Instead, we try to answer two questions: are artists interested in creating such images, and would an artist benefit from and use our system instead of existing tools when designing multi-perspective imagery?

Figure 6: A panorama in the Hockney style formed with our system from 12 images.

We conducted 5 in-depth interviews with student artists from fine art and architecture disciplines, ranging in length from 20 minutes to $1\frac{1}{2}$ hours. We discussed both the aesthetic values and the design approaches of our solutions to the problems explored in this project: making multi-perspective images, colour matching, motion synthesis and object removal. All had experience with standard image-editing tools and four were self-described experts in both image editing and 3D modelling.

When making these multi-perspective images, our artists would normally copy-paste an image region into the target image and then distort it such that it fits the portal. With existing software this process might take several minutes to achieve a desirable result if the portal is complex, but with our system the portal process takes just a few seconds. One artist argued that she would also change the target image if the portal fit was not desirable. As we track feature points throughout the working set of images, it is trivial to change which image is the target without losing any progress.

Colour matching is easy in existing image editing software, with intuitive interfaces and instant feedback. Moreover, artists are usually highly experienced with these tools since colour adjustments are performed very frequently. For smaller working sets of images, these manual approaches were preferred over our automatic system as they provided absolute artistic control over which parts of the copied region and target image to colour match. For larger sets, such as sets from video, manual correction becomes a burden as hundreds of frames may need to be corrected. In this case, our artists would use our automatic approach if time were of the essence.

The artists we interviewed were enthusiastic about the effect of warping between views as if looking around a corner since this was something new that they had not seen before. Moreover, the slight popping artefacts, introduced by this method when the views are very far apart or when dynamic objects change from frame to frame, were not considered distracting.

Professional artists usually remove objects in an image by manual texture synthesis. For our artists, PatchMatch-derived fill methods [BSFG09] were not popular since they do not provide sufficient control and fail to produce cohesive structure for large regions. In these cases, we feel our approach of exploiting additional images to fill holes is valid. Currently, when the missing information is not available, our artists stated that they would use images from the Web, similar to a manual version of the scene completion of Hays and Efros [HE08]. Our automatic system is intuitive for artists since the approach is similar: define a region in one image, then selecting from a list of candidate completions from the other images in the working set.

Perhaps the only way to convince an artist to use a new system such as ours is to demonstrate as good or better results than current systems and that it is easier and faster to use. The advantages that feature tracking brings to multi-perspective imaging speeds up the process from minutes to seconds, and our interviewed artists were happy with the quality of images our system produces. With fast presentation of results, our system provides a much more immediate response, and we believe this increases creativity. Therefore, we argue that we have created a system that is usable and fast for artists. While we do have a pre-processing step, once completed the user can generate many multi-perspective images from a working set or generate new interactive multi-perspective 'look-around-the-corner' experiences and multi-perspective videos.

6. Conclusion and Future Work

We present a fast and easy-to-use system to create multi-perspective imagery of many different kinds and styles. We exploit SfM for image sets and videos to provide an intuitive way to select and automatically propagate portals for multi-perspective effects. This creates a system which produces comparable results to existing work, but which can speed up image creation by a factor of 10. With this, we can create multi-perspective video, including recursive video in video effects. We also demonstrate a new approach to multi-perspective imagery by creating interactive multi-perspective scenes. In these, a user can look around seemingly obstructive corners and interactively explore a scene while still maintaining a strong sense of space. Finally, we evaluate our system with its intended real-world users by conducting in-depth interviews with artists in which we discuss both aesthetic values and system design approaches.

Although, in theory, we are able to handle arbitrarily-shaped polygon portals, special cases do exist where our system would be impractical. For example, curved portals are difficult to accomplish since we need many recovered 3D points along the curve to define the portal. Figure 9 is at the limit of what we can handle. In these cases it would be better to trace continuous curves instead of our piecewise curves.

Even though the majority of our computation time is spent within existing SfM tools, improvements to performance are possible. For example, we do not use an elaborate data structure to represent the recovered point cloud from SfM, such as a k-d tree or an octree. We simply project all points onto the image, even those that do not fall on the image sensor. Using a space-partitioning data structure may speed this up.

Finally, in this paper, we visualized only some of the different effects that could be made with a given set of images. However, additional effects can be explored. For example, we could create a zoom effect in our exploration tool, such that the viewer appears to travel into the scene and look around corners.

References

[AAC*06] AGARWALA A., AGRAWALA M., COHEN M., SALESIN D., SZELISKI R.: Photographing long scenes with multi-viewpoint panoramas. In *ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 853–861. 3

[AATHK04] ADORNO T., ADORNO G., TIEDEMANN R., HULLOT-KENTOR R.: *Aesthetic theory*. Continuum impacts. Continuum, 2004. 6

[BA83] BURT P. J., ADELSON E. H.: A multiresolution spline with application to image mosaics. *ACM Trans. Graph. 2* (October 1983), 217–236. 3

[BL07] BROWN M., LOWE D. G.: Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision 74* (August 2007), 59–73. 2, 3

[BSFG09] BARNES C., SHECHTMAN E., FINKELSTEIN A., GOLDMAN D. B.: Patchmatch: a randomized correspondence algorithm for structural image editing. In *ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 24:1–24:11. 5, 7

[CH03] COLLOMOSSE J. P., HALL P. M.: Cubist style rendering from photographs. *IEEE Transactions on Visualization and Computer Graphics 9* (October 2003), 443–453. 2, 3

[HE08] HAYS J., EFROS A. A.: Scene completion using millions of photographs. *Commun. ACM 51* (October 2008), 87–94. 7

[KSE*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: image and video synthesis using graph cuts. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, ACM, pp. 277–286. 3

[Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60* (November 2004), 91–110. 3

[LYT11] LIU C., YUEN J., TORRALBA A.: Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell. 33* (May 2011), 978–994. 5

[MDWK08] MÖSER S., DEGENER P., WAHL R., KLEIN R.: Context aware terrain visualization for wayfinding and navigation. *Comput. Graph. Forum 27*, 7 (2008), 1853–1860. 3

[NZN07] NOMURA Y., ZHANG L., NAYAR S.: Scene collages and flexible camera arrays. In *Proceedings of Eurographics Symposium on Rendering* (Jun 2007). 2, 3

[PRAV09] POPESCU V., ROSEN P., ADAMO-VILLANI N.: The graph camera. In *ACM SIGGRAPH Asia 2009 papers* (New York, NY, USA, 2009), SIGGRAPH Asia '09, ACM, pp. 158:1–158:8. 2, 3

[PTD11] PASEWALDT S., TRAPP M., DÖLLNER J.: Multi-scale visualization of 3d geovirtual environments using view-dependent multi-perspective views. *Journal of WSCG 19*, 3 (2011), 111–118. 3

[PVH08] PAUWELS K., VAN HULLE M.: Realtime phase-based optical flow on the gpu. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on* (june 2008), pp. 1 –8. 6

[RAGS01] REINHARD E., ASHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Comput. Graph. Appl. 21* (September 2001), 34–41. 5

[RB98] RADEMACHER P., BISHOP G.: Multiple-center-of-projection images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 199–206. 3

[RGL04] ROMAN A., GARG G., LEVOY M.: Interactive design of multi-perspective images for visualizing urban landscapes. In *Proceedings of the conference on Visualization '04* (Washington, DC, USA, 2004), VIS '04, IEEE Computer Society, pp. 537–544. 3

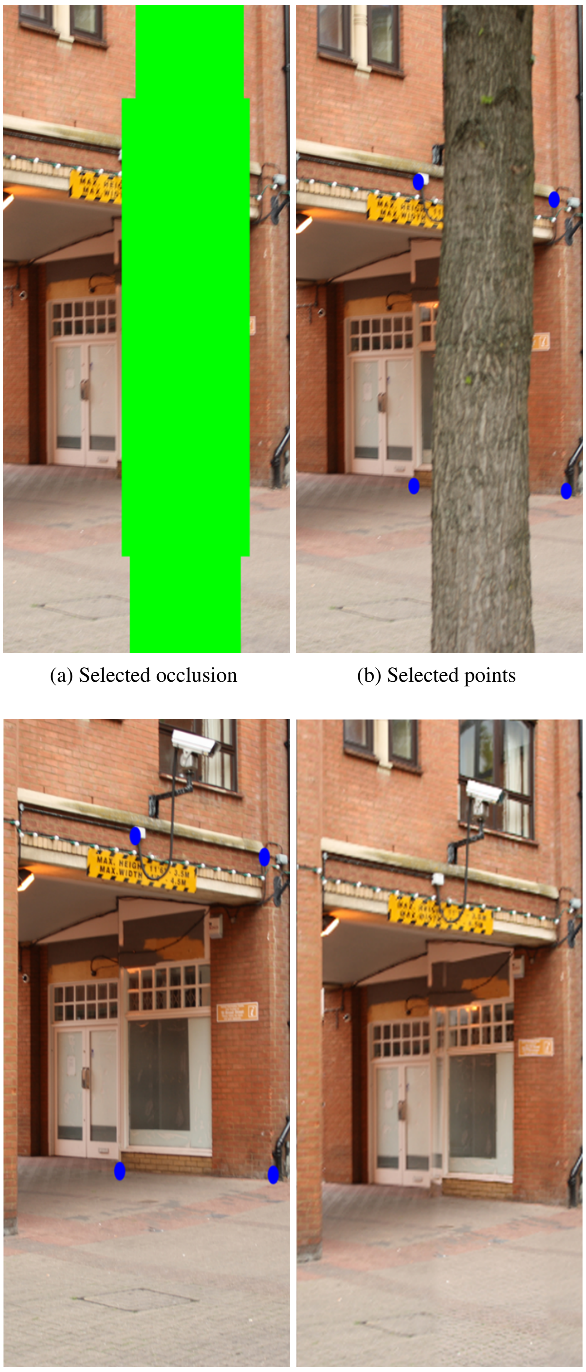
[SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: exploring photo collections in 3d. In *ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 835–846. 4, 6

[Tho06] THORMAHLEN T.: *Zuverlassige Schatzung der Kamerabewegung aus einer Bildfolge*. PhD thesis, University of Hannover, 2006. related software 'Voodoo Camera Tracker' can be downloaded from http://www.digilab.uni-hannover.de. 4

[WFH*97] WOOD D. N., FINKELSTEIN A., HUGHES J. F., THAYER C. E., SALESIN D. H.: Multiperspective panoramas for cel animation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 243–250. 3

[Wu11] WU C.: "VisualSFM: A visual structure from motion system", 2011. 6

[YMS08] YU J., MCMILLAN L., STURM P.: Multiperspective modeling, rendering, and imaging. In *ACM SIGGRAPH ASIA 2008 courses* (New York, NY, USA, 2008), SIGGRAPH Asia '08, ACM, pp. 50:1–50:36. 3



(a) Selected occlusion (b) Selected points

(c) Corresponding points (d) The final result (from the perspective of a)

Figure 5: Inpainting using manual alignment.

Figure 7: Example where the camera is tilted along the camera path. *Inset top left*: Original image.

Figure 8: The camera is tilted along a curved camera path in a Hockney style. *Inset top left*: Original image.

Figure 9: A multi-perspective image where we change perspective projection to improve the visibility in the tunnel. *Inset top left*: Original image.