Image Fusion from uncalibrated Video Sensor Arrays

Thomas Haenselmann, Marcel Busse, Stephan Kopf,
Thomas King, Wolfgang Effelsberg
Department of Applied Computer Science
5 LS PI IV, University of Mannheim, Germany
{haenselmann,busse,kopf,king,effelsberg}@informatik.uni-mannheim.de

September 20, 2006

Abstract

Panoramic images have only been feasible if all contributing image patches originate from a single image source. In contrast to that, we propose a novel warping scheme which allows to merge multi-perspective images originating from a large number of uncalibrated (scattered) cameras. A common trajectory is defined in two source images to be merged. It serves as a cut that allows to concatenate them. Usually, the layout of the cuts does not allow to stitch both images together naively. Thus, two convex combinations of a warped and a canonic coordinate are applied so that both source images fit together at the cutting edge while the inevitable distortion decreases towards the borders of the image to obtain a natural appearance.

1 Introduction

Image sensors have recently become almost as cheap and available as scalar sensor which are used for temperature or light measurements. The *Stanford Multi-Camera Array* project is an early example for the simultaneous usage of more than 100 cheap CCD cameras [8]. Other projects are currently emerging in the field of sensor networks. The ESB sensor node platform by the FU-Berlin is one instance of a small, wireless and video enabled sensor node [4].

Other than scalar values which can be displayed on a virtual map or which can simply be aggregated, it is not obvious how to display a massive amount of (possibly uncalibrated) images, particularly in a way that makes sense for a human observer. Consolidating all images into a single one could be a possible solution. Similar attempts have been made in the field of panoramic images, in which a series of pictures are stitched to one another to produce a continuous view. For a long time, panoramic images have been considered feasible only, if all images have the same focal point respectively if the camera does not alter its location. In this paper we devise a novel method for creating panoramic views from images with varying focal points. The specific problems which arise here are described in the following section along with prior attempts to create panoramic images from movies. Section 3 suggests a basic warping scheme as a solution. In Section 4 we evaluate the basic algorithm and identify a couple of shortcomings which are solved by an extended warping scheme. The outlook in Section 5 sketches future improvements to reduce the amount of human interaction.

2 Related Work

In the context of our paper we will distinguish between truly monoperspective panoramic images and multiperspective imaging.

2.1 Monoperspective Panoramic Images

Panoramic images have been known for more than a century with early applications in war photography, e.g., during the American Civil War in 1860-1865 [2]. Here, photos were captured while rotating a tripod-mounted camera around its optical axis. The panoramic image was simply obtained by setting up the photos next to one another. In the 20th century the so-called *rotating lens cameras* have been engineered.

A lens is mounted pivotable in front of the film. While taking a picture, it rotates between a starting- and finishing-angle. A likewise rotatable aperture with a very narrow slot close to the film prevents the photographic material from being exposed to light more than once while the lens is rotating. The entire procedure reminds of a magnetic tape being recorded.

With the advent of digital cameras, panoramic imaging became popular with a larger audience. Ideally, the images are produces with a tripod-mounted camera which ensures a fixed *center of projection*. Prior to stitching two images together, a perspective dewarping of one of them or preferably even of both at the same time has to be carried out. This process must be applied in a common image space. Mapping the images into such a space is usually done by applying either a tubular or a spherical projection [3].

2.2 Multi-perspective Panoramic Images

Multi-Perspective means that the panoramic image consists of patches which do not have a common projection center, but which are taken from changing viewpoints. This makes stitching particularly difficult or impossible in a naive way since

overlapping parts of neighboring images, which may in principal show the same content, cannot be aligned. This is due to the fact that changing the viewpoint corresponds to a rotation of the objects in the real world. This may hide parts which had been seen before the rotation and reveal new insights afterwards. As a consequence, none of two neighboring images will exhibit simple cuts where one image can be aligned with its neighbor.

One of the earliest instances of multi-perspective imaging is the animated cartoon *Pinocchio* by Walt Disney Productions, which was made in 1940. The film opens with a virtual camera flying over a small village. In contrast to conventional techniques used at that time, the movement of the camera is no mere pan over a scene. In fact the camera seems to perform a rotation at the same time, which alters its viewing direction continuously. The effect was produced by drawing a panoramic image (of ratio 3:1) showing an overview of the village. However, when viewing the image from one end to the other, the perspective seems to change gradually from house to house which creates an impression of a strangely warped scene. The actual shot in the movie was simply made by panning a focused view over the panoramic drawing thus showing only a small clipping at a time. The artists who developed this scene must have had a very sophisticated spatial sense; and it is reported that producing the scene consumed a large portion of the film's budget.

Many decades later, Wood et al. proposed to create similar hand-made animations with the help of a computer in a reverse engineered fashion [9]. The process began with the construction of a 3D-scene in a modeling application. Then, the scene was captured by a moving virtual camera. The resulting digital movie was played back afterwards. Each image is reduced to a column of pixels in the middle. By concatenating each of these columns next to one another for each frame, the animation is "unrolled" into a panorama. We could also say that the X-axis is exchanged for the time-axis. An artist paints the scene on top of the artificial panorama in greater detail. In the final step, an animation is produced as described for the Pinocchio movie. A panning and a rotating camera can well be generated in the 3D-animation, whereas zooms into a scene must be done by zooming into the final drawing made by the artist. Both approaches, the one used by Disney and the one proposed by Wood, create an artificial panoramic image with the aim of extracting a realistic video from it.

The complementary method would be to produce a panorama from existing real world images. Among many others, Kim et al. evaluate the generation of multi-perspective panoramas from videos showing real scenes [5]. Again, the idea is to reduce every image to a single column of pixels, preferably in the middle of the image. Thus, every frame of a captured video contributes a column of the panorama image which is growing from one side to the other, as long as the movie shows a continuous camera operation. The greatest challenge is to move the camera as continuously as possible both in space and time. Even small accelerations result in a warped appearance or complete discontinuities.

Agarwala et al. generalizes the idea of stitching single columns of pixels to stitching entire images as long as the camera moves on a straight path showing a long flat surface [1]. Their aim is to produce a long continuous image of a street where the building fronts form a roughly planar surface. A point P on that surface can be seen likewise from differing viewpoints. The only difference is that neighboring images display point P more on the left or on the right of the photo. However, the authors also discovered that objects in front of the building appear to have different backgrounds when seen from varying perspectives. For this reason they attempt to cut images only at parts showing the building front with no occlusion or transparency.

Rademacher produced similar panoramas [6]. The difference from prior approaches is that he did not aim at producing a result which can easily be interpreted by a human viewer, but which enables the rendering of new perspectives.

Vallance and Calder created ray-tracing images with continuously varying viewpoints. The position of each pixel on the projection surface serves as a parameter of a function which changes the viewpoint slightly. As a consequence, each pixel has an individual center of projection [7]. The benefit of the unnaturally appearing results is that opposing object-surfaces can be seen at the same time.

Today, all approaches based on real images assume a slit camera which produces a sequence of images with only marginally changing viewpoints between successive frames. In the following section we present a new approach to consolidate images with significantly varying viewpoints and viewing directions.

3 Warping for panoramic stitching

In this section we will show that panoramic images are possible, even if the focal point of the camera changes significantly. Of course, the resulting image will imply several changes in perspective and, unlike existing approaches, these changes will by no means be continuous. But, as we will see, this does not necessarily result in an unnatural output.

Figure 1 shows a building from two different perspectives with a certain overlap. In conventional panoramic image generation, semi-transparent images are overlaid as shown on the left side. By compensating the vanishing points of both images, the overlapping regions can be made congruent. Matching two differing perspectives of the building will fail since moving the center of projection does not only alter vanishing points but also changes what is visible and hidden.

Yet, there is a solution to the problem under specific constraints: We have to find a polygonal trajectory (shown as dashed lines in the figure) in both the left and the right images. When being projected into the real world, both should theoretically meet in 3D. Vice versa, if a line was drawn onto objects and surfaces, it should neither be occluded from the left nor from the right perspective.

If the images were cut along the trajectories, the right border of the left image should fit to the left border of the neighboring one regarding the semantics of what can be seen. However,

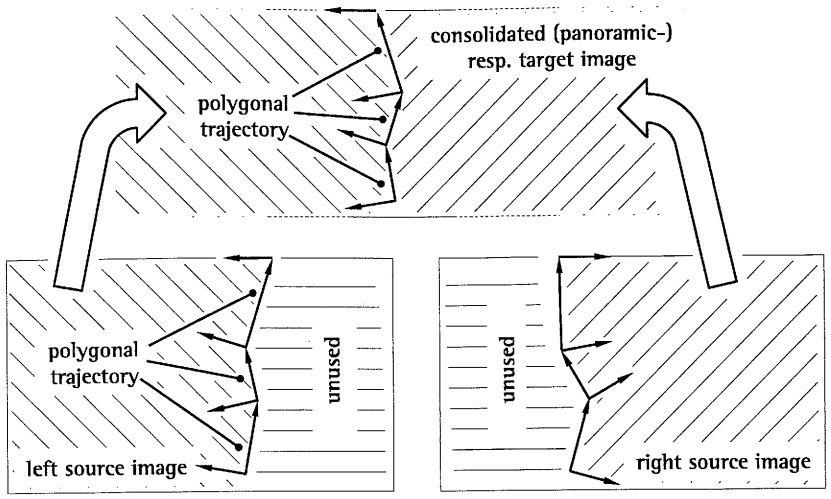

Figure 2: The sketch shows which parts of the left and the right source image in the lower part of the figure contribute to the rendered panoramic image, shown above.

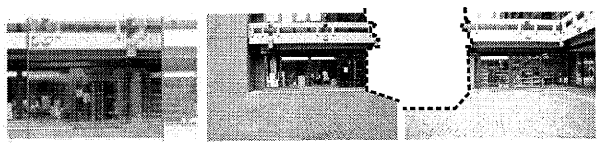

Figure 1: Images taken from different perspectives will usually not match, regardless of how they are moved or distorted (see semi-transparent images on the left). The middle and right images are cut along a common trajectory. Theoretically, both images would fit together semantically but the layout of the cut allows no concatenation.

concatenating both images is not yet possible as the layout of both trajectories is by no means complementary (laying both images next to each other creates holes or overlaps). Warping the images to "meet in the middle" could solve this problem. We will now describe a warping approach that addresses the problem.

Figure 2 exemplifies the process. The upper part of the sketch shows the panoramic image which will also be denoted as the target image. The lower two images are considered source images. Our warping application iterates over every pixel of the panoramic image in the rendering process. For each pixel, the question has to be answered whether the left or the right source image contributes a color value. Once the appropriate contributing source image is chosen, the correct source coordinate has to be calculated.

So far, the trajectory was defined for the left and the right source images. A corresponding trajectory has to be defined for the target image as well. In our implementation this can be done manually by the user. Good results were also obtained by simply averaging the left and the right polygonal trajectory from source images and centering the result in the middle of the target image.

Whenever the panoramic pixel to be rendered is on the left side of the polygonal trajectory, the left source image will contribute a color value; otherwise the right source image contributes a color. The next question to solve for a given pixel in the panoramic image is: Which is the corresponding pixel in the chosen source image? This is shown in detail in Figure 3. The polygonal trajectories are piecewise linear. Each line segment can be considered as the y-axis of a local coordinate system. We will refer to this axis as the *ordinate*. In 2D, a base is entirely defined by calculating the orthogonal x-axis, the *abscissa*. As a result, the left side of the left source image is segmented into distinct areas by those local bases. The same is done with the right side of the right source image and the panoramic target image, accordingly.

The actual mapping works as follows: A coordinate in the target image is expressed by two linear combinations, one consisting of the base $span(\vec{x}_{i-1}^t, \vec{y}_i^t)$ and one by $span(\vec{x}_i^t, \vec{y}_i^t)$ where ^{t,s} in the exponents stand for *target* and *source*, symbolically. Both linear combinations can be considered as two interpretations P_1^t and P_2^t of the same location P^t. Both are based on the same ordinate $\vec{y}_i^t$ but they consist of different abscissas, namely $\vec{x}_{i-1}^t$ at the tip of the y-vector and $\vec{x}_i^t$ at y's base.

$$\begin{aligned} P_1^t &= a_1\vec{x}_i^t + b_1\vec{y}_i^t \\ P_2^t &= a_2\vec{x}_{i-1}^t + b_2\vec{y}_i^t \\ P_1^t &= P_2^t \end{aligned} \tag{1}$$

In the lower part of Figure 3, the two linear combinations are applied to the corresponding spanning vectors $span(\vec{x}_{i-1}^s, \vec{y}_i^s)$ and $span(\vec{x}_i^s, \vec{y}_i^s)$ in the source image. Since almost all involved vectors point into different directions (as compared to the target image) it is not surprising that the two linear combinations do not yield the same coordinate:

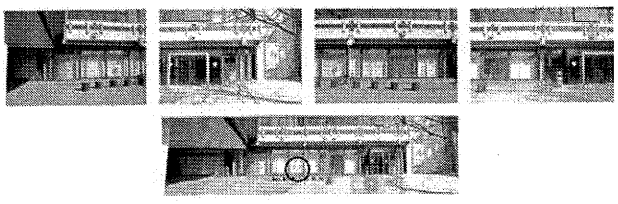

Figure 8: The above view of the building cannot be accomplished using classical panorama techniques since the opposite construction prevents the camera from gaining enough distance. Despite the fact that the viewpoint changes four times, the image still appears credible. However, some artifacts like the duplication of the street lamp's shadow (see black circle) cannot be avoided.

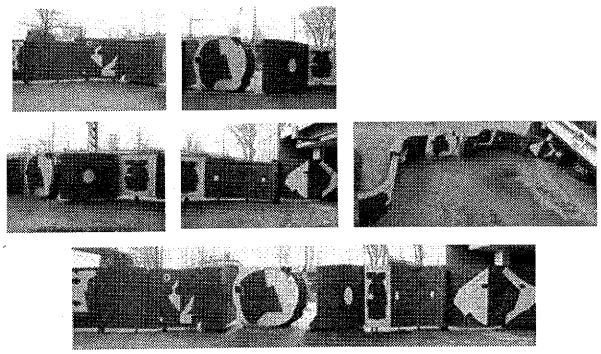

Figure 9: The curved sculpture (overview on middle right image) was unrolled (result on bottom image) from several camera perspectives with changing locations and viewing directions.

Fortunately, the remaining artifacts can be compensated easily. In Figure 7, a regular grid is shown in the lower right. The underlying image will now be calculated in a similar fashion as was done before in the panoramic image. Each pixel is contained in a unique grid cell of the undistorted image to be rendered. Thus, it can be linearly combined by means of the spanning cell boundaries which results in two scalars s and t. The scalars are then applied to the boundaries of the corresponding cell of the distorted grid shown on the lower left of the figure. The color value at the resulting coordinate serves the original pixel in the undistorted image as input. Whenever $s+t<1$ holds true, the pixel is in the upper triangular half of the cell. Otherwise, a new linear combination has to again be obtained by the lower and the left cell boundaries.

The bending of the image originating from the warping process is only one reason for its waviness. Another reason is the distortion caused by the lens itself, which results in a fish eye-like appearance of the left and the right part[1]. The final dewarping compensates for the lens-distortion as well.

[1]The images were taken with a focal length of 18mm

5 Conclusion and Outlook

A new approach for producing panoramic images from photos with varying centers of projection was proposed. A vertical warping scheme distorts neighboring images such that they fit together. Another horizontal mapping makes the distorted parts of the image near the cut converge against the natural image near the left and right border of the panorama thus omitting different kinds of artifacts. In the future we will try to further decrease the amount of user effort for finding the trajectory. One way to go would be to find matching edges in two neighboring images, another would be to project e.g. a light- or laser beam onto the scene in the real world. The projection can then directly be used as trajectory.

References

[1] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, Boston, MA, USA, July 2006.

[2] G. N. Barnard. *Barnard's Photographic Views of the Sherman Campaign*. Press of Wynkoop and Hallenbeck, New York, NY, USA, 1866.

[3] D. Farin. *Automatic Video Segmentation Employing Object/Camera Modeling Techniques*. CIP-Data Library Technische Universiteit Eindhoven, Eindhoven, Netherlands, 2005. pp. 423–425.

[4] E. Kappe, A. Liers, H. Ritter, and J. Schiller. Low-power image transmission in wireless sensor networks using scatterweb technologies. In *Workshop on Broadband Advanced Sensor Networks*, San Jose (CA), USA, October 2004.

[5] J. Kim and S. Seitz. Multiperspective images from videos. *IEEE Computer Graphics and Applications*, pages 16–19, 11/12 2003.

[6] P. Rademacher and G. Bishop. Multiple-center-of-projection images. *Proc. SIGGRAPH*, 1998.

[7] S. Vallance and P. Calder. Multi-perspective images for visualization. In *Pan-Sydney Area Workshop on Visualization Information Processing*, 2002.

[8] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz. High speed video using a dense camera array. In *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[9] D. N. Wood, A. Finkelstein, J. F. Hughes, C. E. Thayer, and D. H. Salesin. Multiperspecive panoramas for cel animation. *Proc. SIGGRAPH*, 1997.