Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Multi perspective panoramic imaging

Thomas Haenselmann [*], Marcel Busse, Stephan Kopf, Thomas King, Wolfgang Effelsberg

University of Mannheim, Department of Applied Computer Science IV, A5(6), 68159 Mannheim, Germany

ARTICLE INFO

Article history:
Received 26 September 2006
Received in revised form 8 August 2007
Accepted 12 June 2008

Keywords:
Panoramic imaging
Multi-perspective
Image stitching
Warping

ABSTRACT

Panoramic images have only been feasible if all contributing image patches share a common center of projection. Then, they can be consolidated into a single image using perspective transforms. In contrast to that, we propose a novel non-linear warping scheme which allows the merging of multi-perspective images, thus taking advantage of scattered cameras. Therefore, a polygonal cut is defined in two source images to be merged. Usually, the layout of the cuts does not allow a user to stitch both images together naively. Thus, two convex combinations of a warped and a canonic coordinate system 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 inexpensive and available as scalar sensors 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 inexpensive CCD cameras [1–4]. 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 device [5].

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 descriptions of prior attempts to create panoramic images from movies. Section 3 suggests a basic warping scheme as a solution. In Section 4, we identify some shortcomings of the basic scheme which are solved by an extension. The evaluation in Section 5 shows some examples and analyzes the new degree of freedom, but also fundamental limitations, of multi-perspective still imaging and video in general. The outlook in Section 6 sketches future improvements to reduce the amount of human interaction.

2. Related work

In the context of our paper, we will distinguish between mono-perspective panoramic images and multi-perspective imaging.

2.1. Mono-perspective 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 [6]. Here, a series of photos was captured while rotating a tripod-mounted camera around its optical axis. The panoramic image was then obtained by aligning the photos next to one another.

In the 20th century, the so-called *rotating lens cameras* have been developed. They consist of a rotating lens which exposes a long strip of film during a rotation of up to 360°.

With the advent of digital cameras, panoramic imaging became popular with a larger audience. Here, views with a wide angle are produced by stitching together images of a normal aspect ratio of 4:3. Ideally, the images are produced with a tripod-mounted camera. This ensures a fixed focal point, also known as the *center of projection*. By rotating the camera around its vertical axis, only its viewing direction is altered. This means that the projection of the

[*] Corresponding author. Tel.: +49 621 181 2603; fax: +49 621 181 2601.
E-mail addresses: haenselmann@informatik.uni-mannheim.de (T. Haenselmann), effelsberg@informatik.uni-mannheim.de (W. Effelsberg).

0262-8856/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2008.06.008

three-dimensional world onto the CCD-chip will never change. The rotation itself only makes one part of the image disappear while another moves in. Unfortunately, this does not mean that images can be put together by a simple concatenation because the rotation changes the vanishing points within the images. This is most obvious in architectural photos, in which lines being parallel in the real world converge against a common vanishing point in the projection.

2.1.1. Tubular and spherical projections

Prior to stitching two images together, a perspective dewarping of one of them, or preferably even of both, has be carried out at the same time. 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 as shown in Fig. 1 [7,8].

It can be seen from the left side of the figure that a coordinate (x, y) in the image plane maps to an angle θ and the height h which are defined as Fig. 2

$$\theta = \arctan\left(\frac{x}{f}\right),$$
$$h = \frac{y}{\sqrt{x^2 + f^2}}. \tag{1}$$

The focal length is denoted with f.

The tubular projection is easy to calculate and use if the focal length is known and does not change. Unlike the spherical projection, it does not cope well with camera rotations other than around the vertical axis since this degree of freedom is not contained in the tubular model. An artifact of both approaches is the natural but disturbing fish eye-like appearance of the projected images. The spherical model allows both vertical and horizontal rotations. Unfortunately, the horizontal angle (the longitude) becomes numerically instable or even undefined at the poles of the sphere.

2.1.2. Eight-parameter model

In order to stitch images originating from smaller camera rotations, an eight-parameter planar projection is often used. It aims at placing all images into a single planar surface which eliminates the bent appearance of the aforementioned two approaches. The eight parameters enclose a rotation within the image plane, a perspective turn in the vertical and horizontal direction, a scaling factor and even a horizontal and vertical translation.

As the translation is non-linear, it can be written in matrix form only by means of the homogeneous coordinates introduced by Maxwell [9,10] and later applied to computer graphics by Roberts [11].

$$\begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ w \end{pmatrix}. \tag{2}$$

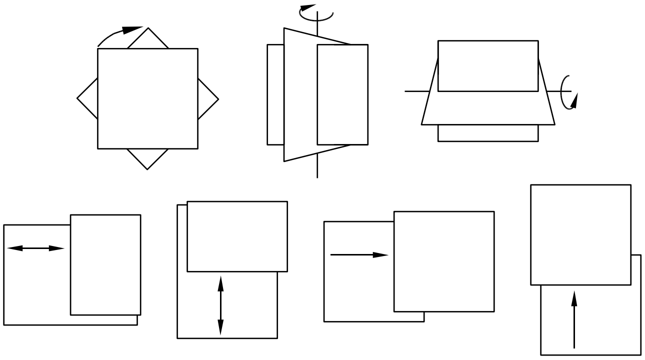


Fig. 2. Seven degrees of freedom are required for image alignment (from left top to right bottom): rotations within the image plane, horizontal and vertical perspective turns, horizontal and vertical scalings and translations in two directions.

The translation from homogeneous coordinates to image coordinates is easily accomplished by dividing by the third parameter, yielding:

$$I_x = \frac{xp_1 + yp_2 + p_3}{xp_7 + yp_8 + 1},$$
$$I_y = \frac{xp_4 + yp_5 + p_6}{xp_7 + yp_8 + 1}. \tag{3}$$

The determination of the eight parameters can be accomplished in two ways. One approach aims at varying the parameters according to a least square minimization [12]. The best parameter set is found if the overlapping parts of both images to be stitched match as well as possible, especially if the sum of their squared pixel-differences becomes minimal. Theoretically, this approach works automatically; however, in practice, the least squares optimization, which is usually implemented as a fixed-point iteration, converges only against a local minimum. This makes it crucial for the user to find a good initialization for the approach to work.

A more realistic method involves choosing four points in one image and the corresponding points in the overlap of the neighboring image. Since one set of points corresponds to two constraints in 2D, four points are enough to determine all parameters. By inserting feature points as (x, y) and corresponding feature points as (x', y') into Expression 2, the eight parameters can be determined directly. This method is fast and reliable, but needs a user to find four corresponding points.

Instead, a feature detector can be used to determine matching points automatically. Since perfectly matching points can hardly be expected from an algorithm, a greater number of them is determined. The model-parameters $p_1, \ldots, p_8$ are then optimized such that for every feature point (x_i, y_i) in one image, the corresponding one (x'_i, y'_i) is predicted as well as possible as shown in expression (4). The automatic approach can tolerate some outliers in a larger set of points.

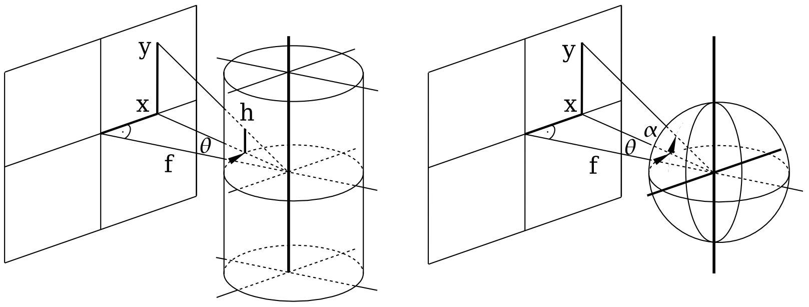


Fig. 1. Stitching panoramic images is done in a common space, which can be either tubular or spherical. The figure exemplifies the mapping of (x, y) coordinates from the image plane to angles and/or lengths.

$$\min_{\forall p_1,\ldots,p_8} \sum_{\forall i} \left(\left[\left(\frac{x_i p_1 + y_i p_2 + p_3}{x_i p_7 + y_i p_8 + 1} - x'_i \right)^2 + \left(\frac{x_i p_4 + y_i p_5 + p_6}{x_i p_7 + y_i p_8 + 1} - y'_i \right)^2 \right] \right). \tag{4}$$

Szeliski reports this method works well in images with evenly distributed feature points [13,14]. However, problems arise if the overlap is only sparsely scattered with feature points, or is too dense, like in textured regions.

The problems arising from finding appropriate feature points was addressed by Mann with a featureless approach which optimizes over the entire image rather than only over single points and which proved to converge stably, particularly in the presence of noise and light fluctuations [15]. Candocia extended the eight-parameter registration by simultaneously adjusting the tone of neighboring gray-scale images [16]. Davis developed a registration algorithm based on the Mellin transform (a close relative of the Fourier transform) which can cope with scenes that include moving objects [17]. After segmenting images into distinct regions, only a single region is chosen to contribute to the panoramic image, thus avoiding blurred moving objects.

The transforms described above try to map images into a single image space by means of relatively "rigid" affine or projective transforms. As we will see in Section 3, applying non-linear transforms enable images to be consolidated with significantly fewer disturbing artifacts.

2.1.3. Early work on changing viewpoints

Regardless of the type of projection used, whenever the focal point is fixed, the perspective on an object will not change. Intuitively, a part of an object which is hidden will never be revealed by the rotation of the camera. This changes in the case of *Multi-Perspective Panoramic Images*. Mann and Picard wrote early on about how to handle changing viewpoints in their work on "Virtual Bellows" for the special case of

- planar objects and different viewpoints or
- none-planar objects an fixed viewpoints [18].

Bellows cameras basically consist of separate film- and lens-boxes which are connected by a flexible bellows. Some of these cameras allow the photographer to change the angle of the plane in which the lens is mounted with respect to the plane of the film. This can be used to compensate for perspective distortions of flat objects like the front of a house. Since the distortion takes place solely in the image plane, it can also be achieved digitally, e.g., in order to produce panoramic or high-resolution images. Interested readers may want to refer to the *comparametric* project on sourceforge.net.[1] The project provides a toolkit for image mosaicing which even takes advantage of GPU-based hardware acceleration.

In contrast to that, what is proposed in Section 3 is particularly suited for handling parallax caused by multiple perspectives.

2.1.4. Lens distortion and mosaicing

Early work has been done by Sawhney and Kumar in the context of image mosaicing and lens distortion correction [19,20]. Their approach tackles the problem of image registration with a common center of projection. In contrast to prior approaches, they compensate for the lens distortion in a first step. Then, the result is mapped into a global coordinate system using an eight-parameter model. In the end, the images are aligned in the global coordinate system to create a panoramic result.

The compensation of the lens distortion creates better overall results. In particular, the distortion does not accumulate and intensify over longer chains of stitched images. The commonness with the multi-perspective approach presented in Section 3 is to consolidate all images into a joint coordinate system using non-linear transforms. The difference is that we focus on a more general problem allowing multiple viewpoints, which creates strong parallax.

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 [21]. 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 [22]. 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

[1] http://comparametric.sourceforge.net.

a straight path showing a long flat surface [23]. 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. The problems discovered by Agarwala et al. will be analyzed more formally in Section 5.

Rademacher produced similar panoramas [24]. 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.

Shum and He suggested the concept of concentric mosaics, which are panoramic images taken from a camera moving on a circular path [25]. Unlike in traditional panoramic image generation, a spinning disc is mounted on top of the tripod. The camera itself is mounted on the boundary of the disc and it is viewing into the direction of the tangent at the mount point. While the camera is moving along its concentric path on the disc, every captured image contributes its middle column of pixels to the panoramic image. In their work, the authors explore the results by generating ray-traced and real images.

Vallance and Calder generalized the approach by Shun and He by generating ray-traced 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 [26]. The benefit of the unnaturally appearing results is that opposing object-surfaces can be seen in a continuous image.

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.

2.2.1. Image metamorphosis

Though multi-perspective stitching has not yet been the focus of much research, Beier and Neely presented a close relative of our suggestion in the field of image metamorphosis [27].

In their work, the authors tackle a problem known as *morphing* or *image metamorphosis*. The aim is to transform one photo, usually showing a specific shape like, e.g., a face, into another. Rather than doing a classical *dissolve*, which is known from TV and video editing, morphing involves both: The dissolve of color values and a warping of a starting-image towards a target-image. Especially the combination of gradually changing color and of a shape that is changing at the same time produces highly realistic frames in-between.

Fig. 3 shows a sketch of the basic idea proposed by Beier and Neely. An interpolated image between a starting and a target image is to be produced. A classical example is to morph one face into another similar photo. In order to define a correspondence between two images, the facial features like the nose, the eyebrows and the silhouette of the head are marked with vectors, both in the starting and the target images, respectively. In all intermediate images, those vectors are simply interpolated linearly so that the vectors in the starting image converge against those in the target-image.

The image somewhere in the middle of the metamorphosis is called the interpolated image in the figure. The algorithm iterates over all pixels of the interpolated image and for each of them asks from where to take the corresponding color values from the source images. Note that the source will be the starting and the target-image each in turn.

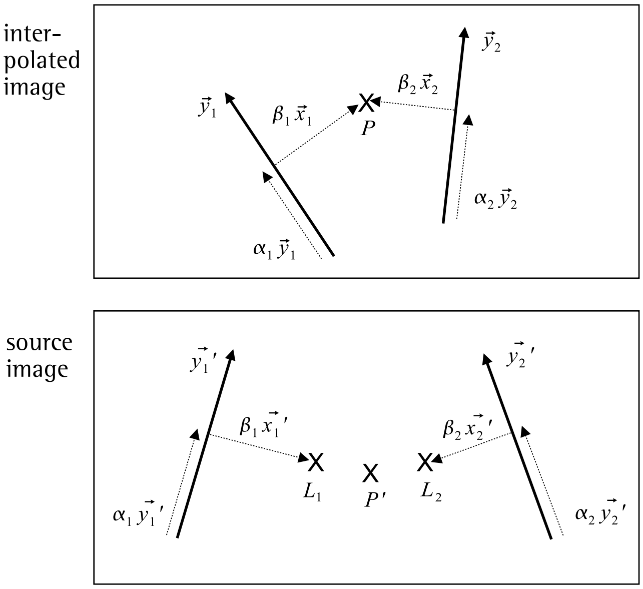


Fig. 3. Warping scheme by Beier and Neely, which introduced the idea of image warping in the context of image metamorphosis.

The sketch shows two of the vectors mentioned above which, e.g., mark some facial features. In 2D, each of those vectors (together with an orthogonal vector) defines a local base. In the interpolated image, a pixel at location P can be linearly combined by means of each local base. The two scalars α_1 and β_1 lead to location P based on the left coordinate system, and α_2 and β_2 lead to P based on the right one.

When bringing these two linear combinations to the source image, they will usually yield two different linear combinations, which are labeled as L_1 and L_2 in the lower part of Fig. 3. The final task is to combine those two linear combinations into a single one P'. The core contribution of Beier and Neely is to convex combine them in the following way:

$$P' = \left(1 - \frac{\beta_1}{\beta_1 + \beta_2}\right)L_1 + \left(1 - \frac{\beta_2}{\beta_1 + \beta_2}\right)L_2. \tag{5}$$

If P is close to the $\vec{y}_1$-axis in the interpolated image, then the weight of L_1 is greater and that of L_2 negligible. As P converges against the coordinate system on the right, L_2 becomes more dominant in the source image.

Note that the above example is only a simplified version for two feature vectors. However, the analog proceeding applies for an arbitrary number (thus replacing $\beta_1 + \beta_2$ by a larger sum of scalars, each of which originate from its own coordinate system).

For the application of morphing, the same convex combination has to be done once for the starting and for the target image. Finally, the two color values originating from these images are interpolated and displayed at location P in the interpolated image, which is generated this way, pixel by pixel.

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.

Fig. 4 shows a building from two different perspectives with a certain overlap. In conventional panoramic image generation,

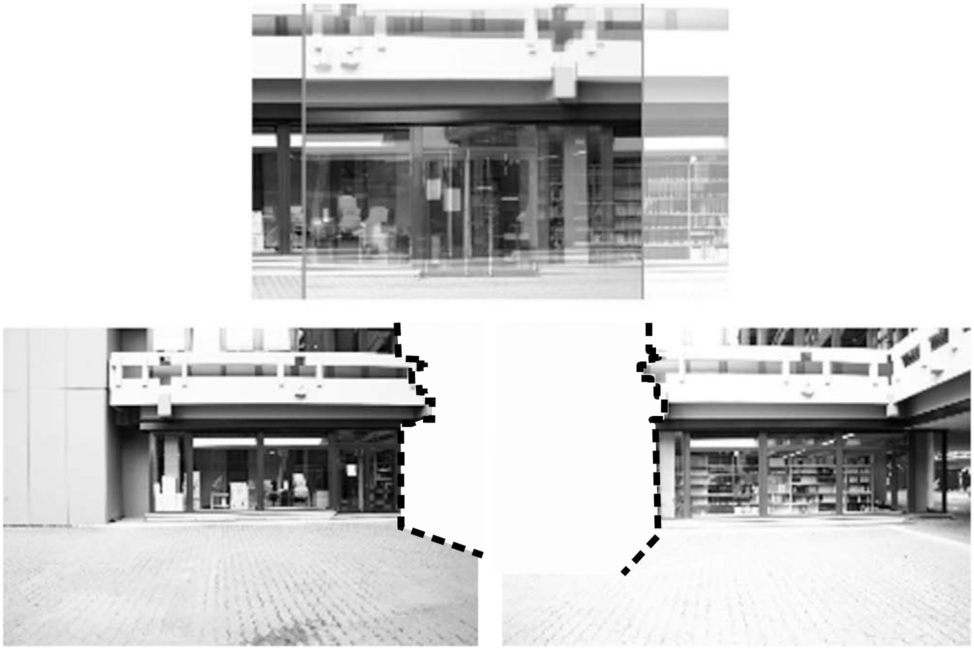


Fig. 4. Images taken from different perspectives will usually not match, regardless of how they are moved or distorted (see semi-transparent images on the left). The middle and right images are cut along a common trajectory. Theoretically, both images would fit together semantically but the layout of the cut allows no concatenation.

semi-transparent images are overlaid as shown in the upper part. 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, 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.

Fig. 5 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.

An example is shown in Fig. 6. The user defined four poly-lines, two of them in the middle image, and one each in the left and in the right image, respectively. Then, the merging took place in pairs, the left and middle images first. The result was merged with the right image. Finally, the object was segmented. Removing the background is useful whenever there is no obvious continuity in the backgrounds of the contributing images. E.g., the part of the line between A1 and A2 is of light color in the outer images, while the middle one is dark, which does not result in a convincing concatenation. The parts between A2 and A5, however, fit together well.

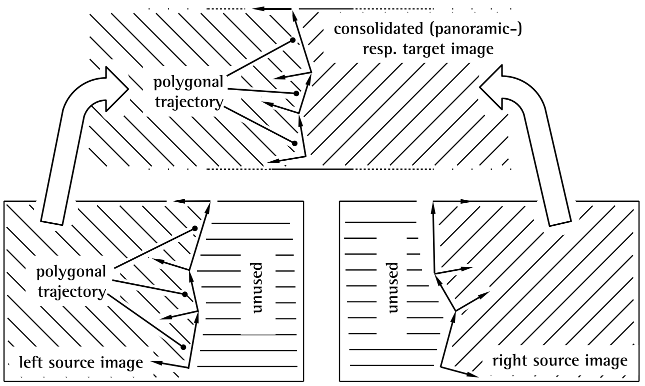


Fig. 5. The sketch shows which parts of the left and the right source image in the lower part of the figure contribute to the rendered panoramic image, shown above.

The software[2] can be obtained free of charge under the terms of the GPL.

So far, the trajectory was defined for the left and the right source image. 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 Fig. 7. The polygonal trajectories are piecewise linear. Each line segment can be considered as the y-axis

[2] http://www.informatik.uni-mannheim.de/~haensel/multi_persp_pan.tgz.

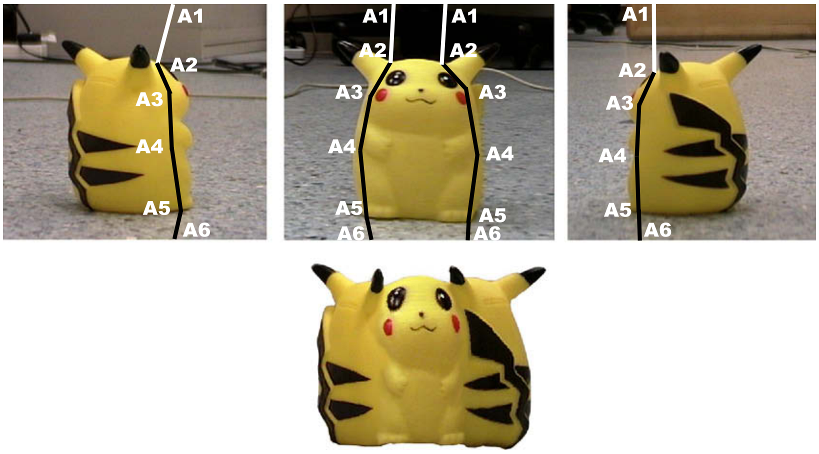


Fig. 6. Three images each with a differing perspective of the same object were stitched into a single panoramic image. The merging took place to the left and to the right along the user-defined poly-lines. The background was removed from the result since an object and its background can often not be consolidated into a continuous image at the same time.

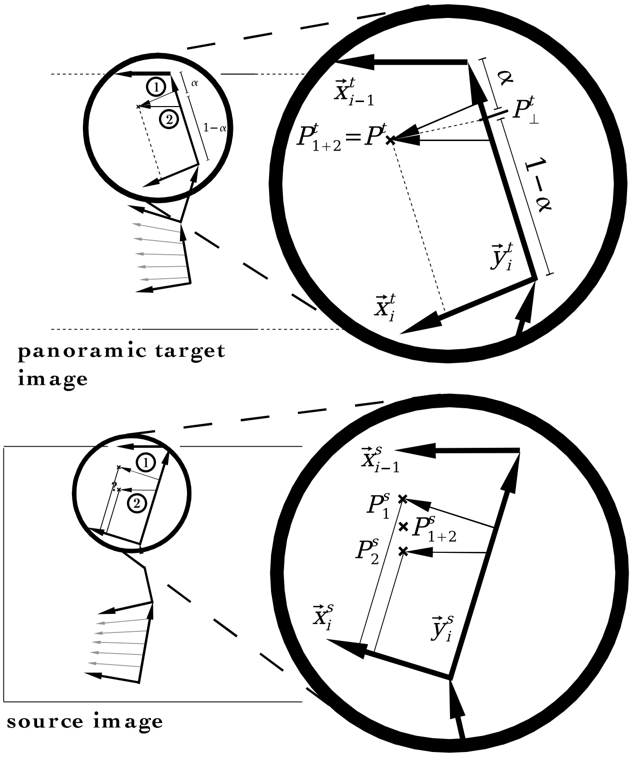


Fig. 7. Each pixel in the panoramic target image is linearly combined by two bases, span($\vec{x}^t_{i-1},\vec{y}^t_i$) and span($\vec{x}^t_i,\vec{y}^t_i$). The two linear combinations yield different pixels P^s_1 and P^s_2 in the source image as the spanning vectors point in different directions. They have to be merged into a single weighted average P^s_{1+2}.

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($\vec{x}^t_{i-1},\vec{y}^t_i$) and one by span($\vec{x}^t_i,\vec{y}^t_i$) where ^{t,s} in the exponents stand for *target* and *source*, symbolically. Both linear combinations can be considered as two interpretations P^t_1 and P^t_2 of the same location P^t. Both are based on the same ordinate $\vec{y}^t_i$ but they consist of different abscissas, namely $\vec{x}^t_{i-1}$ at the tip of the y-vector and $\vec{x}^t_i$ at y's base.

$$\begin{aligned} P^t_1 &= a_1\vec{x}^t_i + b_1\vec{y}^t_i,\\ P^t_2 &= a_2\vec{x}^t_{i-1} + b_2\vec{y}^t_i,\\ P^t_1 &\equiv P^t_2. \end{aligned} \tag{6}$$

In the lower part of Fig. 7, the two linear combinations are applied to the corresponding spanning vectors span($\vec{x}^s_{i-1},\vec{y}^s_i$) and span($\vec{x}^s_i,\vec{y}^s_i$) 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

$$\begin{aligned} P^s_1 &= a_1\vec{x}^s_i + b_1\vec{y}^s_i,\\ P^s_2 &= a_2\vec{x}^s_{i-1} + b_2\vec{y}^s_i,\\ P^s_1 &\not\equiv P^s_2. \end{aligned} \tag{7}$$

That is why the resulting points P^s_1 and P^s_2 have to be merged into a single coordinate P^s_{1+2} using a simple convex combination

$$P^s_{1+2} = (1-\alpha)P^s_1 + \alpha P^s_2. \tag{8}$$

The upper side of the figure indicates how the value α and $(1-\alpha)$ can be obtained. First, the point P^t in the target image is projected onto the ordinate. The projected point $P^t_\perp$ splits the ordinate $\vec{y}^t_i$ into two parts where $\alpha \in [0,1]$ denotes the ratio the split point defines. The effect of the convex weighting scheme is that the abscissa $\vec{x}^s_i$ becomes more important if P^t converges against it. If P^t moves into the opposite direction, the influence of $\vec{x}^s_i$ is diminished in favor of $\vec{x}^s_{i-1}$. The effect of the weighting scheme can be interpreted as a continuous warping from one abscissa to the next. In Fig. 7, the resulting intermediate axes are drawn in gray at the lower part of the trajectory. The basic warping scheme described above seems to solve the alignment problems in multi-view panoramic images, so we implemented it and ran the software on some test image sequences.

4. Extended warping

A typical result of the above-described simple warping scheme can be seen in Fig. 8. We identified four different kinds of artifacts, namely *expansions*, *contractions*, *undefined areas*, and *reflections* (see Fig. 9).

The emergence of expansions and contractions are most obvious. If two neighboring abscissas exhibit a large opening angle in the source image, but a smaller angle in the target image, this

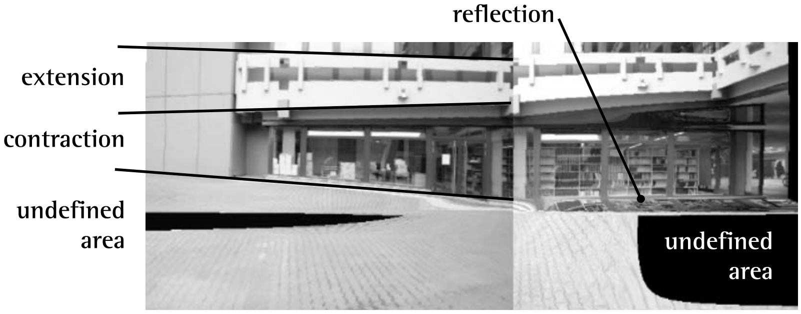


Fig. 8. The naive warping approach described in Section 3 creates four classes of artifacts, namely *expansions*, *contractions*, *undefined areas*, and *reflections*.

means that a large image patch in the source area will be squeezed into a small patch within the panoramic image, which will obviously lead to a contraction. In the opposite case, few source pixels have to contribute to many target pixels which results in an expansion which is getting even worse near the borders of the image.

The most disturbing artifacts are caused by undefined and mirrored regions. Undefined areas can be found whenever two neighboring abscissas intersect in the target image. Beyond the intersection, the order of the two vectors change. The one that used to be above its neighbor will be below afterwards, and vice versa. In the rendering process, a pixel has to determine which line of the polygonal trajectory (or which ordinate) is responsible for it. This is true for the one with the first abscissa (with a smaller index) being above the pixel and the second (with a higher index) being below. After the intersection this does not hold true for any of the line segments. Thus, no linear combination can be obtained and, as a consequence, no source pixel can be chosen.

Mirrored image patches emerge in the same intersection scenario, however this time the intersection takes place in the source image. The source pixel will be derived as usual, but beyond the intersection, once again, the order of the vectors changes which causes the image to swap vertically.

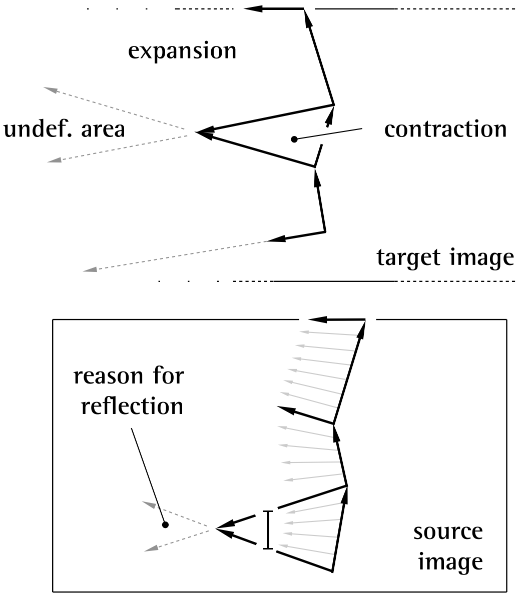


Fig. 9. Expansions originate from the fact that spanning vectors are converging in the source image and diverging in the target image. The complementary case is responsible for contractions. Crossing abscissas in the target image result in undefined areas while they produce mirrored image parts in the source image.

We will now analyze how these problems can be overcome. On one hand, the distortion of the source images cannot be avoided as this allows them to meet in the middle of the target image. On the other hand, the distortion becomes increasingly worse towards the borders of the image where it actually is no longer needed. This led to the idea of using another convex combination in a similar fashion as described above. So far, the warping of the image was done vertically by weighting neighboring abscissas as described in the previous section. Now we will perform the same process in the horizontal direction. Fig. 10 depicts the proceeding. Again, we have two points to merge into one by a factor β. The warped linear combination of a point is denoted with A in the figure. Another is labeled with B. B is a trivial linear combination by the image borders (which span the natural coordinate system of the image). We will refer to this as the natural base. Once again, both linear combinations A and B point to the same coordinate in the target image but to different locations in the source image. So the question arises how to weight both coordinates. The weighing factor is denoted with β. Let us consider the scan line through P starting at the left border and ending at the trajectory in the target image. The line is naturally split by P in the ratio $(1-\beta):\beta$. Eventually, a final location P_F is obtained in the source image by

$$P_F = \beta B + (1-\beta)A. \quad (9)$$

The more that point P converges against the left image border, the more dominant becomes the canonical coordinate B. Otherwise, the warped coordinate A is gaining a larger weight. The result of this weighting scheme is that the zigzag shape of the trajectory is getting straighter against the left side thus converging against the vertical border of the image.

Fig. 11 shows an improved version of Fig. 8. Most parts of the image appear far more natural, but some warping effects remain. They have been outlined with dashed lines on the left side. The naturalness increases smoothly from the middle of the image towards the borders. The speed of convergence can be adjusted by manipulating β. If the parameter is for example squared, the warped image converges much faster against the natural one. Therefore, a stronger local bending takes place near the trajectory. Depending on the image content, this can at times be more disturbing than distributing the distortion over a larger area.

Fortunately, the remaining artifacts can be compensated easily. In Fig. 11, 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

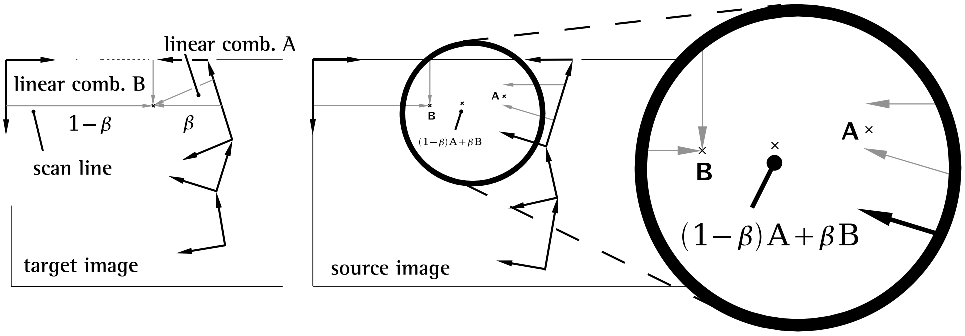


Fig. 10. Here, in the target image, every pixel is linearly combined once by the bent coordinate system (A) originating from the trajectory and by a canonical coordinate system (B) spanned by the natural borders of the image. In the source image, these two interpretations can again be weighted by a factor β. Decreasing values for β result in increasingly undistorted image coordinates.

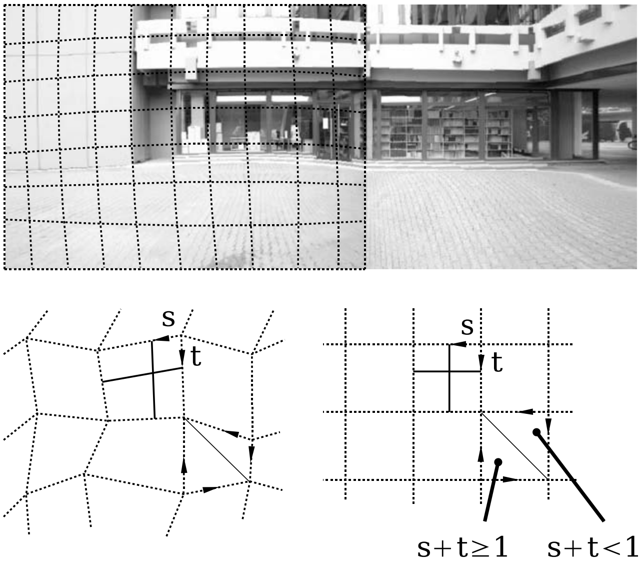


Fig. 11. The extended warping scheme creates a more natural result (upper part). The remaining bending is emphasized by dashed lines. Mapping this distorted grid, which is symbolized on the lower left to the regular one on the right of the figure, diminishes the residual errors.

half of the cell. Otherwise, a new linear combination has to be obtained again 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.[3] The final dewarping compensates for the lens-distortion as well.

5. Evaluation and limitations

Figs. 12 and 13 show examples of panoramic images with changing perspectives. Intuitively speaking, warping and stitching images work best if the object being shown can be unfolded, theoretically. This is often true for buildings. In particular, stitching works without problems if the camera follows a straight path. The reason is that succeeding images can be concatenated to existing ones if the above-mentioned trajectory exists and if it is not occluded from two neighboring viewpoints. We can even generalize feasible scenarios further by assuming that a camera must never be able to see the location of another camera. This situation is shown in Fig. 16. Here, the same plants appear multiple times from differing perspectives, according to the number of cameras. A consolidation of these views into a single continuous image is not possible.

Formally, it has to hold true for the surface of an object that every point should be uniquely addressable by two parameters only. This is true, for example, for cubes, spheres and tori. Then, the mapping of the object surface onto a flat image is unique. All other cases have to solve the mapping problem by ignoring some faces of an object and by producing a pseudo-continuous mapping. In any case, the trajectory for matching two images should be taken from the parameterizable parts of the surface.

Fig. 14 is an example of an object which cannot be unfolded perfectly. The leftmost eight images show a stuffed animal on a turntable. In each view, the dog was rotated by a multiple of 45°. The next four images (or the third column of pictures) are rendered images. Each of them consolidates its two left neighbors (from the same line) into one. The stitched surface of the object itself appears visually natural, while some artifacts can be perceived in the background and on the turntable (e.g., note the number of highlights). The two images in the fourth column combine their four stitched neighbors into two, and the final outcome is shown on the very right. It is composed of all eight images, whereas a part of the dog's head is duplicated.

For optimal stitching, neighboring images with similar perspectives on a scene or on an object are beneficial. As mentioned in the beginning, two images are always concatenated along a common poly-line which is visible from both perspectives. The smaller the difference between two perspectives, the higher the likelihood will be that a common line can be identified without any occlusion.

The lab setting using a turntable makes the evaluation against a theoretical optimum possible which is shown in Fig. 15. Here, the table was rotated in steps of about 3° each. Only a few pixel-columns in the center of the image were captured, each of which contributes about 3/320 to the resulting image. The 120 slices were simply put next to one another. The result distributes the distortion equally over the entire object. In contrast to the manually stitched version above, there are no highlights on the turntable. In addition, the background is continuous in the horizontal direction, since the same part is repeated in every image slice. Only the object on the turntable was moving. Small discontinuities can be seen on the back of the dog or the mouth line. As no warping was applied to the slices, a perfect continuity of the image cannot be expected.

Fig. 13 shows an example produced by four cameras which do not lie on a common line and which do not view in the same direction. Yet, the unwarped sculpture looks credible as each camera sees the solid object and none of the cameras face each

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

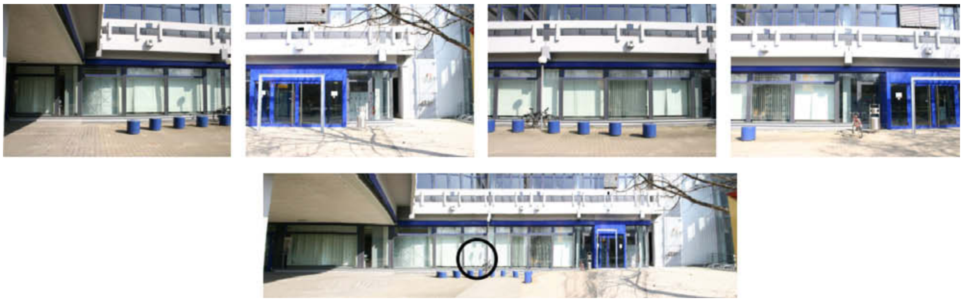


Fig. 12. The above view of the building cannot be accomplished using classical panorama techniques since the opposite construction prevents the camera from gaining enough distance. Despite the fact that the viewpoint changes four times, the image still appears credible. However, some artifacts described in Section 5 like the duplication of the street lamp's shadow (see black circle) cannot be avoided.

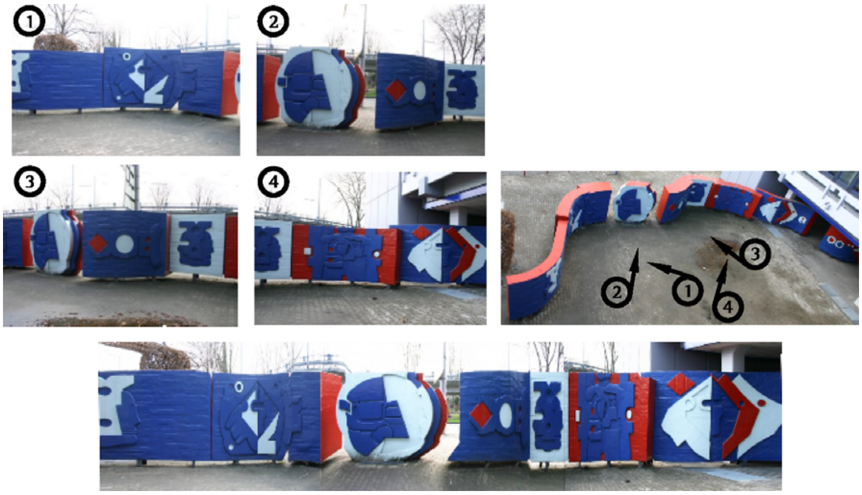


Fig. 13. The curved sculpture (overview on middle right image) was unrolled (resulting in bottom image) from several camera perspectives with changing locations and viewing directions. The numbered arrows in the overview image show the camera positions.

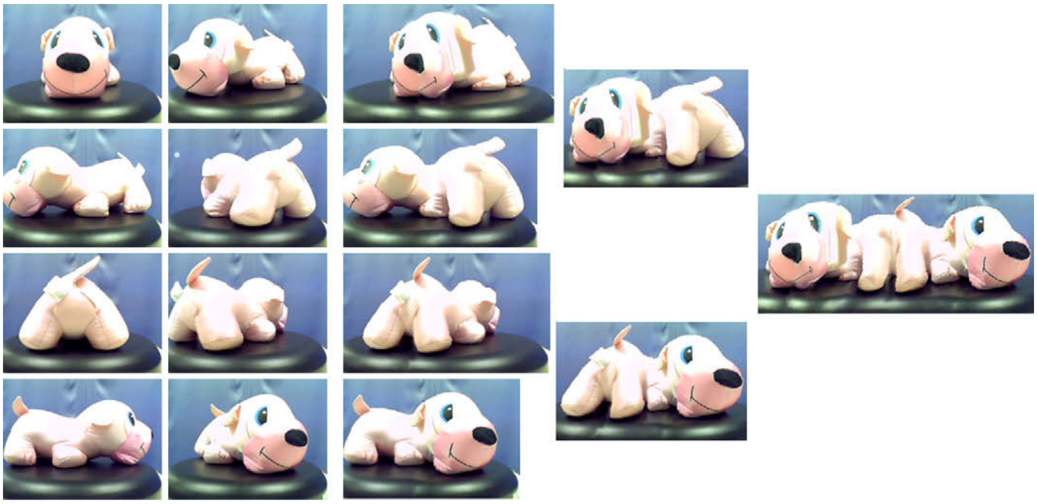


Fig. 14. The eight leftmost images were taken in steps of 45° each. Then, two were consolidated in turn. The resulting *object panorama* on the right shows the final result, which includes a slice of all eight original images.

other. The fact that the sculpture could theoretically be flattened into a plane is a sufficient precondition for producing a panoramic view.

In Fig. 16(b), two distant cameras look at a house. Merging the more distant views into the wide-angle view would be straightforward. It would simply result in a larger picture with differing resolutions throughout the image. However, the camera with the least distance to the house shows the door and parts of the interior of the inner right side. Here, no obvious way can be found to merge the *new* content into a global image, as the interior seen through the door would in any case have to overwrite existing parts of the global picture.

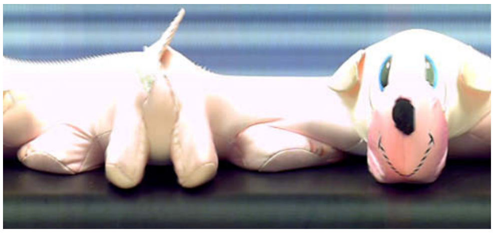


Fig. 15. The object was rotated in steps of 3° each. After each rotation, a small slice was added to the resulting image. In settings which allow the use of a turntable, an optimal result can be obtained, which spreads the distortion equally over the image.

The above-mentioned artifacts apply to still images and moving ones in the same way. An effect which would be more obvious in moving images is exemplified in Fig. 16(c). Here, two cameras face the front of a house. Their views are cut and merged to a continuous panoramic image, for example, like the one shown in Fig. 12.

A person walking from the left to the right between the building and the cameras, would first appear in the viewing volume of the left camera. After a while, she would walk inside the blind area. Within the panoramic video, this means the person would disappear behind a certain column of pixels without any reason. The house itself appears to be continuous for the viewer. After a while she would enter the right viewing volume which means that the person would suddenly appear after the same column of pixels mentioned before. For a human observer there is no obvious reason for the disappearance as no occluding obstacle can be seen. Yet, the effect is a natural consequence of merging two differing viewpoints. A traditional panoramic image does not exhibit this problem since viewing volumes are always neighboring with no intermediate gap.

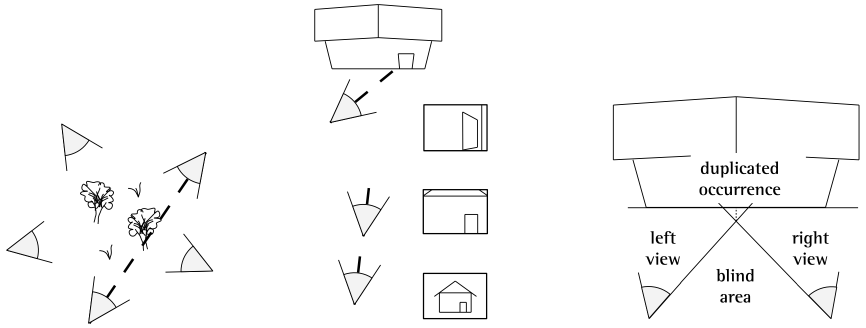
Another equally undesired but natural effect is the duplication of entities beyond intersecting viewing volumes. This is denoted as *duplicated occurrence* in Fig. 16. The panorama of the building was designed such that its front appears continuous. Likewise, the screens of the building in Fig. 12 have preserved their natural size and aspect ratio. However, objects behind the screen can be seen twice next to the intersecting line of the left and right image. Everything behind the intersection point of the boundaries of the viewing volumes can both be seen from the left and the right camera. So, by examining Fig. 12 in more detail the reader may notice that the shadow of the street lamp (marked with a black circle) is projected twice onto the white blinds behind the window. Theoretically, a person walking behind the window would appear natural in the beginning and then suddenly be duplicated in the area of *duplicated occurrence* before becoming singular again. A requirement stated above said the trajectory should lie in a part of the object's surface so as to be parameterized uniquely. In the case of the house and the window surface this does not hold true for the trajectory. It is chosen from a part of the building which is a surface in the front (the window) and in the back (the interior of the room) at the same time. Both levels cannot be addressed by a two-dimensional parameter space uniquely. Other opaque parts of the building would be suited better for choosing the trajectory.

5.1. Performance evaluation

As was reported by Beier et al. in the context of image metamorphosis, most of the time needed to produce an image is taken by the interaction between the user and the program [27].

Rendering a panoramic image requires to linearly combine every pixel in the panoramic image by the coordinate system above and below the pixel. This linear combination is then transferred either to the left or to the right source image. The two coordinates are then merged into a single one by means of the α and $(1 - \alpha)$ convex combination. The same is done in the horizontal direction, which requires a final convex combination.

Unlike the approach by Beier and Neely, where the time complexity depends on the number of pixels and the number of involved vectors, our approach depends on the number of pixels only, since all other quantities are constant. The (not yet carefully optimized) implementation we provide for download requires time in the order of magnitude of a second to render a panoramic image of two VGA-sized source images originating from the cheap CCD cameras we use on a typical office PC of the year 2007.



(a) Different perspectives of a single translucent object cannot be merged into a common view

(b) The uppermost view into the door shows a different scenario than the ones that can be seen by the frontal-facing cameras.

(c) Even two cameras facing different parts of a flat object produce blind areas or areas of duplicated appearance.

Fig. 16. Different artifacts produced by multi-perspective panoramas. (a) Different perspectives of a single translucent object cannot be merged into a common view. (b) The uppermost view into the door shows a different scenario than the ones that can be seen by the frontal-facing cameras. (c) Even two cameras facing different parts of a flat object produce blind areas or areas of duplicated appearance.

6. Conclusion and outlook

A new approach for producing panoramic images from photos with varying centers of projection is proposed. A trajectory has to be found in neighboring images which serves as a cut. A vertical warping scheme distorts neighboring images such that they fit together. Another horizontal warping process makes the distorted parts of the image near the cut converge against the natural image near the left and right borders of the panorama. Finally, artifacts originating from the warping and from the lens distortion are corrected.

So far, the approach requires some user interaction for finding an appropriate cut through the images and for dewarping the final result. The cut, or more precisely, the polygonal trajectory might also be found by means of feature points visible from two perspectives. An alternative and robust solution would be to project a colored light beam onto the objects in the real world and take photos from two perspectives. The beam itself can be used directly as a cut if it occurs without any occlusion in the two images.

Furthermore, the final dewarping requires the user to define the bent grid shown in Fig. 11. In fact, the grip is the antiderivative of the vectors spanned by the abscissas and ordinates originating from the trajectory in the panoramic image. It could be derived numerically without the need for manual interaction.

Acknowledgements

We want to thanks Mr. Allen Wrenn and Mrs. Weyerer for their suggestions and improvements which contributed significantly to the quality of this work.

References

[1] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Levoy, M. Horowitz, High performance imaging using large camera arrays, in: Proc. of SIGGRAPH, 2005.

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

[3] V. Vaish, B. Wilburn, N. Joshi, M. Levoy, Using plane and parallax for calibrating dense camera arrays, in: Proc. of CVPR 2004, Washington, DC, USA, 2004.

[4] B. Wilburn, M. Smulski, H.K. Lee, M. Horowitz, The light field video camera, in: Proc. of Media Processors 2002, SPIE Electronic Imaging 2002, San Jose, CA, USA, 2002.

[5] E. Kappe, A. Liers, H. Ritter, J. Schiller, Low-power image transmission in wireless sensor networks using scatterweb technologies, in: Workshop on Broadband Advanced Sensor Networks, San Jose, CA, USA, 2004.

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

[7] D. Farin, Automatic video segmentation employing object/camera modeling techniques, CIP-Data Library Technische Universiteit Eindhoven, Eindhoven, Netherlands, 2005.

[8] R. Szeliski, Video mosaics for virtual environments, IEEE Computer Graphics and Applications (1996) 22–30.

[9] E.A. Maxwell, Methods of Plane Projective Geometry Based on the Use of General Homogeneous Coordinates, Cambridge University Press, Cambridge, UK, 1946.

[10] E.A. Maxwell, General Homogeneous Coordinates in Space of Three Dimensions, Cambridge University Press, Cambridge, UK, 1951.

[11] L.G. Roberts, Homogeneous matrix representations and manipulations of *n*-dimensional constructs, Tech. Rep. Document MS 1405, Lincoln Laboratory, MIT, Cambridge, MA, USA, 1965.

[12] R. Szeliski, H.-Y. Shum, Creating full view panoramic image mosaics and environment maps, in: Proc. of the ACM SIGGRAPH, Sarasota, FL, USA, 1997.

[13] R. Szeliski, Image mosaicing for tele-reality applications, in: Proc. of the Second IEEE Workshop on Applications of Computer Vision, Sarasota, FL, USA, 1994, pp. 44–53.

[14] R. Szeliski, Handbook of Mathematical Models in Computer Vision – Image Alignment and Stitching, Springer, Cambridge, UK, 2005.

[15] S. Mann, R.W. Picard, Video orbits of the projective group: a simple approach to featureless estimation of parameters, TR 338, Massachusetts Institute of Technology, Cambridge, MA, also appears IEEE Trans. Image Proc., Sept 1997, vol. 6, no. 9 (see <http://n1nlf-1.eecg.toronto.edu/tip.ps.gz/>, 1995).

[16] F.M. Candocia, Simultaneous homographic comparametric alignment of multiple exposure-adjusted pictures of the same scene, IEEE Transactions on Image Processing 12 (2003) 1485–1494.

[17] J. Davis, Mosaics of scenes with moving objects, in: Proc. of the IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition (CVPR98), Santa Barbara, CA, USA, 1998, pp. 354–360.

[18] S. Mann, R.W. Picard, Virtual bellows: constructing high quality stills from video, in: Proc. of the IEEE Transactions on Image Processing, Austin, TX, USA, 1994.

[19] R. Kumar, P. Anandan, K. Hanna, True multi-image alignment and its application to mosaicing and lens distortion correction, in: Proc. of the ARPA Image Understanding Workshop, Monterey, CA, USA, 1994.

[20] H.S. Sawhney, R. Kumar, True multi-image alignment and its application to mosaicing and lens distortion correction, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI, vol. 21, IEEE Institute of Eletrical and Electronics, 1999, pp. 235–243.

[21] D.N. Wood, A. Finkelstein, J.F. Hughes, C.E. Thayer, D.H. Salesin, Multiperspective panoramas for cel animation, in: Proc. of SIGGRAPH.

[22] J. Kim, S. Seitz, Multiperspective images from videos, IEEE Computer Graphics and Applications (2003) 16–19.

[23] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, R. Szeliski, Photographing long scenes with multi-viewpoint panoramas, in: ACM Transactions on Graphics, Proc. of SIGGRAPH 2006, Boston, MA, USA, 2006.

[24] P. Rademacher, G. Bishop, Multiple-center-of-projection images, in: Proc. of the SIGGRAPH.

[25] H.-Y. Shum, L.-W. He, Rendering with concentric mosaics, in: Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999, pp. 299–306.

[26] S. Vallance, P. Calder, Multi-perspective images for visualization, in: Proc. of the Pan-Sydney Area Workshop on Visualization Information Processing, 2002.

[27] T. Beier, S. Neely, Feature-based image metamorphosis, in: SIGGRAPH'92 Proc. of the 19th Annual Conference on Computer Graphics and Interactive Techniques, vol. 26, 1992, pp. 35–42.