Homography Optimization for Consistent Circular Panorama Generation

Masatoshi Sakamoto[1], Yasuyuki Sugaya[2] and Kenichi Kanatani[1]

[1]Department of Computer Science, Okayama University, Okayama 700-8530, Japan
{sakamoto,kanatani}@suri.it.okayama-u.ac.jp
[2]Department of Information and Computer Sciences,
Toyohashi University of Technology, Toyohashi, Aichi 441-8480 Japan
sugaya@iim.ics.tut.ac.jp

Abstract. A 360° panorama is generated from images freely taken by a hand-held camera: images are pasted together and mapped onto a cylindrical surface. First, the lack of orientation information is overcome by invoking "oriented projective geometry". Then, the inconsistencies arising from a 360° rotation of the viewing direction are resolved by optimizing all the homographies between images simultaneously, using Gauss-Newton iterations based on a Lie algebra representation. The effectiveness of our method is demonstrated using real images.

1 Introduction

A *panorama* is an image with a large angle of view, giving viewers an impression as if they were in front of a real scene. In this paper, we consider a panorama that covers all 360° directions around the viewer. A typical approach for realizing this is to take images with a special optical system such as an fish-eye lens camera, a mirror-based omnidirectional camera, a composite multicamera system, or a camera rotation mechanism [16]. Such optical systems have developed for autonomous robot navigation applications [17].

Another approach is to paste multiple images together, known as *image mosaicing* [13, 15]. This technique has been studied in relation to such video image processing as image coding, image compression, background extraction, and moving object detection [2, 3, 10, 12].

However, the aim of this paper is not such industrial or media applications. We consider a situation where travelers take pictures around them using an ordinary digital camera, go home, and create panoramic images for personal entertainment. The purpose of this paper is to present a software system that allows this without using any special device or requiring any knowledge about the camera and the way the pictures were taken.

The principle of image mosaicing is simple. If we find four or more corresponding points between two images, we can compute the *homography* (or *projective transformation*) that maps one image onto the other. Hence, we can warp one image according to the computed homography and past it onto the other image. Continuing this, we can create a panoramic image.

L.-W. Chang, W.-N. Lie, and R. Chang (Eds.): PSIVT 2006, LNCS 4319, pp. 1195–1205, 2006.

However, the images to be pasted distort as we proceed and diverge to infinity when the viewing direction changes by 90°. This can be avoided if we map the images onto a cylindrical surface and unfold it. For this, however, we need to know the orientations of the cameras that took the individual images. That information would be obtained if we used a special device such as an omnidirectional camera or a camera rotation mechanism, but we are assuming that no knowledge is available about camera orientations.

We resolve this difficulty by noting that even though a homography-based panorama cannot be *displayed* on a planar surface beyond a ±90° range, it is nevertheless *mathematically defined* over the entire 360° range. If we invoke the formalism of *oriented projective geometry* [11, 14], we can consistently define the pixel values in all directions expressed in homogeneous coordinates, which can then be mapped onto a cylindrical surface.

However, another critical issue arises: if we warp one image and successively paste it onto another, the final image may not agree with the initial image due to accumulated errors. To overcome this, we present a numerical scheme for optimizing all the homographies between images simultaneously subject to the condition that no inconsistency arises. This can be done by using Gauss-Newton iterations based on a Lie algebra representation. We demonstrate the effectiveness of our method using real images.

2 Panorama Generation using Homographies

2.1 Homographies

As is well known, images taken by a camera rotating around the center of the lens are related to each other by *homographies* (*projective transformations*). This holds for whatever camera motion if the scene is planar or is sufficiently far away. In whichever case, let (x, y) be a point in one image and (x', y') the corresponding point in another. If we represent these by 3-D vectors

$$\boldsymbol{x} = \begin{pmatrix} x/f_0 \\ y/f_0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{x}' = \begin{pmatrix} x'/f_0 \\ y'/f_0 \\ 1 \end{pmatrix}, \tag{1}$$

where f_0 is an arbitrary constant, the homography relationship is written in the form

$$\boldsymbol{x}' = Z[\boldsymbol{H}\boldsymbol{x}]. \tag{2}$$

Here, $Z[\,\cdot\,]$ denotes scale normalization to make the third component 1, and $\boldsymbol{H}$ is a 3×3 nonsingular matrix determined by the relative motion of the camera and its intrinsic parameters. For simplicity, we call the matrix $\boldsymbol{H}$ also a "homography". As Eq. (2) implies, the absolute magnitude and the sign of the matrix $\boldsymbol{H}$ are indeterminate. If we replace the constant f_0 in Eqs. (1) by another value $\tilde{f}_0$, the matrix $\boldsymbol{H}$ in Eq. (2) changes into

$$\tilde{\boldsymbol{H}} = \mathrm{diag}(1, 1, \frac{\tilde{f}_0}{f_0})\boldsymbol{H}\mathrm{diag}(1, 1, \frac{f_0}{\tilde{f}_0}), \tag{3}$$

where diag($\cdots$) denotes the diagonal matrix with diagonal elements $\cdots$ in that order).

2.2 Optimal Estimation of Homographies

Equation (2) states that vectors $\boldsymbol{x}'$ and $\boldsymbol{H}\boldsymbol{x}$ are parallel to each other, so it is equivalently rewritten as

$$\boldsymbol{x}' \times \boldsymbol{H}\boldsymbol{x} = \boldsymbol{0}. \tag{4}$$

Given N corresponding points (x_α, y_α) and (x'_α, y'_α), $\alpha = 1, ..., N$, between two images, we let $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}'_\alpha$ be their vector representations in the form of Eqs. (1), and $\bar{\boldsymbol{x}}_\alpha$ and $\bar{\boldsymbol{x}}'_\alpha$ their true positions in the absence of noise. If we regard the uncertainty of the x and y coordinates of each point as random Gaussian noise of mean 0 and a constant standard deviation, statistically optimal estimation of the homography $\boldsymbol{H}$ reduces to the minimization of

$$J = \frac{1}{2}\sum_{\alpha=1}^{N} \|\boldsymbol{x}_\alpha - \bar{\boldsymbol{x}}_\alpha\|^2, \tag{5}$$

subject to the constraint

$$\bar{\boldsymbol{x}}'_\alpha \times \boldsymbol{H}\bar{\boldsymbol{x}}_\alpha = \boldsymbol{0}. \tag{6}$$

Eliminating the constraint by introducing Lagrange multipliers and ignoring higher order error terms, we can rewrite Eq. (5) in the following form[1] [6]:

$$J = \frac{1}{2}\sum_{\alpha=1}^{N} (\boldsymbol{x}'_\alpha \times \boldsymbol{H}\boldsymbol{x}_\alpha, \boldsymbol{W}_\alpha(\boldsymbol{x}'_\alpha \times \boldsymbol{H}\boldsymbol{x}_\alpha)), \tag{7}$$

$$\boldsymbol{W}_\alpha = \Big(\boldsymbol{x}'_\alpha \times \boldsymbol{H}\boldsymbol{P}_{\mathbf{k}}\boldsymbol{H}^\top \times \boldsymbol{x}'_\alpha + (\boldsymbol{H}\boldsymbol{x}_\alpha) \times \boldsymbol{P}_{\mathbf{k}} \times (\boldsymbol{H}\boldsymbol{x}_\alpha)\Big)^-. \tag{8}$$

Here, $(\,\cdot\,)^-$ denotes pseudoinverse, and $\boldsymbol{P}_{\mathbf{k}}$ is the following projection matrix:

$$\boldsymbol{P}_{\mathbf{k}} = \mathrm{diag}(1, 1, 0). \tag{9}$$

In this paper, we denote by $\boldsymbol{u} \times \boldsymbol{A}$ the matrix whose columns are the vector products of $\boldsymbol{u}$ and the columns of the matrix $\boldsymbol{A}$, and by $\boldsymbol{A} \times \boldsymbol{v}$ the matrix whose rows are the vector products of $\boldsymbol{v}$ and the rows of $\boldsymbol{A}$.

2.3 Panorama Generation

Suppose the user holds a camera roughly horizontally and takes pictures around him roughly at an equal angle. We assume that the scene is sufficiently far away. Since no mechanical device is used, this is merely an approximation. We assume that the focal length is unknown and different from picture to picture.

The user first specifies corresponding points between adjacent images. Various automatic matching techniques have been proposed [8, 9, 18], but some mismatches are unavoidable using them. In our experiments, we manually selected corresponding points.

[1] The source code of the program that minimizes Eq. (7) by a technique called *renormalization* [7] is available at `http://www.suri.it.okayama-u.ac.jp`

Fig. 1. Successively warping and pasting images using homographies.

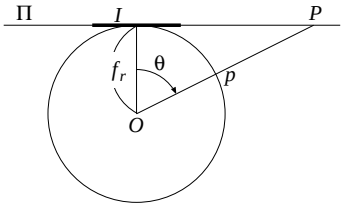

Fig. 2. The rectangular region $\mathcal{I}$ on the tangent plane Π to the cylinder.

Then, we compute the homography $\boldsymbol{H}$ between two neighboring images and paste one image onto the other via Eq. (2). Figure 1 is a part of the panoramic image thus created. We can see that images distort more as the viewing direction moves; the distortion diverges to infinity when the viewing direction is orthogonal to its initial orientation.

3 Cylindrical Panorama Generation

3.1 Mapping onto a Cylindrical Surface

The image divergence is avoided if we map the input images onto a cylindrical surface around the viewpoint and unfold it. However, the camera orientation information is missing. This is resolved by invoking *oriented projective geometry* [11, 14]. First, we do the following preparation:

- Imagine a hypothetical cylinder of radius f_r around the viewpoint O and define a (θ, h) cylindrical coordinate system (θ around the circumference and h in the axial direction). A point with cylindrical coordinates (θ, h) is unfolded onto a point with Cartesian coordinates $(f_r\theta, h)$.
- Let Π be the plane tangent to the cylinder along the line $\theta = 0$. Define an xy coordinate system on it such that the x-axis coincides with the line $h = 0$.
- Define an xy coordinate system on each input image such that the origin is at the center of the frame with the x-axis extending upward and the y-axis rightward. We identify this xy coordinate system with the xy coordinate system on Π and define on Π a rectangular region $\mathcal{I}$ of the same size as input images centered on $(\theta, h) = (0,0)$ (Fig. 2).
- Number the input images in the order of adjacency, and compute the homography $\boldsymbol{H}_{k(k+1)}$ that maps the kth image onto the $(k+1)$th image from

Fig. 3. Circular panorama corresponding to Fig. 1.

the specified corresponding points, $k = 1, ..., M$, where we use f_r in the place of f_0 in Eqs. (1) (see Eq. (3)). We choose the sign[2] of $\boldsymbol{H}_{k(k+1)}$ so that $\det \boldsymbol{H}_{k(k+1)} > 0$.

Then, we compute the pixel value of each point p with (discretized) cylindrical coordinates (θ, h) as follows (Fig. 2):

1. Compute the intersection P of the plane Π with the line passing through the viewpoint O and the point p on the cylinder, using homogeneous coordinates.
2. If P is inside the region $\mathcal{I}$ and *if the vectors* $\vec{Op}$ *and* $\vec{OP}$ *have the same orientation*, copy the pixel value[3] of P in the first image to p.
3. Else, let p' be the point on the cylinder such that[4] $\vec{Op'} \overset{+}{\simeq} \boldsymbol{H}_{12}\vec{Op}$, and compute the intersection P' of Π with the line passing through O and p'.
4. If P' is inside the region $\mathcal{I}$ and *if the vectors* $\vec{Op'}$ *and* $\vec{OP'}$ *have the same orientation*, copy the pixel value of P in the second image to p.
5. Else, let p'' be the point on the cylinder such that $\vec{Op''} \overset{+}{\simeq} \boldsymbol{H}_{23}\vec{Op'}$, and compute the intersection P'' of Π with the line passing through O and p''.
6. If P'' is inside the region $\mathcal{I}$ and *if the vectors* $\vec{Op''}$ *and* $\vec{OP''}$ *have the same orientation*, copy the pixel value of P in the third image to p.
7. Repeat the same process over all the images and stop. If no pixel value is obtained, the value of p is undefined.

The radius f_r of the cylinder is arbitrary in principle. However, if we require that the viewing direction should agree with the physical direction, e.g., two viewing directions that make 30° actually make 30° in the scene, the radius f_r should coincide with the focal length f_1 for the first image. This is because the mapping onto the cylinder starts with the first image; the subsequent images are successively warped so as to agree with it. The computation of the focal lengths for all the images will be described in Sec. 4. Figure 3 is a circular panorama thus generated using the images for Fig. 1.

Since the first image is mapped onto the cylinder, next the second image onto the rest of the cylinder, then the third, and so on, images with smaller numbers look "above" images with larger numbers (we can reverse the order, of course).

[2] In practice, it is sufficient if the (33) element is positive. If it is 0, the image origin is mapped to infinity. This does not happen between two overlapping images.

[3] The pixel value of a point with non-integer coordinates is bilinearly interpolated from surrounding pixels.

[4] The relation $\overset{+}{\simeq}$ denotes that one side is a multiple of the other side by a *positive* number.

Fig. 4. If neighboring images are successively pasted, the final image does not correctly match the initial image.

However, this makes the last image "below" the first image. For consistency, we place the last image above the first image where they overlap (we omit the details).

3.2 Oriented Projective Geometry

In the above procedure, the intersection P of the tangent plane Π is computed in homogeneous coordinates, so no computational failure occurs if P is at infinity.

In the standard projective geometry, a point P on the plane Π is identified with the "line of sight" l passing through P and the viewpoint O (the point P is regarded as located at infinity if the line l is parallel to Π). Since the line of sight l is not oriented, no distinction can be made between "in front of" or "behind" the viewpoint O. It is shown, however, that almost all properties of projective geometry is preserved if the line of sight is oriented [14]. This "oriented projective geometry" was found to be very useful for computer vision applications [11]. In the preceding procedure, we utilized this framework when we signed $\boldsymbol{H}_{k(k+1)}$ so that $\det \boldsymbol{H}_{k(k+1)} > 0$ and compared the "orientations" of the vectors $\vec{Op}$ and $\vec{Op'}$, etc.

4 Simultaneous Homography Optimization

4.1 Discrepancies of the Circular Mapping

If we compute the homographies between two images independently for all pairs, the final image does not necessarily match the initial image correctly, as shown in Fig. 4. This is because due to the accumulation of numerical errors and image distortions, the composite mapping

$$\boldsymbol{H}_{M1}\boldsymbol{H}_{(M-1)M}\boldsymbol{H}_{(M-2)(M-1)}\cdots\boldsymbol{H}_{23}\boldsymbol{H}_{12} \tag{10}$$

does not necessarily define the identity mapping (M is the number of images).

This is not a serious problem if the purpose of the circular panorama is simply for displaying the unfolded image. A more important application is, however, to let the user feel virtual reality by interactively moving the scene as the viewer changes the viewing direction[5]. This can be don by remapping, each time the

[5] The viewing direction is controlled by a mouse, keyboard, or a joystick. If the user wares a head-mount display (HMD), the head orientation can be measured from the signal it emits.

user specifies the viewing direction, the corresponding part of the cylinder onto its tangent plane[6]. For such applications, no inconsistency is allowed anywhere around the cylinder.

To resolve this, Shum and Szeliski [13] introduced a postprocessing for distributing the inconsistency equally over all image overlaps. Here, we adopt a more consistent approach: we optimize all the homographies simultaneously subject to the constraint that Eq. (10) define the identity map. As a byproduct, the orientations and the focal lengths of the cameras that took the input images are optimally estimated.

4.2 Parameterization of Homographies

While a general homography has 8 degrees of freedom[7], the homography arising from camera rotation and focal length change has only 5 degrees of freedom[8]. If we take the kth image with focal length f, rotate the camera around the lens center by $\boldsymbol{R}$ (rotation matrix), and take the $(k+1)$th image with focal length f', the homography $\boldsymbol{H}_{k(k+1)}$ that maps the kth image onto the $(k+1)$th image has the form

$$\boldsymbol{H}_{k(k+1)} = \mathrm{diag}(1, 1, \frac{f_0}{f_{k+1}})\boldsymbol{R}_{k(k+1)}^{\top}\mathrm{diag}(1, 1, \frac{f_1}{f_0}). \tag{11}$$

Equation (10) defines the identity mapping if and only if

$$\boldsymbol{R}_{12}\boldsymbol{R}_{23}\cdots\boldsymbol{R}_{(M-1)M}\boldsymbol{R}_{M1} = \boldsymbol{I}, \tag{12}$$

where $\boldsymbol{I}$ is the unit matrix. So, we minimize (cf. Eq. (7))

$$J = \frac{1}{2}\sum_{k=1}^{M}\sum_{\alpha=1}^{N_{k(k+1)}}(\boldsymbol{x}_\alpha^{k+1} \times \boldsymbol{H}_{k(k+1)}\boldsymbol{x}_\alpha^k, \boldsymbol{W}_\alpha^{k(k+1)}(\boldsymbol{x}_\alpha^{k+1} \times \boldsymbol{H}_{12}\boldsymbol{x}_\alpha^k)), \tag{13}$$

subject to the constraint that all $\boldsymbol{H}_{k(k+1)}$ have the form of Eq. (11) in such a way that Eq. (12) is satisfied. In Eq. (13), we assume that $N_{k(k+1)}$ points $\boldsymbol{x}_\alpha^k$ in the kth image correspond to the $N_{k(k+1)}$ points $\boldsymbol{x}_\alpha^{k+1}$ in the $(k+1)$th image, and the subscript k is computed modulo M. The matrix $\boldsymbol{W}_\alpha^{k(k+1)}$ is the value of $\boldsymbol{W}$ in Eq. (8) for the kth and the $(k+1)$th images.

Through Eq. (11), the function J in Eq. (13) is regarded as a function of the focal lengths f_1, f_2, ..., f_M and the rotations $\boldsymbol{R}_{12}$, $\boldsymbol{R}_{23}$, ..., $\boldsymbol{R}_{M1}$; we minimize J with respect to them subject to Eq. (12).

---

[6] A well known such system is QuickTime VR [1], for which input images are taken using a special camera rotation mechanism.

[7] The 3×3 matrix $\boldsymbol{H}$ has 9 elements, but there is an overall scale indeterminacy.

[8] The camera rotation matrix $\boldsymbol{R}$ has 3 degrees of freedom, to which are added the focal lengths f and f' before and after the camera rotation.

4.3 Lie Algebra Approach

Each rotation $\boldsymbol{R}_{k(k+1)}$ is specified by three parameters, but using a specific parameterization such as the Euler angles θ, ϕ, and ψ complicates the equation. So, we adopt the well known method of Lie algebra [4, 5]. Namely, instead of directly parameterizing $\boldsymbol{R}_{k(k+1)}$, we specify its "increment[9]" in each step. To be specific, we exploit the fact that a small change of rotation $\boldsymbol{R}_{k(k+1)}$ has the following form [4, 5]:

$$\boldsymbol{R}_{k(k+1)} + \boldsymbol{\omega}_{k(k+1)} \times \boldsymbol{R}_{k(k+1)} + \cdots. \tag{14}$$

Here, $\cdots$ denotes terms of order 2 or higher in $\boldsymbol{\omega}_{k(k+1)}$. If we replace $\boldsymbol{R}_{k(k+1)}$ in $\boldsymbol{H}_{k(k+1)}$ by Eq. (14), Eq. (13) can be regarded as a function of f_1, ..., f_M and $\boldsymbol{\omega}_{12}$, ..., $\boldsymbol{\omega}_{M1}$. After minimizing it with respect to them, the rotations $\boldsymbol{R}_{k(k+1)}$ are updated by

$$\boldsymbol{R}_{k(k+1)} \leftarrow \mathcal{R}(\boldsymbol{\omega}_{k(k+1)})\boldsymbol{R}_{k(k+1)}, \tag{15}$$

where $\mathcal{R}(\boldsymbol{\omega})$ denotes the rotation around axis $\boldsymbol{\omega}$ by angle $\|\boldsymbol{\omega}\|$.

The derivatives of J with respect to f_1, ..., f_M and $\boldsymbol{\omega}_{12}$, ..., $\boldsymbol{\omega}_{M1}$ can be analytically calculated (we omit the details). If we introduce the Gauss-Newton approximation, we can also calculate the second derivatives in simple analytic forms (we omit the details). It seems, therefore, that we can minimize J by Gauss-Newton iterations. However, there is a serious difficulty in doing this.

4.4 Alternating Optimization Approach

We need to enforce the constraint of Eq. (12). To a first approximation, Eq. (12) is expressed in the following form (we omit the details):

$$\boldsymbol{\omega}_{12} + \boldsymbol{R}_{12}\boldsymbol{\omega}_{23} + \boldsymbol{R}_{12}\boldsymbol{R}_{23}\boldsymbol{\omega}_{34} + \cdots + \boldsymbol{R}_{12}\boldsymbol{R}_{23}\cdots\boldsymbol{R}_{(M-1)M}\boldsymbol{\omega}_{M1} = \boldsymbol{0}. \tag{16}$$

A well known strategy for constrained optimization is the *method of projection*: the parameters are incremented without considering the constraint and then projected onto the constraint surface in the parameter space. However, if we try to minimize Eq. (13) without considering Eq. (16), *the solution is indeterminate* (the Hessian has determinant 0).

We resolve this difficulty by adopting the *alternate optimization*. Namely, Eq. (13) is first minimized with respect to f_1, ..., f_M with $\boldsymbol{R}_{12}$, ..., $\boldsymbol{R}_{M1}$ fixed. Then, the result is minimized with respect to $\boldsymbol{\omega}_{12}$, ..., $\boldsymbol{\omega}_{M1}$ with f_1, ..., f_M fixed. This time, the solution is unique because the Hessian of J with respect to $\boldsymbol{\omega}_{12}$, ..., $\boldsymbol{\omega}_{M1}$ alone is nonsingular.

Next, Eq. (16) is imposed by projection in the form of

$$\hat{\boldsymbol{\omega}}_{k(k+1)} = \boldsymbol{\omega}_{k(k+1)} - \Delta\boldsymbol{\omega}_{k(k+1)}. \tag{17}$$

[9] The linear space defined by such (mathematically infinitesimal) increments is called the *Lie algebra so*(3) of the group of rotations $SO(3)$ [4].

Fig. 5. After the simultaneous optimization of homographies, the discrepancy in Fig. 4(b) disappears.

The correction $\Delta\boldsymbol{\omega}_{k(k+1)}$ is determined as follows. The condition for Eq. (17) to satisfy Eq. (16) is written as

$$\boldsymbol{S}\Delta\tilde{\boldsymbol{\omega}} = \boldsymbol{e}, \tag{18}$$

where we define

$$\begin{aligned} \boldsymbol{S} &= \left(\boldsymbol{I}\ \boldsymbol{R}_{12}\ \boldsymbol{R}_{12}\boldsymbol{R}_{23} \cdots \boldsymbol{R}_{12}\boldsymbol{R}_{23} \cdots \boldsymbol{R}_{(M-1)M} \right), \\ \Delta\tilde{\boldsymbol{\omega}} &= \left(\Delta\boldsymbol{\omega}_{12}^{\top}\ \Delta\boldsymbol{\omega}_{23}^{\top} \cdots \Delta\boldsymbol{\omega}_{M1}^{\top} \right)^{\top}, \\ \boldsymbol{e} &= \boldsymbol{\omega}_{12} + \boldsymbol{R}_{12}\boldsymbol{\omega}_{23} + \boldsymbol{R}_{12}\boldsymbol{R}_{23}\boldsymbol{\omega}_{34} + \cdots + \boldsymbol{R}_{12}\boldsymbol{R}_{23} \cdots \boldsymbol{R}_{(M-1)M}\boldsymbol{\omega}_{M1}. \end{aligned} \tag{19}$$

Introducing Lagrange multipliers, we can obtain the solution $\Delta\tilde{\boldsymbol{\omega}}$ that minimizes $\|\Delta\tilde{\boldsymbol{\omega}}\|$ subject to Eq. (18) in the form

$$\Delta\tilde{\boldsymbol{\omega}} = \boldsymbol{S}^{\top}(\boldsymbol{S}\boldsymbol{S}^{\top})^{-1}\boldsymbol{e}. \tag{20}$$

From this, the correction formula of Eq. (17) reduces to the following form (we omit the details):

$$\hat{\boldsymbol{\omega}}_{12} = \boldsymbol{\omega}_{12} - \frac{1}{M}\boldsymbol{e},\ \ \hat{\boldsymbol{\omega}}_{23} = \boldsymbol{\omega}_{23} - \frac{1}{M}\boldsymbol{R}_{12}^{\top}\boldsymbol{e},\ \ \hat{\boldsymbol{\omega}}_{34} = \boldsymbol{\omega}_{34} - \frac{1}{M}\boldsymbol{R}_{12}^{\top}\boldsymbol{R}_{23}^{\top}\boldsymbol{e},$$

$$..., \qquad \hat{\boldsymbol{\omega}}_{M1} = \boldsymbol{\omega}_{M1} - \frac{1}{M}\boldsymbol{R}_{12}^{\top}\boldsymbol{R}_{23}^{\top} \cdots \boldsymbol{R}_{(M-1)M}^{\top}\boldsymbol{e}. \tag{21}$$

The rotations $\boldsymbol{R}_{12}$, ..., $\boldsymbol{R}_{M1}$ are updated by Eq. (15). With these fixed, Eq. (13) is again minimized with respect to f_1, ..., f_M, and the same procedure is iterated.

Since Eq. (16) is a first approximation in $\boldsymbol{\omega}_{k(k+1)}$, the rotations $\boldsymbol{R}_{k(k+1)}$ updated by Eq. (15) may not strictly satisfy Eq. (12). However, the discrepancy is very small, so we randomly choose one rotation matrix and replace it by the value that strictly satisfies Eq. (15), i.e., replace it by the inverse of the product of the remaining rotation matrices.

Figure 5 is the result corresponding to Fig. 4. No discrepancy occurs this time. This can be confirmed by many examples, which are not shown here due to page limitation, though.

5 Concluding Remarks

We have shown a consistent technique for generating a 360° circular panorama from images taken by freely moving a hand-held camera.

The first issue is the lack of camera orientation information, which was resolved by computing the homographies between neighboring images and invoking the framework of "oriented projective geometry". The second issue is the discrepancies between the final image and the initial image due to accumulated errors. We resolved this by simultaneously optimizing all the homographies. To this end, we introduced the Gauss-Newton iterations using the Lie algebra representation and the alternating optimization scheme.

In practice, we need to add further corrections and refinements for eliminating intensity discontinuities and geometric discrepancies at individual image boundaries. This can be done easily using existing techniques (we omit the details).

References

1. S.E. Chen, QuickTime VR—An image-based approach to virtual environment navigation, *Proc. SIGGRAPH'95*, August 1995, Los Angeles, U.S.A., pp. 29–38.
2. M. Irani, S. Hsu, and P. Anandan, Video compression using mosaicing representations, *Signal Process: Image Comm.*, **7**-4/5/6 (1995-11), 529–552.
3. M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, Efficient representations of video sequences and their applications, *Signal Process: Image Comm.*, **8**-4/5/6 (1996-11), 327–351.
4. K. Kanatani, *Group-Theoretical Methods in Image Understanding*, Springer, Berlin, Germany, 1990.
5. K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, U.K., 1993.
6. K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam, the Netherlands, 1996; reprinted, Dover, New York, NY, U.S.A., 2005.
7. K. Kanatani, N. Ohta and Y. Kanazawa, Optimal homography computation with a reliability measure, *IEICE Trans. Inf. & Syst.*, **E83-D**-7 (2000-7), 1369–1374.
8. Y. Kanazawa and K. Kanatani, Robust image matching preserving global consistency, *Proc. 6th Asian Conf. Comput. Vision*, January 2004, Jeju, Korea, Vol. 2, pp.1128–1133.
9. Y. Kanazawa and K. Kanatani, Image mosaicing by stratified matching, *Image Vision Computing*, **22**-2 (2004-2), 93–103.
10. M.-C. Lee, C.-l. B. Lin, C. Gu, T. Markoc, S I. Zabinski, and R. Szeliski, A layered video object coding system using sprite and affine motion model, *IEEE Trans. Circuits Video Tech.*, **7**-1 (1997-2), 130–145.
11. S. Leveau and O. Faugeras, Oriented projective geometry for computer vision, *Proc. 4th Euro. Conf. Comput. Vis.*, Vol. 1, April 1996, Cambridge, U.K., pp. 147–156.
12. H.S. Sawhney and S. Ayer, Compact representations of videos through dominant and multiple motion estimation, *IEEE Trans. Patt. Anal. Machine Intell.*, **18**-8 (1996-4), 814–830.
13. H.-Y. Shum and R. Szeliski, Construction of panoramic image mosaics with global and local alignment, *Int. J. Comput. Vis.*, **36**-2 (2000-2), 101–130.
14. J. Stolfi, *Oriented Projective Geometry: A Framework for Geometric Computation*, Academic Press, San Diego, CA, U.S.A., 1991.
15. R. Szeliski and H.-U. Shum, Creating full view panoramic image mosaics and environment maps, *Proc. SIGGRAPH'97*, August 1997, Los Angeles, CA, U.S.A., pp.251–258.

16. Y. Yagi, Omnidirectional sensing and its applications, *IEICE Trans. Inf. & Syst.*, **E82-D**-3 (1999-3), 568–579.
17. Y. Yagi, K. Imai, K. Tsuji and M. Yachida, Iconic memory-based omnidirectional route panorama navigation, *IEEE Trans. Patt. Anal. Mach. Intell.*, **27**-1 (2005-1), 78–87.
18. Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry, *Artif. Intell.*, **78**-1/2 (1995-10), 87–119.