Construction and refinement of panoramic mosaics with global and local alignment

Heung-Yeung Shum and Richard Szeliski
Microsoft Research

Abstract

This paper presents techniques for constructing full view panoramic mosaics from sequences of images. Our representation associates a rotation matrix (and optionally a focal length) with each input image, rather than explicitly projecting all of the images onto a common surface (e.g., a cylinder). In order to reduce accumulated registration errors, we apply global alignment (block adjustment) to the whole sequence of images, which results in an optimal image mosaic (in the least-squares sense). To compensate for small amounts of motion parallax introduced by translations of the camera and other unmodeled distortions, we develop a local alignment (deghosting) technique which warps each image based on the results of pairwise local image registrations. By combining both global and local alignment, we significantly improve the quality of our image mosaics, thereby enabling the creation of full view panoramic mosaics with hand-held cameras.

1 Introduction

The automatic construction of large, high-resolution image mosaics is an active area of research in the fields of photogrammetry, computer vision, image processing, and computer graphics. Image mosaics can be used for many different applications [4]. In computer vision, image mosaics are part of a larger recent trend, namely the study of visual scene representations. The complete description of visual scenes and scene models often entails the recovery of depth or parallax information as well [5, 8]. In graphics, image mosaics play an important role in the field of image-based rendering, which aims to rapidly render photorealistic novel views from collections of real (or pre-rendered) images [2].

For applications such as virtual travel and architectural walkthroughs, it is desirable to have complete (full view) panoramas, i.e., mosaics which cover the whole viewing sphere and hence allow the user to look in any direction. Unfortunately, most of the results to date have been limited to cylindrical panoramas obtained with cameras rotating on leveled tripods adjusted to minimize motion parallax [2, 7, 3]. This has limited the users of mosaic building to researchers and professional photographers who can afford such specialized equipment.

The goal of our work is to remove the need for pure panning motion with no motion parallax. Ideally, we would like any user to be able to "paint" a full view panoramic mosaic with a hand-held camera. To support this vision, several problems must be overcome.

First, we need to avoid using cylindrical or spherical coordinates for constructing the mosaic, since these representations introduce singularities near the poles of the viewing sphere. We solve this problem by associating a rotation matrix (and optionally focal length) with each input image, and performing registration in the input image's coordinate system (we call such mosaics *rotational mosaics*) [9].

Second, we need to deal with accumulated misregistration errors, which are always present in any large image mosaic. For example, if we register a sequence of images using pairwise alignments, there is usually a gap between the last image and the first one even if these two images are the same. In this paper, we develop a global optimization technique, derived from *simultaneous bundle block adjustment* in photogrammetry [10], to find the optimal overall registration.

Third, any deviations from the pure parallax-free motion model or ideal pinhole (projective) camera model may result in local misregistrations, which are visible as a loss of detail or multiple images (*ghosting*). To overcome this problem, we compute local motion estimates (block-based optical flow) between pairs of overlapping images, and use these estimates to warp each input image so as to reduce the misregistration. Note that this is less ambitious than actually recovering a projective depth value for each pixel [5, 8], but has the advantage of being able to simultaneously model other effects such as radial lens distortions and small movements in the image.

The overall flow of processing in our system is thus the following. First, a complete initial panoramic mosaic is assembled sequentially (adding one image at a time and adjusting its position) using our rotational motion model (Section 2). Then, global alignment (*block adjustment*) is invoked to modify each image's transformation (and focal

length) such that the global error across all possible overlapping image pairs is minimized (Section 3). This stage also removes any large inconsistencies in the mosaic, e.g., the "gaps" that might be present in a panoramic mosaic assembled using the sequential algorithm. Lastly, the local alignment (*deghosting*) algorithm is invoked to reduce any local misregistration errors (Section 4).

2 Alignment framework

In our work, we represent image mosaics as collections of images with associated geometrical transformations. We use a hierarchical motion estimation framework [1], which consists of four parts: (i) pyramid construction, (ii) motion estimation, (iii) image warping, and (iv) coarse-to-fine refinement.

2.1 8-parameter homographies

Given two images taken from the same viewpoint (optical center) but in potentially different directions (and/or with different intrinsic parameters), the relationship between two overlapping images can be described by a planar perspective motion model

$$\mathbf{x}' \sim \mathbf{M}\mathbf{x} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & m_8 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (1)$$

where $\mathbf{x} = (x, y, 1)$ and $\mathbf{x}' = (x', y', 1)$ are homogeneous or projective coordinates, and $\sim$ indicates equality up to scale.[1] To recover the parameters, we iteratively update the transformation matrix using

$$\mathbf{M} \leftarrow (\mathbf{I} + \mathbf{D})\mathbf{M}. \quad (2)$$

Resampling image I_1 with the new transformation $\mathbf{x}' \sim (\mathbf{I} + \mathbf{D})\mathbf{M}\mathbf{x}$ is the same as warping the resampled image $\tilde{I}_1$ by $\mathbf{x}'' \sim (\mathbf{I} + \mathbf{D})\mathbf{x}$. We wish to minimize the squared error metric

$$E(\mathbf{d}) = \sum_i [\tilde{I}_1(\mathbf{x}_i'') - I_0(\mathbf{x}_i)]^2 \approx \sum_i [\mathbf{g}_i^T \mathbf{J}_i^T \mathbf{d} + e_i]^2 \quad (3)$$

where $e_i = \tilde{I}_1(\mathbf{x}_i) - I_0(\mathbf{x}_i)$ is the intensity or color error, $\mathbf{g}_i^T = \nabla \tilde{I}_1(\mathbf{x}_i)$ is the image gradient of $\tilde{I}_1$ at $\mathbf{x}_i$, $\mathbf{d} = (d_0, \ldots, d_8)$ is the incremental motion parameter vector, and $\mathbf{J}_i = \mathbf{J}_\mathbf{d}(\mathbf{x}_i) = \frac{\partial \mathbf{x}''}{\partial \mathbf{d}}$, where

$$\mathbf{J}_\mathbf{d}(\mathbf{x}) = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x^2 & -xy & -x \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2 & -y \end{bmatrix}^T \quad (4)$$

is the Jacobian of the resampled point coordinate $\mathbf{x}_i''$ with respect to $\mathbf{d}$.[2]

[1] Since the $\mathbf{M}$ matrix is invariant to scaling, there are only 8 independent parameters.

[2] The entries in the Jacobian correspond to the optical flow induced by the instantaneous motion of a plane in 3D [1].

This least-squares problem (3) has a simple solution through the *normal equations*

$$\mathbf{A}\mathbf{d} = -\mathbf{b}, \quad (5)$$

where

$$\mathbf{A} = \sum_i \mathbf{J}_i \mathbf{g}_i \mathbf{g}_i^T \mathbf{J}_i^T, \qquad \mathbf{b} = \sum_i e_i \mathbf{J}_i \mathbf{g}_i \quad (6)$$

are the *Hessian*, and the *accumulated gradient* or *residual*. These equations can be solved using a symmetric positive definite (SPD) solver. In practice, we set $d_8 = 0$ so that $\mathbf{A}$ is nonsingular, and therefore only solve an 8×8 system.

The 8-parameter algorithm works well provided that initial estimates of the correct transformation are close enough. However, since the motion model contains more free parameters than necessary, it suffers from slow convergence and sometimes gets stuck in local minima. For this reason, we prefer to use the 3-parameter rotational model described next.

2.2 3D rotations and zooms

For a camera centered at the origin, the relationship between a 3D point $\mathbf{p} = (X, Y, Z)$ and its image coordinates $\mathbf{x} = (x, y, 1)$ can be described by

$$\mathbf{x} \sim \mathbf{T}\mathbf{V}\mathbf{R}\mathbf{p}, \quad (7)$$

where

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{V} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and $\mathbf{R}$ are the image plane translation, focal length scaling, and 3D rotation matrices. For simplicity of notation, we assume that pixels are numbered so that the origin is at the image center, i.e., $c_x = c_y = 0$, allowing us to dispense with $\mathbf{T}$. The 3D direction corresponding to a screen pixel $\mathbf{x}$ is given by $\mathbf{p} \sim \mathbf{R}^{-1}\mathbf{V}^{-1}\mathbf{x}$.

For a camera rotating around its center of projection, the mapping (perspective projection) between two images k and l is therefore given by

$$\mathbf{M} \sim \mathbf{V}_k \mathbf{R}_k \mathbf{R}_l^{-1} \mathbf{V}_l^{-1} \quad (8)$$

where each image is represented by $\mathbf{V}_k\mathbf{R}_k$, i.e., a focal length and a 3D rotation.

Assume for now that the focal length is known and is the same for all images, i.e, $\mathbf{V}_k = \mathbf{V}$. To recover the rotation, we perform an incremental update to $\mathbf{R}_k$ based on the angular velocity $\mathbf{\Omega} = (\omega_x, \omega_y, \omega_z)$,

$$\mathbf{R}_k \leftarrow \hat{\mathbf{R}}(\mathbf{\Omega})\mathbf{R}_k \quad \text{or} \quad \mathbf{M} \leftarrow \mathbf{V}\hat{\mathbf{R}}(\mathbf{\Omega})\mathbf{R}_k\mathbf{R}_l^{-1}\mathbf{V}^{-1}. \quad (9)$$

Keeping only terms linear in $\mathbf{\Omega}$, we get

$$\mathbf{M}' \approx \mathbf{V}[\mathbf{I} + \mathbf{X}(\mathbf{\Omega})]\mathbf{R}_k\mathbf{R}_l^{-1}\mathbf{V}^{-1} = (\mathbf{I} + \mathbf{D}_\Omega)\mathbf{M}, \quad (10)$$

where

$$\mathbf{D}_\Omega = \mathbf{V}\mathbf{X}(\mathbf{\Omega})\mathbf{V}^{-1} = \begin{bmatrix} 0 & -\omega_z & f\omega_y \\ \omega_z & 0 & -f\omega_x \\ -\omega_y/f & \omega_x/f & 0 \end{bmatrix}$$

is the deformation matrix which plays the same role as $\mathbf{D}$ in (2), and $\mathbf{X}$ is the cross product operator.

Computing the Jacobian of the entries in $\mathbf{D}_\Omega$ with respect to $\mathbf{\Omega}$ and applying the chain rule, we obtain the new Jacobian,[3]

$$\mathbf{J}_\Omega = \frac{\partial \mathbf{x}''}{\partial \mathbf{d}} \frac{\partial \mathbf{d}}{\partial \mathbf{\Omega}} = \begin{bmatrix} -xy/f & f + x^2/f & -y \\ -f - y^2/f & xy/f & x \end{bmatrix}^T. \quad (11)$$

This Jacobian is then plugged into the previous minimization pipeline to estimate the incremental rotation vector $(\omega_x\ \omega_y\ \omega_z)$, after which $\mathbf{R}_k$ can be updated using (9).

The same general strategy can be followed to obtain the gradient and Hessian associated with any other motion parameters. Details can be found in [6].

The normal equations given in the previous section, together with an appropriately chosen Jacobian matrix, can be used to directly improve the current motion estimate by first computing local intensity errors and gradients, and then accumulating the entries in the parameter gradient vector and Hessian matrix. This straightforward algorithm suffers from several drawbacks: it is susceptible to local minima and outliers, and is also unnecessarily inefficient. We have developed a patch-based alignment algorithm which is much more robust and efficient. The implementation details of our algorithm are given in [6].

3 Global alignment (block adjustment)

The sequential mosaic construction technique described in the previous two sections does a good job of aligning each new image with the previously composited mosaic. Unfortunately, for long image sequences, this approach suffers from the problem of accumulated misregistration errors. In this section, we present a new global alignment method that reduces accumulated error by simultaneously minimizing the misregistration between all overlapping pairs of images. Our method is similar to the *"simultaneous bundle block adjustment"* [10] technique used in photogrammetry but has the following distinct characteristics:

- Corresponding points between pairs of images are automatically obtained using patch-based alignment.
- Our objective function minimizes the difference between ray directions going through corresponding points, and uses a rotational panoramic representation.
- The minimization is formulated as a constrained least-squares problem with hard linear constraints for identical focal lengths and repeated frames.

[3] This is the same as the rotational component of instantaneous rigid flow [1].

3.1 Problem formulation

Our global alignment algorithm is a *feature-based* technique. We divide each image into a number of patches (e.g., 16×16 pixels), and use the patch centers as prospective "feature" points. For a patch j in image k, let $l \in \mathcal{N}_{jk}$ be the set of overlapping images in which patch j is totally contained (under the current set of transformations). Let $\mathbf{x}_{jk}$ be the center of this patch. To compute the patch alignment, we use image k as I_0 and image l as I_1 and invoke the local search algorithm [6], which returns an estimated displacement $\mathbf{u}_{jl} = \mathbf{u}_j^*$. The corresponding point in the warped image $\tilde{I}_1$ is thus $\tilde{\mathbf{x}}_{jl} = \mathbf{x}_{jk} + \mathbf{u}_{jl}$. In image l, this point's coordinate is $\mathbf{x}_{jl} \sim \mathbf{M}_l\mathbf{M}_k^{-1}\tilde{\mathbf{x}}_{jl}$, or $\mathbf{x}_{jl} \sim \mathbf{V}_l\mathbf{R}_l\mathbf{R}_k^{-1}\mathbf{V}_k^{-1}\tilde{\mathbf{x}}_{jl}$ if the rotational panoramic representation is used.

Given these point correspondences, one way to formulate the global alignment is to minimize the difference between screen coordinates of all overlapping pairs of images [6]. Unfortunatley, the gradients with respect to the motion parameters are complicated. A simpler formulation is to minimize the difference between the ray directions of corresponding points using a rotational panoramic representation with unknown focal length. Geometrically, this is equivalent to adjusting the rotation and focal length for each frame so that the bundle of corresponding rays converge.

Let the ray direction in the final composited image mosaic be a unit vector $\mathbf{p}_j$, and its corresponding ray direction in the kth frame as $\mathbf{p}_{jk} \sim \mathbf{R}_k^{-1}\mathbf{V}_k^{-1}\mathbf{x}_{jk}$. We can formulate block adjustment to simultaneously optimize over both the pose (rotation and focal length $\{\mathbf{R}_k, f_k\}$) and structure (ray direction $\{\mathbf{p}_j\}$) parameters,

$$E(\{\mathbf{R}_k, f_k\}, \{\mathbf{p}_j\}) = \sum_{j,k} \|\mathbf{p}_{jk} - \mathbf{p}_j\|^2 = \sum_{j,k} \|\mathbf{R}_k^{-1}\hat{\mathbf{x}}_{jk} - \mathbf{p}_j\|^2 \quad (12)$$

where

$$\hat{\mathbf{x}}_{jk} = \begin{bmatrix} x_{jk} \\ y_{jk} \\ f_k \end{bmatrix} / l_{jk}, \quad l_{jk} = \sqrt{{x_{jk}}^2 + {y_{jk}}^2 + {f_k}^2} \quad (13)$$

is the ray direction going through the jth feature point located at (x_{jk}, y_{jk}) in the kth frame. Note that this absorbs the f_k parameter in V_k into the coordinate definition.

The advantage of the above direct minimization (12) is that both pose and structure can be solved independently

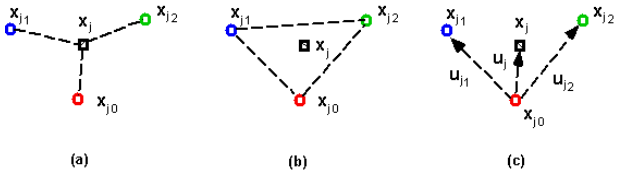

Figure 1: (a) minimizing the difference between $\mathbf{x}_j$ and all $\mathbf{x}_{jk}$; (b) minimizing the difference between all pairs of $\mathbf{x}_{jk}$ and $\mathbf{x}_{jl}$; (c) desired flow for deghosting $\bar{\mathbf{u}}_{jk}$ is a down-weighted average of all pairwise $\mathbf{u}_{jl}$.

for each frame. For instance, we can solve $\mathbf{p}_j$ using linear least-squares, $\mathbf{R}_k$ using relative orientation, and f_k using nonlinear least-squares. The disadvantage of this method is its slow convergence due to the highly coupled nature of the equations and unknowns.

For the purpose of global alignment, however, it is not necessary to explicitly recover the ray directions. We can reformulate block adjustment to only minimize over pose ($\{\mathbf{R}_k, f_k\}$) for all frames k, without computing the $\{\mathbf{p}_j\}$. More specifically, we estimate the pose by minimizing the difference in ray directions between all pairs (k and l) of overlapping images. $E(\{\mathbf{R}_k, f_k\})$ is given by

$$\sum_{j,k,l\in\mathcal{N}_{jk}} \|\mathbf{p}_{jk}-\mathbf{p}_{jl}\|^2 = \sum_{j,k,l\in\mathcal{N}_{jk}} \|\mathbf{R}_k^{-1}\hat{\mathbf{x}}_{jk}-\mathbf{R}_l^{-1}\hat{\mathbf{x}}_{jl}\|^2 \quad (14)$$

Once the pose has been computed, we can compute the estimated directions $\mathbf{p}_j$ using the known correspondence from all overlapping frames $\mathcal{N}_{jk}$ where the feature point j is visible,

$$\mathbf{p}_j \sim \frac{1}{n_{jk}+1} \sum_{l\in\mathcal{N}_{jk}\cup k} \mathbf{R}_l^{-1}\mathbf{V}_l^{-1}\mathbf{x}_{jl}. \quad (15)$$

where $n_{jk} = |\mathcal{N}_{jk}|$ is the number of overlapping images where patch j is completely visible (this information will be used later in the deghosting stage).

Figure 1 shows the difference between the above two formulations.

3.2 Solution technique

The least-squares problem (14) can be solved using our regular gradient descent method. To recover the pose $\{\mathbf{R}_k, f_k\}$, we iteratively update the rotation matrix and focal length

$$\mathbf{R}_k^{-1} \leftarrow \hat{\mathbf{R}}(\boldsymbol{\Omega}_k)\mathbf{R}_k^{-1} \quad \text{and} \quad f_k \leftarrow f_k + \delta f_k. \quad (16)$$

The minimization problem (14) can be rewritten as

$$E(\{\boldsymbol{\Omega}_k, \delta f_k\}) \approx \sum_{j,k,l\in\mathcal{N}_{jk}} \|\mathbf{H}_{jk}\mathbf{y}_k - \mathbf{H}_{jl}\mathbf{y}_l + \mathbf{e}_j\|^2 \quad (17)$$

Figure 2: Reducing accumulated errors of mosaics: (a) with gaps/overlap; (b) after block adjustment.

where $\mathbf{e}_j = \mathbf{p}_{jk} - \mathbf{p}_{jl}$, $\mathbf{y}_k = \begin{bmatrix} \boldsymbol{\Omega}_k \\ \delta f_k \end{bmatrix}$, $\mathbf{H}_{jk} = \begin{bmatrix} \frac{\partial \mathbf{p}_{jk}}{\partial \boldsymbol{\Omega}_k} \\ \frac{\partial \mathbf{p}_{jk}}{\partial f_k} \end{bmatrix}$, and

$$\frac{\partial \mathbf{p}_{jk}}{\partial \boldsymbol{\Omega}_k} = \frac{\partial(\mathbf{I}+\mathbf{X}(\boldsymbol{\Omega}))\mathbf{p}_{jk}}{\partial \boldsymbol{\Omega}_k} = \mathbf{X}(\mathbf{p}_{jk}), \quad (18)$$

$$\frac{\partial \mathbf{p}_{jk}}{\partial f_k} = \mathbf{R}_k^{-1}\frac{\partial \tilde{\mathbf{x}}_{jk}}{\partial f_j} = \mathbf{R}_k^{-1}\begin{bmatrix} -x_{jk}f_k \\ -y_{jk}f_k \\ {l_{jk}}^2 - {f_k}^2 \end{bmatrix} / {l_{jk}}^3. \quad (19)$$

The normal equations can be formed directly from (17), updating four different subblocks of $\mathbf{A}$ and two different subvectors of $\mathbf{b}$ for each patch correspondence. Because $\mathbf{A}$ is symmetric, the normal equations can be stably solved using a symmetric positive definite (SPD) linear system solver. By incorporating additional constraints on the pose, we can formulate our minimization problem (14) as a constrained least-squares problem which can be solved using Lagrange multipliers. Possible linear constraints include:

- $\Omega_0 = 0$. First frame pose is unchanged.
- $\delta f_k = 0$ for $k = 0, ..., N-1$. Focal lengths known.
- $\delta f_k = \delta f_0$ for $k = 1, ..., N$. Same but unknown.
- $\delta f_k = \delta f_l$, $\Omega_k = \Omega_l$, Frame k is the same as frame l. To apply this constraint, we set $f_k = f_l$ and $\mathbf{R}_k = \mathbf{R}_l$.

The above minimization process converges quickly (several iterations) in practice. The running time for the iterative non-linear least-squares solver is much less than the time required to build the point correspondences.

Figure 2 shows how misregistration errors quickly accumulate in sequential registration. Figure 2a shows a big gap at the end of registering a sequence of 24 images (image size 300×384) where an initial estimate of focal length 256 is used. The double image of the right painting on the wall signals a big misalignment. This double image is removed, as shown in Figure 2b, by applying our global alignment method which simultaneously adjusts all frame rotations and computes a new estimated focal length of 251.8.

Figure 3: Deghosting a mosaic with motion parallax: (a) with parallax; (b) after single deghosting step (patch size 32); (c) multiple steps (sizes 32, 16 and 8).

Figure 4: Deghosting a mosaic with optical distortion: (a) with distortion; (b) after multiple steps.

4 Local alignment (deghosting)

After the global alignment has been run, there may still be localized mis-registrations present in the image mosaic, due to deviations from the idealized parallax-free camera model. Such deviations might include camera translation (especially for hand-held cameras), radial distortion, the mis-location of the optical center (which can be significant for scanned photographs or Photo CDs), and moving objects.

To compensate for these effects, we would like to estimate the amount of mis-registration and to then locally warp each image so that the overall mosaic does not contain visible *ghosting* (double images) or blurred details. If our mosaic contains just a few images, we could choose one image as the *base*, and then compute the optical flow between it and all other images, which could then be deformed to match the base. Another possibility would be to explicitly estimate the camera motion and residual parallax, but this would not compensate for other distortions.

However, since we are dealing with large image mosaics, we need an approach which makes all of the images globally consistent, without a preferred base. One approach might be to warp each image so that it best matches the current mosaic. For small amounts of misregistration, where most of the visual effects are simple blurring (loss of detail), this should work fairly well. However, for larger misregistrations, where ghosting is present, the local motion estimation would likely fail. An alternative approach, which is the one we propose, is to compute the flow between all pairs of images, and to then infer the desired local warps from these computations.

Recall that the block adjustment algorithm (15) provides an estimate $\mathbf{p}_j$ of the true direction in space corresponding to the jth patch center in the kth image, $\mathbf{x}_{jk}$. The projection of this direction onto the kth image is

$$\begin{aligned}\overline{\mathbf{x}}_{jk} &\sim \mathbf{V}_k\mathbf{R}_k\frac{1}{n_{jk}+1}\sum_{l\in\mathcal{N}_{jk}\cup k}\mathbf{R}_l^{-1}\mathbf{V}_l^{-1}\mathbf{x}_{jl} \\ &= \frac{1}{n_{jk}+1}\left(\mathbf{x}_{jk}+\sum_{l\in\mathcal{N}_{jk}}\tilde{\mathbf{x}}_{jl}\right).\end{aligned}$$

This can be converted into a motion estimate

$$\overline{\mathbf{u}}_{jk} = \overline{\mathbf{x}}_{jk} - \mathbf{x}_{jk} = \frac{1}{n_{jk}+1}\sum_{l\in\mathcal{N}_{jk}}\mathbf{u}_{jl}. \qquad (20)$$

This formula has a very nice, intuitively satisfying explanation (Figure 1c). The local motion required to bring patch center j in image k into global registration is simply the average of the pairwise motion estimates with all overlapping images, *downweighted* by the fraction $n_{jk}/(n_{jk}+1)$. This factor prevents local motion estimates from "overshooting" in their corrections (consider, for example, just two images, where each image warps itself to match its neighbor). Thus, we can compute the location motion estimate for each image by simply examining its misregistration with its neighbors, without having to worry about what warps these other neighbors might be undergoing themselves.

Once the local motion estimates have been computed, we use an *inverse mapping* algorithm to warp each image so as to reduce ghosting. For each pixel in the *new* (warped) image I'_k, we need to know the relative distance (flow) to the appropriate source pixel. We compute this field using a sparse data interpolation technique. The input to this algorithm is the set of negative flows $-\overline{\mathbf{u}}_{jk}$ located at pixel coordinates $\overline{\mathbf{x}}_{jk} = \mathbf{x}_{jk} + \overline{\mathbf{u}}_{jk}$. At present, we simply place a tent (bilinear) function over each flow sample (the size is currently twice the patch size). To make this interpolator locally *reproducible* (no "dips" in the interpolated surface), we divide each accumulated flow value by the accumulated weight (plus a small amount, say 0.1, to round the transitions into regions with no motion estimates).

The next two examples illustrate the use of local alignment for sequences where the global motion model is clearly violated. The first example consists of two images taken with a hand-held digital camera (Kodak DC40) where some camera translation is present. The parallax introduced by this camera translation can be observed in the registered image (Figure 3a) where the front object (a stop sign) shows up as a double image. This misalignment is significantly reduced using our local alignment method (Figure 3b). However, some visual artifacts still exist because our local alignment is patch-based (e.g. patch size 32 is used in Figure 3b). To overcome this problem, we re-apply local alignment with smaller patch sizes. Figure 3c shows the result

Figure 5: Four views of an image mosaic of lobby constructed from 54 images.

after applying local alignment three times with patch sizes of 32, 16 and 8.

The global motion model is also invalid when registering two images with strong optical distortion. One way to deal with radial distortion is to carefully calibrate the camera. An alternative is to use local alignment. Figure 4a shows a mosaic of two images taken with a Pulnix camera and a Fujinon F2.8 wide angle lens. The picture shows significant misalignment due to radial distortion (notice how straight lines, e.g. the door, are curved). Figure 4b shows an improved mosaic using repeated local alignment with patch sizes 32, 16, 8.

Our final example shows a full view panoramic mosaic. Three panoramic image sequences of a building lobby were taken with the camera on a tripod tilted at three different angles. The camera motion covers more than two thirds of the viewing sphere, including the top. After registering all of the images sequentially with patch-based alignment, we apply our global and local alignment techniques to obtain the final image mosaic, shown in Figure 5. These four views of the final image mosaic are equivalent to images taken with a very large rectilinear lens. Additional examples including large panoramic mosaics can be found in [6].

5 Summary

In this paper, we have developed some novel techniques for constructing full view panoramic image mosaics from image sequences. Instead of projecting all of the images onto a common surface (e.g., a cylinder or a sphere), we associate a rotation matrix and a (usually unknown) focal length with each input image. Based on this rotational panoramic representation, we have developed block adjustment (global alignment) and deghosting (local alignment) techniques to significantly improve the quality of image mosaics, thereby enabling the construction of mosaics from images taken with hand-held cameras. We believe that this work will make panoramic photography and the construction of virtual environments much more interesting to a wide range of users, and stimulate further research and development in image-based rendering and the representation of visual scenes.

References

[1] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Second European Conference on Computer Vision (ECCV'92)*, pages 237–252, May 1992.

[2] S. E. Chen. QuickTime VR – an image-based approach to virtual environment navigation. *Computer Graphics (SIGGRAPH'95)*, pages 29–38, August 1995.

[3] S. B. Kang and R Weiss. Characterization of errors in compositing panoramic images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 103–109, June 1997.

[4] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna. Representation of scenes from collections of images. In *IEEE Workshop on Representations of Visual Scenes*, pages 10–17, June 1995.

[5] H. S. Sawhney. Simplifying motion and structure analysis using planar parallax and image warping. In *Twelfth International Conference on Pattern Recognition (ICPR'94)*, volume A, pages 403–408, October 1994.

[6] H.-Y. Shum and R. Szeliski. Panoramic image mosaicing. Technical Report MSR-TR-97-??, Microsoft Research, September 1997.

[7] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, pages 22–30, March 1996.

[8] R. Szeliski and S. B. Kang. Direct methods for visual scene reconstruction. In *IEEE Workshop on Representations of Visual Scenes*, pages 26–33, June 1995.

[9] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and texture-mapped models. In *Computer Graphics (SIGGRAPH'97) Proceedings*, pages 251–258, August 1997.

[10] P. R. Wolf. *Elements of photogrammetry*. McGraw-Hill, New York, 1974.