Automatic Panoramic Image Stitching using Invariant Features

Matthew Brown and David G. Lowe
{mbrown|lowe}@cs.ubc.ca
Department of Computer Science,
University of British Columbia,
Vancouver, Canada.

Abstract

This paper concerns the problem of fully automated panoramic image stitching. Though the 1D problem (single axis of rotation) is well studied, 2D or multi-row stitching is more difficult. Previous approaches have used human input or restrictions on the image sequence in order to establish matching images. In this work, we formulate stitching as a multi-image matching problem, and use invariant local features to find matches between all of the images. Because of this our method is insensitive to the ordering, orientation, scale and illumination of the input images. It is also insensitive to noise images that are not part of a panorama, and can recognise multiple panoramas in an unordered image dataset. In addition to providing more detail, this paper extends our previous work in the area [BL03] by introducing gain compensation and automatic straightening steps.

1 Introduction

Panoramic image stitching has an extensive research literature [Sze04, Mil75, BL03] and several commercial applications [Che95, REA, MSF]. The basic geometry of the problem is well understood, and consists of estimating a 3×3 camera matrix or homography for each image [HZ04, SS97]. This estimation process needs an initialisation, which is typically provided by user input to approximately align the images, or a fixed image ordering. For example, the PhotoStitch software bundled with Canon digital cameras requires a horizontal or vertical sweep, or a square matrix of images. REALVIZ Stitcher version 4 [REA] has a user interface to roughly position the images with a mouse, before automatic registration proceeds. Our work is novel in that we require no such initialisation to be provided.

In the research literature methods for automatic image alignment and stitching fall broadly into two categories – direct [SK95, IA99, SK99, SS00] and feature based [ZFD97, CZ98, MJ02]. Direct methods have the advantage that they use all of the available image data and hence can provide very accurate registration, but they require a close initialisation. Feature based registration does not require initialisation, but traditional feature matching methods (e.g., correlation of image patches around Harris corners [Har92, ST94]) lack the invariance properties needed to enable reliable matching of arbitrary panoramic image sequences.

In this paper we describe an invariant feature based approach to fully automatic panoramic image stitching. This has several advantages over previous approaches. Firstly, our use of invariant features enables reliable matching of panoramic image sequences despite rotation, zoom and illumination change in the input images. Secondly, by viewing image stitching as a multi-image matching problem, we can automatically discover the matching relationships between the images, and recognise panoramas in unordered datasets. Thirdly, we generate high-quality results using multi-band blending to render seamless output panoramas. This paper extends our earlier work in the area [BL03] by introducing gain compensation and automatic straightening steps. We also describe an efficient bundle adjustment implementation and show how to perform multi-band blending for multiple overlapping images with any number of bands.

The remainder of the paper is structured as follows. Section 2 develops the geometry of the problem and motivates our choice of invariant features. Section 3 describes our image matching methodology (RANSAC) and a probabilistic model for image match verification. In section 4 we describe our image alignment algorithm (bundle adjustment) which jointly optimises the parameters of each camera. Sections 5 - 7 describe the rendering pipeline including automatic straightening, gain compensation and multi-band blending. In section 9 we present conclusions and ideas for future work.

2 Feature Matching

The first step in the panoramic recognition algorithm is to extract and match SIFT [Low04] features between all of

the images. SIFT features are located at scale-space maxima/minima of a difference of Gaussian function. At each feature location, a characteristic scale and orientation is established. This gives a similarity-invariant frame in which to make measurements. Although simply sampling intensity values in this frame would be similarity invariant, the invariant descriptor is actually computed by accumulating local gradients in orientation histograms. This allows edges to shift slightly without altering the descriptor vector, giving some robustness to affine change. This spatial accumulation is also important for shift invariance, since the interest point locations are typically only accurate in the 0-3 pixel range [BSW05, SZ03]. Illumination invariance is achieved by using gradients (which eliminates bias) and normalising the descriptor vector (which eliminates gain).

Since SIFT features are invariant under rotation and scale changes, our system can handle images with varying orientation and zoom (see figure 8). Note that this would not be possible using traditional feature matching techniques such as correlation of image patches around Harris corners. Ordinary (translational) correlation is not invariant under rotation, and Harris corners are not invariant to changes in scale.

Assuming that the camera rotates about its optical centre, the group of transformations the images may undergo is a special group of homographies. We parameterise each camera by a rotation vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]$ and focal length f. This gives pairwise homographies $\tilde{\mathbf{u}}_i = \mathbf{H}_{ij}\tilde{\mathbf{u}}_j$ where

$$\mathbf{H}_{ij} = \mathbf{K}_i \mathbf{R}_i \mathbf{R}_j^T \mathbf{K}_j^{-1} \tag{1}$$

and $\tilde{\mathbf{u}}_i$, $\tilde{\mathbf{u}}_j$ are the homogeneous image positions ($\tilde{\mathbf{u}}_i = s_i[\mathbf{u}_i, 1]$, where $\mathbf{u}_i$ is the 2-dimensional image position). The 4 parameter camera model is defined by

$$\mathbf{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

and (using the exponential representation for rotations)

$$\mathbf{R}_i = e^{[\boldsymbol{\theta}_i]_\times}, \quad [\boldsymbol{\theta}_i]_\times = \begin{bmatrix} 0 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 0 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 0 \end{bmatrix}. \tag{3}$$

Ideally one would use image features that are invariant under this group of transformations. However, for small changes in image position

$$\mathbf{u}_i = \mathbf{u}_{i0} + \left.\frac{\partial \mathbf{u}_i}{\partial \mathbf{u}_j}\right|_{\mathbf{u}_{i0}} \Delta\mathbf{u}_j \tag{4}$$

or equivalently $\tilde{\mathbf{u}}_i = \mathbf{A}_{ij}\tilde{\mathbf{u}}_j$, where

$$\mathbf{A}_{ij} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

is an affine transformation obtained by linearising the homography about $\mathbf{u}_{i0}$. This implies that each small image patch undergoes an affine transformation, and justifies the use of SIFT features which are partially invariant under affine change.

Once features have been extracted from all n images (linear time), they must be matched. Since multiple images may overlap a single ray, each feature is matched to its k nearest neighbours in feature space (we use $k = 4$). This can be done in $O(n \log n)$ time by using a k-d tree to find approximate nearest neighbours [BL97]. A k-d tree is an axis aligned binary space partition, which recursively partitions the feature space at the mean in the dimension with highest variance.

3 Image Matching

At this stage the objective is to find all matching (i.e. overlapping) images. Connected sets of image matches will later become panoramas. Since each image could potentially match every other one, this problem appears at first to be quadratic in the number of images. However, it is only necessary to match each image to a small number of overlapping images in order to get a good solution for the image geometry.

From the feature matching step, we have identified images that have a large number of matches between them. We consider a constant number m images, that have the greatest number of feature matches to the current image, as potential image matches (we use $m = 6$). First, we use RANSAC to select a set of inliers that are compatible with a homography between the images. Next we apply a probabilistic model to verify the match.

3.1 Robust Homography Estimation using RANSAC

RANSAC (random sample consensus) [FB81] is a robust estimation procedure that uses a minimal set of randomly sampled correspondences to estimate image transformation parameters, and finds a solution that has the best consensus with the data. In the case of panoramas we select sets of $r = 4$ feature correspondences and compute the homography $\mathbf{H}$ between them using the direct linear transformation (DLT) method [HZ04]. We repeat this with $n = 500$ trials and select the solution that has the maximum number of inliers (whose projections are consistent with $\mathbf{H}$ within a tolerance ϵ pixels). Given the probability that a feature

match is correct between a pair of matching images (the inlier probability) is p_i, the probability of finding the correct transformation after n trials is

$$p(\mathbf{H}\ is\ correct) = 1 - (1 - (p_i)^r)^n. \quad (6)$$

After a large number of trials the probability of finding the correct homography is very high. For example, for an inlier probability $p_i = 0.5$, the probability that the correct homography is *not* found after 500 trials is approximately 1×10^{-14}.

RANSAC is essentially a sampling approach to estimating $\mathbf{H}$. If instead of maximising the number of inliers one maximises the sum of the log likelihoods, the result is maximum likelihood estimation (MLE). Furthermore, if priors on the transformation parameters are available, one can compute a maximum a posteriori estimate (MAP). These algorithms are known as MLESAC and MAPSAC respectively [Tor02].

3.2 Probabilistic Model for Image Match Verification

For each pair of potentially matching images we have a set of feature matches that are geometrically consistent (RANSAC inliers) and a set of features that are inside the area of overlap but not consistent (RANSAC outliers). The idea of our verification model is to compare the probabilities that this set of inliers/outliers was generated by a correct image match or by a false image match.

For a given image we denote the total number of features in the area of overlap n_f and the number of inliers n_i. The event that this image matches correctly/incorrectly is represented by the binary variable $m \in \{0, 1\}$. The event that the i^{th} feature match $f^{(i)} \in \{0, 1\}$ is an inlier/outlier is assumed to be independent Bernoulli, so that the total number of inliers is Binomial

$$p(f^{(1:n_f)}|m = 1) = B(n_i; n_f, p_1) \quad (7)$$

$$p(f^{(1:n_f)}|m = 0) = B(n_i; n_f, p_0) \quad (8)$$

where p_1 is the probability a feature is an inlier given a correct image match, and p_0 is the probability a feature is an inlier given a false image match. The set of feature match variables $\{f^{(i)}, i = 1, 2, \ldots, n_f\}$ is denoted $f^{(1:n_f)}$. The number of inliers $n_i = \sum_{i=1}^{n_f} f^{(i)}$ and $B(.)$ is the Binomial distribution

$$B(x; n, p) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}. \quad (9)$$

We choose values $p_1 = 0.6$ and $p_0 = 0.1$. We can now evaluate the posterior probability that an image match is correct using Bayes' Rule

$$p(m = 1|f^{(1:n_f)}) = \frac{p(f^{(1:n_f)}|m = 1)p(m = 1)}{p(f^{(1:n_f)})} \quad (10)$$

$$= \frac{1}{1 + \frac{p(f^{(1:n_f)}|m=0)p(m=0)}{p(f^{(1:n_f)}|m=1)p(m=1)}} \quad (11)$$

We accept an image match if $p(m = 1|f^{(1:n_f)}) > p_{min}$

$$\frac{B(n_i; n_f, p_1)p(m = 1)}{B(n_i; n_f, p_0)p(m = 0)} \underset{reject}{\overset{accept}{\gtrless}} \frac{1}{\frac{1}{p_{min}} - 1}. \quad (12)$$

Choosing values $p(m = 1) = 10^{-6}$ and $p_{min} = 0.999$ gives the condition

$$n_i > \alpha + \beta n_f \quad (13)$$

for a correct image match, where $\alpha = 8.0$ and $\beta = 0.3$. Though in practice we have chosen values for $p_0, p_1, p(m = 0)$, $p(m = 1)$ and p_{min}, they could in principle be learnt from the data. For example, p_1 could be estimated by computing the fraction of matches consistent with correct homographies over a large dataset.

Once pairwise matches have been established between images, we can find panoramic sequences as connected sets of matching images. This allows us to recognise multiple panoramas in a set of images, and reject noise images which match to no other images (see figure (2)).

4 Bundle Adjustment

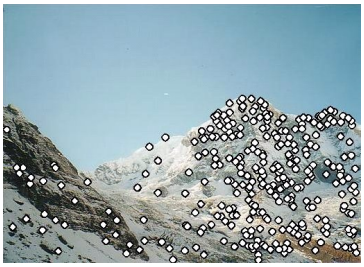
Given a set of geometrically consistent matches between the images, we use bundle adjustment [TMHF99] to solve for all of the camera parameters jointly. This is an essential step as concatenation of pairwise homographies would cause accumulated errors and disregard multiple constraints between images, e.g., that the ends of a panorama should join up. Images are added to the bundle adjuster one by one, with the best matching image (maximum number of consistent matches) being added at each step. The new image is initialised with the same rotation and focal length as the image to which it best matches. Then the parameters are updated using Levenberg-Marquardt.

The objective function we use is a robustified sum squared projection error. That is, each feature is projected into all the images in which it matches, and the sum of squared image distances is minimised with respect to the camera parameters[1]. Given a correspondence $\mathbf{u}_i^k \leftrightarrow \mathbf{u}_j^l$ ($\mathbf{u}_i^k$

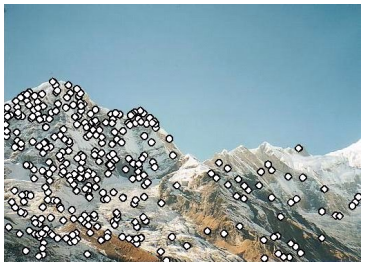
[1]Note that it would also be possible (and in fact statistically optimal) to represent the unknown ray directions $\mathbf{X}$ explicitly, and to estimate them jointly with the camera parameters. This would not increase the complexity of the algorithm if a sparse bundle adjustment method was used [TMHF99].

(a) Image 1

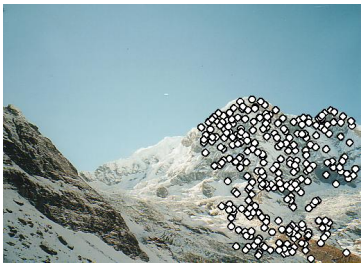
(b) Image 2



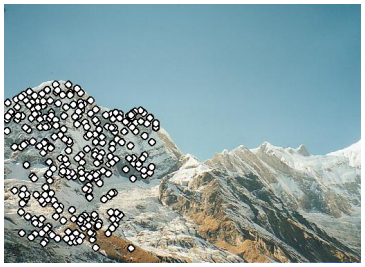
(c) SIFT matches 1



(d) SIFT matches 2



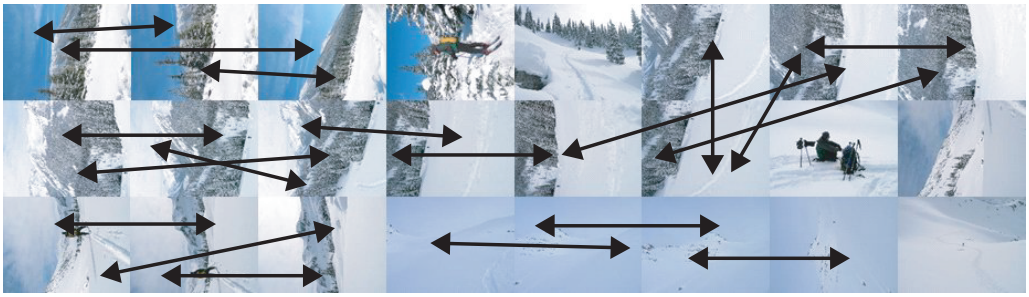
(e) RANSAC inliers 1



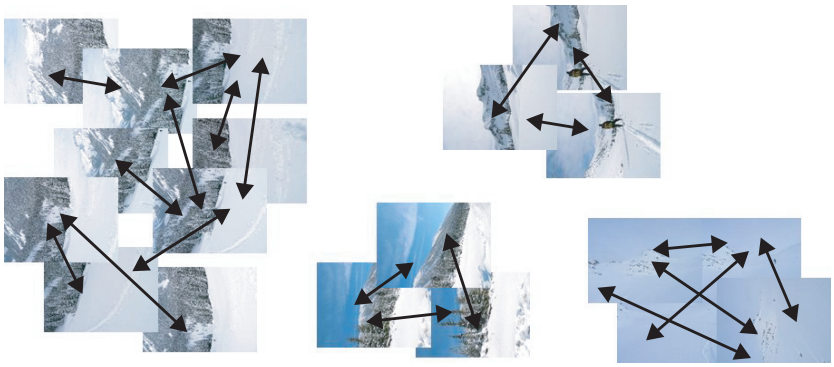
(f) RANSAC inliers 2

(g) Images aligned according to a homography

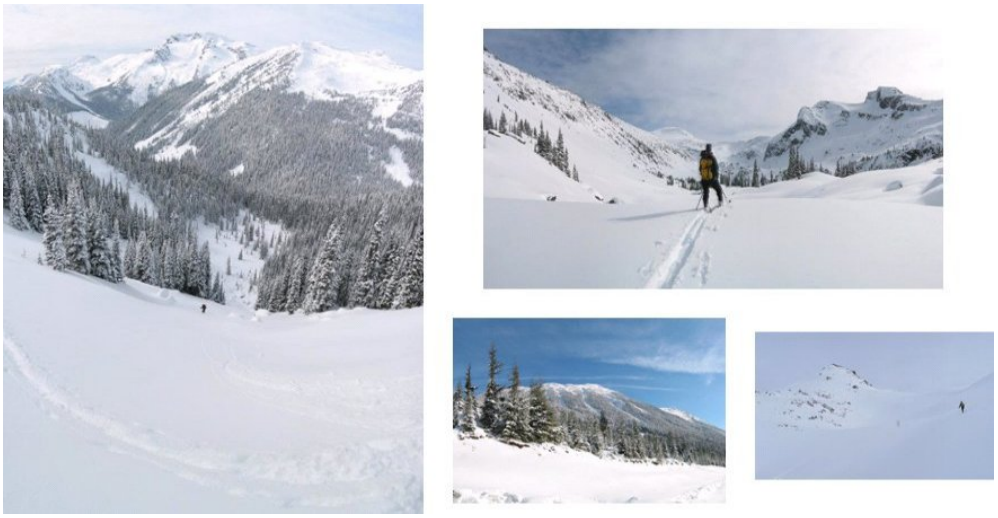
Figure 1. SIFT features are extracted from all of the images. After matching all of the features using a k-d tree, the m images with the greatest number of feature matches to a given image are checked for an image match. First RANSAC is performed to compute the homography, then a probabilistic model is invoked to verify the image match based on the number of inliers. In this example the input images are 517×374 pixels and there are 247 correct feature matches.



(a) Image matches



(b) Connected components of image matches



(c) Output panoramas

Figure 2. Recognising panoramas. Given a noisy set of feature matches, we use RANSAC and a probabilistic verification procedure to find consistent image matches (a). Each arrow between a pair of images indicates that a consistent set of feature matches was found between that pair. Connected components of image matches are detected (b) and stitched into panoramas (c). Note that the algorithm is insensitive to noise images that do not belong to a panorama (connected components of size 1 image).

denotes the position of the kth feature in image i), the residual is

$$\mathbf{r}_{ij}^k = \mathbf{u}_i^k - \mathbf{p}_{ij}^k \tag{14}$$

where $\mathbf{p}_{ij}^k$ is the projection from image j to image i of the point corresponding to $\mathbf{u}_i^k$

$$\tilde{\mathbf{p}}_{ij}^k = \mathbf{K}_i \mathbf{R}_i \mathbf{R}_j^T \mathbf{K}_j^{-1} \tilde{\mathbf{u}}_j^l. \tag{15}$$

The error function is the sum over all images of the robustified residual errors

$$e = \sum_{i=1}^{n} \sum_{j \epsilon \mathcal{I}(i)} \sum_{k \epsilon \mathcal{F}(i,j)} h(\mathbf{r}_{ij}^k) \tag{16}$$

where n is the number of images, $\mathcal{I}(i)$ is the set of images matching to image i, $\mathcal{F}(i,j)$ is the set of feature matches between images i and j. We use a Huber robust error function [Hub81]

$$h(\mathbf{x}) = \begin{cases} |\mathbf{x}|^2, & \text{if } |\mathbf{x}| < \sigma \\ 2\sigma|\mathbf{x}| - \sigma^2, & \text{if } |\mathbf{x}| \geq \sigma \end{cases}. \tag{17}$$

This error function combines the fast convergence properties of an L_2 norm optimisation scheme for inliers (distance less than σ), with the robustness of an L_1 norm scheme for outliers (distance greater than σ). We use an outlier distance $\sigma = \infty$ during initialisation and $\sigma = 2$ pixels for the final solution.

This is a non-linear least squares problem which we solve using the Levenberg-Marquardt algorithm. Each iteration step is of the form

$$\mathbf{\Phi} = (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{C}_p^{-1})^{-1} \mathbf{J}^T \mathbf{r} \tag{18}$$

where $\mathbf{\Phi}$ are all the parameters, $\mathbf{r}$ the residuals and $\mathbf{J} = \partial \mathbf{r} / \partial \mathbf{\Phi}$. We encode our prior belief about the parameter changes in the (diagonal) covariance matrix $\mathbf{C}_p$

$$\mathbf{C}_p = \begin{bmatrix} \sigma_\theta^2 & 0 & 0 & 0 & 0 & \dots \\ 0 & \sigma_\theta^2 & 0 & 0 & 0 & \dots \\ 0 & 0 & \sigma_\theta^2 & 0 & 0 & \dots \\ 0 & 0 & 0 & \sigma_f^2 & 0 & \dots \\ 0 & 0 & 0 & 0 & \sigma_\theta^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \end{bmatrix} \tag{19}$$

This is set such that the standard deviation of angles is $\sigma_\theta = \pi/16$ and focal lengths $\sigma_f = \bar{f}/10$ (where $\bar{f}$ is the mean of the focal lengths estimated so far). This helps in choosing suitable step sizes, and hence speeding up convergence. For example, if a spherical covariance matrix were used, a change of 1 radian in rotation would be equally penalised as a change of 1 pixel in the focal length parameter. Finally, the λ parameter is varied at each iteration to ensure that the objective function of equation 16 does in fact decrease.

The derivatives are computed analytically via the chain rule, for example

$$\frac{\partial \mathbf{p}_{ij}^k}{\partial \theta_{i1}} = \frac{\partial \mathbf{p}_{ij}^k}{\partial \tilde{\mathbf{p}}_{ij}^k} \frac{\partial \tilde{\mathbf{p}}_{ij}^k}{\partial \theta_{i1}} \tag{20}$$

where

$$\frac{\partial \mathbf{p}_{ij}^k}{\partial \tilde{\mathbf{p}}_{ij}^k} = \frac{\partial \begin{bmatrix} x/z & y/z \end{bmatrix}}{\partial \begin{bmatrix} x & y & z \end{bmatrix}} = \begin{bmatrix} 1/z & 0 & -x/z^2 \\ 0 & 1/z & -y/z^2 \end{bmatrix} \tag{21}$$

and

$$\frac{\partial \tilde{\mathbf{p}}_{ij}^k}{\partial \theta_{i1}} = \mathbf{K}_i \frac{\partial \mathbf{R}_i}{\partial \theta_{i1}} \mathbf{R}_j \mathbf{K}_j^{-1} \tilde{\mathbf{u}}_j^l \tag{22}$$

$$\frac{\partial \mathbf{R}_i}{\partial \theta_{i1}} = \frac{\partial}{\partial \theta_{i1}} e^{[\boldsymbol{\theta_i}]_\times} = e^{[\boldsymbol{\theta_i}]_\times} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{23}$$

4.1 Fast Solution by Direct Computation of the Linear System

Since the matrix $\mathbf{J}$ is sparse, forming $\mathbf{J}^T\mathbf{J}$ by explicitly multiplying $\mathbf{J}$ by its transpose is inefficient. In fact, this would be the most expensive step in bundle adjustment, costing $O(MN^2)$ for an $M \times N$ matrix $\mathbf{J}$ (M is twice the number of measurements and N is the number of parameters). The sparseness arises because each image typically only matches to a small subset of the other images. This means that in practice each element of $\mathbf{J}^T\mathbf{J}$ can be computed in much fewer than M multiplications

$$(\mathbf{J}^T\mathbf{J})_{ij} = \sum_{k \epsilon \mathcal{F}(i,j)} \frac{\partial \mathbf{r}_{ij}^k}{\partial \Phi_i}^T \frac{\partial \mathbf{r}_{ij}^k}{\partial \Phi_j} = \mathbf{C}_{\mathbf{\Phi}}^{-1} \tag{24}$$

i.e., the inverse covariance between cameras i and j depends only on the residuals of feature matches between i and j.

Similarly, $\mathbf{J}^T\mathbf{r}$ need not be computed explicitly, but can be computed via

$$(\mathbf{J}^T\mathbf{r})_i = \sum_{i=1}^{n} \sum_{j \epsilon \mathcal{I}(i)} \sum_{k \epsilon \mathcal{F}(i,j)} \frac{\partial \mathbf{r}_{ij}^k}{\partial \Phi_i}^T \mathbf{r}_{ij}^k. \tag{25}$$

In both cases each summation would require M multiplications if each feature matched to every single image, but in practice the number of feature matches for a given image is much less than this. Hence each iteration of bundle adjustment is $O(N^3)$, which is the cost of solving the $N \times N$ linear system. The number of parameters N is 4 times the number of images, and typically M is around 100 times larger than N.

5 Automatic Panorama Straightening

Image registration using the steps of sections 2 - 4 gives the relative rotations between the cameras, but there remains an unknown 3D rotation to a chosen world coordinate frame. If we simply assume that $\mathbf{R} = \mathbf{I}$ for one of the images, we typically find a wavy effect in the output panorama. This is because the real camera was unlikely to be perfectly level and un-tilted. We can correct this wavy output and automatically straighten the panorama by making use of a heuristic about the way people typically shoot panoramic images. The idea is that it is rare for people to *twist* the camera relative to the horizon, so the camera $\mathbf{X}$ vectors (horizontal axis) typically lie in a plane (see figure 4). By finding the null vector of the covariance matrix of the camera $\mathbf{X}$ vectors, we can find the "up-vector" $\mathbf{u}$ (normal to the plane containing the camera centre and the horizon)

$$\left(\sum_{i=1}^{n} \mathbf{X}_i \mathbf{X}_i^T\right) \mathbf{u} = \mathbf{0}. \tag{26}$$

Applying a global rotation such that up-vector $\mathbf{u}$ is vertical (in the rendering frame) effectively removes the wavy effect from output panoramas as shown in figure 4.

6 Gain Compensation

In previous sections, we described a method for computing the geometric parameters (orientation and focal length) of each camera. In this section, we show how to solve for a photometric parameter, namely the overall gain between images. This is set up in a similar manner, with an error function defined over all images. The error function is the sum of gain normalised intensity errors for all overlapping pixels

$$e = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{\substack{\mathbf{u}_i \epsilon \mathcal{R}(i,j) \\ \tilde{\mathbf{u}}_i = \mathbf{H}_{ij} \tilde{\mathbf{u}}_j}} (g_i I_i(\mathbf{u}_i) - g_j I_j(\mathbf{u}_j))^2 \tag{27}$$

where g_i, g_j are the gains, and $\mathcal{R}(i,j)$ is the region of overlap between images i and j. In practice we approximate $I(\mathbf{u}_i)$ by the mean in each overlapping region $\bar{I}_{ij}$

$$\bar{I}_{ij} = \frac{\sum_{\mathbf{u}_i \epsilon \mathcal{R}(i,j)} I_i(\mathbf{u}_i)}{\sum_{\mathbf{u}_i \epsilon \mathcal{R}(i,j)} 1}. \tag{28}$$

This simplifies the computation and gives some robustness to outliers, which might arise due to small misregistrations between the images. Also, since $\mathbf{g} = 0$ is an optimal solution to the problem, we add a prior term to keep the gains close to unity. Hence the error function becomes

$$e = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} N_{ij} \left((g_i \bar{I}_{ij} - g_j \bar{I}_{ji})^2 / \sigma_N^2 + (1 - g_i)^2 / \sigma_g^2 \right) \tag{29}$$

where $N_{ij} = |\mathcal{R}(i,j)|$ equals the number of pixels in image i that overlap in image j. The parameters σ_N and σ_g are the standard deviations of the normalised intensity error and gain respectively. We choose values $\sigma_N = 10.0$, ($I \, \epsilon \{0..255\}$) and $\sigma_g = 0.1$. This is a quadratic objective function in the gain parameters $\mathbf{g}$ which can be solved in closed form by setting the derivative to 0 (see figure 5).

7 Multi-Band Blending

Ideally each sample (pixel) along a ray would have the same intensity in every image that it intersects, but in reality this is not the case. Even after gain compensation some image edges are still visible due to a number of unmodelled effects, such as vignetting (intensity decreases towards the edge of the image), parallax effects due to unwanted motion of the optical centre, mis-registration errors due to mismodelling of the camera, radial distortion and so on. Because of this a good blending strategy is important.

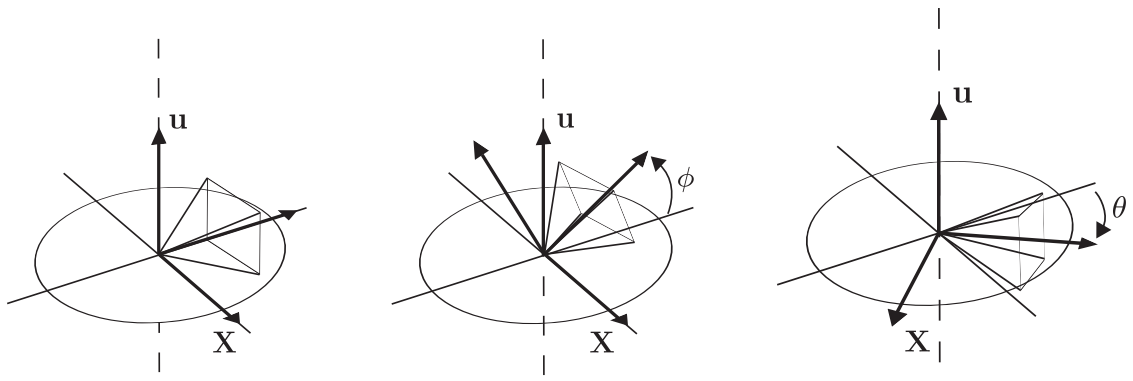
From the previous steps we have n images $I^i(x,y)$ ($i \; \epsilon \; \{1..n\}$) which, given the known registration, may be expressed in a common (spherical) coordinate system as $I^i(\theta,\phi)$. In order to combine information from multiple images we assign a weight function to each image $W(x,y) = w(x)w(y)$ where $w(x)$ varies linearly from 1 at the centre of the image to 0 at the edge. The weight functions are also resampled in spherical coordinates $W^i(\theta,\phi)$. A simple approach to blending is to perform a weighted sum of the image intensities along each ray using these weight functions

$$I^{linear}(\theta,\phi) = \frac{\sum_{i=1}^{n} I^i(\theta,\phi) W^i(\theta,\phi)}{\sum_{i=1}^{n} W^i(\theta,\phi)} \tag{30}$$

where $I^{linear}(\theta,\phi)$ is a composite spherical image formed using linear blending. However, this approach can cause blurring of high frequency detail if there are small registration errors (see figure 7). To prevent this we use the multi-band blending algorithm of Burt and Adelson [BA83]. The idea behind multi-band blending is to blend low frequencies over a large spatial range, and high frequencies over a short range.

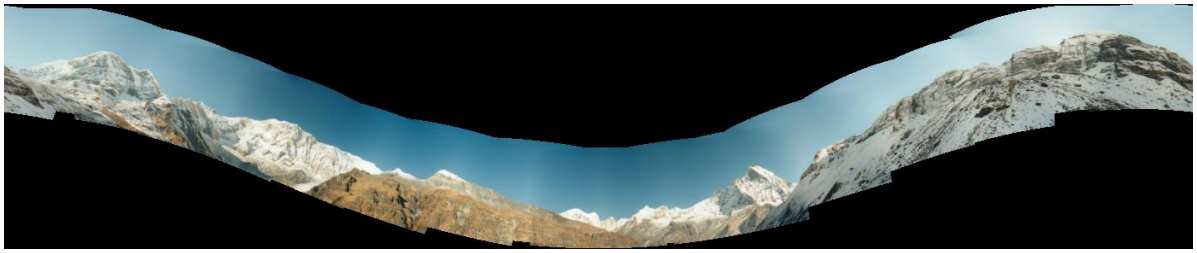
We initialise blending weights for each image by finding the set of points for which image i is most responsible

$$W^i_{max}(\theta,\phi) = \begin{cases} 1 & \text{if } W^i(\theta,\phi) = \arg\max_j W^j(\theta,\phi) \\ 0 & \text{otherwise} \end{cases} \tag{31}$$



(a) (b) (c)

Figure 3. Finding the up-vector **u**. A good heuristic to align wavy panoramas is to note that people rarely *twist* the camera relative to the horizon. Hence despite tilt (b) and rotation (c), the camera **X** vectors typically lie in a plane. The up-vector **u** (opposite to the direction of gravity) is the vector normal to this plane.



(a) Without automatic straightening

(b) With automatic straightening

Figure 4. Automatic panorama straightening. Using the heuristic that users rarely twist the camera relative to the horizon allows us to straighten wavy panoramas by computing the up-vector (perpendicular to the plane containing the horizon and the camera centre).

(a) Half of the images registered

(b) Without gain compensation

(c) With gain compensation

(d) With gain compensation and multi-band blending

Figure 5. Gain compensation. Note that large changes in brightness between the images are visible if gain compensation is not applied (a)-(b). After gain compensation, some image edges are still visible due to unmodelled effects such as vignetting (c). These can be effectively smoothed out using multi-band blending (d).

i.e. $W^i_{max}(\theta,\phi)$ is 1 for (θ,ϕ) values where image i has maximum weight, and 0 where some other image has a higher weight. These max-weight maps are successively blurred to form the blending weights for each band.

A high pass version of the rendered image is formed

$$B^i_\sigma(\theta,\phi) = I^i(\theta,\phi) - I^i_\sigma(\theta,\phi) \quad (32)$$
$$I^i_\sigma(\theta,\phi) = I^i(\theta,\phi) * g_\sigma(\theta,\phi) \quad (33)$$

where $g_\sigma(\theta,\phi)$ is a Gaussian of standard deviation σ, and the $*$ operator denotes convolution. $B_\sigma(\theta,\phi)$ represents spatial frequencies in the range of wavelengths $\lambda \in [0,\sigma]$. We blend this band between images using a blending weight formed by blurring the max-weight map for this image

$$W^i_\sigma(\theta,\phi) = W^i_{max}(\theta,\phi) * g_\sigma(\theta,\phi) \quad (34)$$

where $W^i_\sigma(\theta,\phi)$ is the blend weight for the wavelength $\lambda \in [0,\sigma]$ band. Subsequent frequency bands are blended using lower frequency bandpass images and further blurring the blend weights, i.e. for $k \geq 1$

$$B^i_{(k+1)\sigma} = I^i_{k\sigma} - I^i_{(k+1)\sigma} \quad (35)$$
$$I^i_{(k+1)\sigma} = I^i_{k\sigma} * g_{\sigma'} \quad (36)$$
$$W^i_{(k+1)\sigma} = W^i_{k\sigma} * g_{\sigma'} \quad (37)$$

where the standard deviation of the Gaussian blurring kernel $\sigma' = \sqrt{(2k+1)}\sigma$ is set such that subsequent bands have the same range of wavelengths.

For each band, overlapping images are linearly combined using the corresponding blend weights

$$I^{multi}_{k\sigma}(\theta,\phi) = \frac{\sum_{i=1}^n B^i_{k\sigma}(\theta,\phi) W^i_{k\sigma}(\theta,\phi)}{\sum_{i=1}^n W^i_{k\sigma}(\theta,\phi)}. \quad (38)$$

This causes high frequency bands (small $k\sigma$) to be blended over short ranges whilst low frequency bands (large $k\sigma$) are blended over larger ranges (see figure (6)).

Note that we have chosen to render the panorama in spherical coordinates θ,ϕ. In principle one could choose any 2-dimensional parameterisation of a surface around the viewpoint for rendering. One good choice would be to render to a triangulated sphere, constructing the blending weights in the image plane. This would have the advantage of uniform treatment of all images, and it would also allow easy resampling to other surfaces (in graphics hardware). Note that the θ,ϕ parameterisation suffers from singularities at the poles.

Algorithm: Automatic Panorama Stitching

Input: n unordered images

I. Extract SIFT features from all n images

II. Find k nearest-neighbours for each feature using a k-d tree

III. For each image:
  (i) Select m candidate matching images that have the most feature matches to this image
  (ii) Find geometrically consistent feature matches using RANSAC to solve for the homography between pairs of images
  (iii) Verify image matches using a probabilistic model

IV. Find connected components of image matches

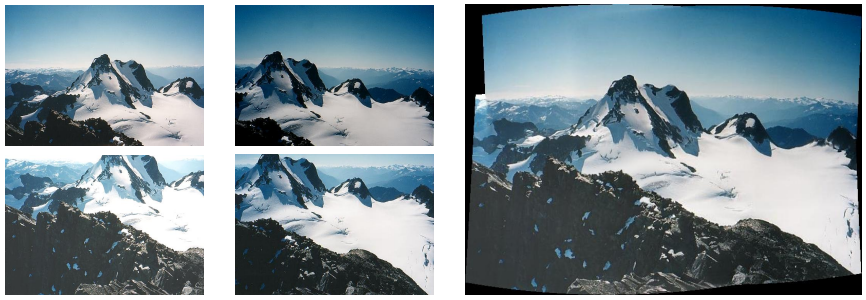
V. For each connected component:
  (i) Perform bundle adjustment to solve for the rotation $\theta_1,\theta_2,\theta_3$ and focal length f of all cameras
  (ii) Render panorama using multi-band blending

Output: Panoramic image(s)

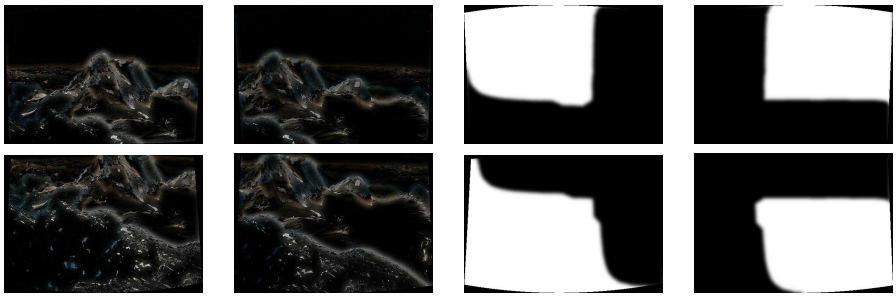
8 Results

Figure 2 shows typical operation of the panoramic recognition algorithm. A set of images containing 4 panoramas and 4 noise images was input. The algorithm detected connected components of image matches and unmatched images, and output 4 blended panoramas.

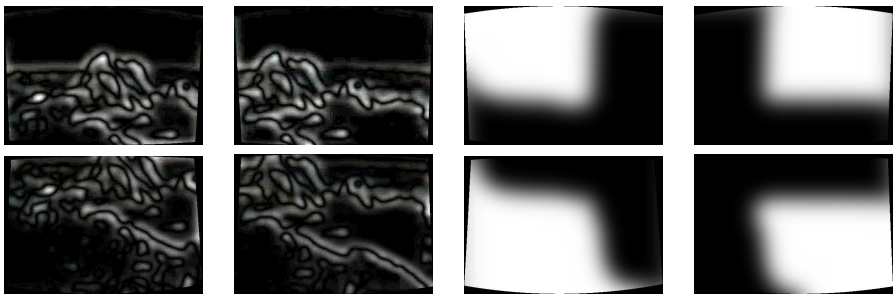
Figure 5 shows a larger example. This sequence was shot using the camera's automatic mode, which allowed the aperture and exposure time to vary, and the flash to fire on some images. Despite these changes in illumination, the SIFT features match robustly and the multi-band blending strategy yields a seamless panorama. The output is 360° $\times$ 100° degrees and has been rendered in spherical coordinates (θ,ϕ). All 57 images were matched fully automatically with no user input, and a $4 \times 57 = 228$ parameter optimisation problem was solved for the final registration. The 2272×1704 pixel input images were matched and registered in 60 seconds, and a further 15 minutes were taken to render the 8908×2552 (23 megapixel) output panorama. A 2000×573 preview was rendered in 57 seconds. Tests were conducted on a 1.6GHz Pentium M.



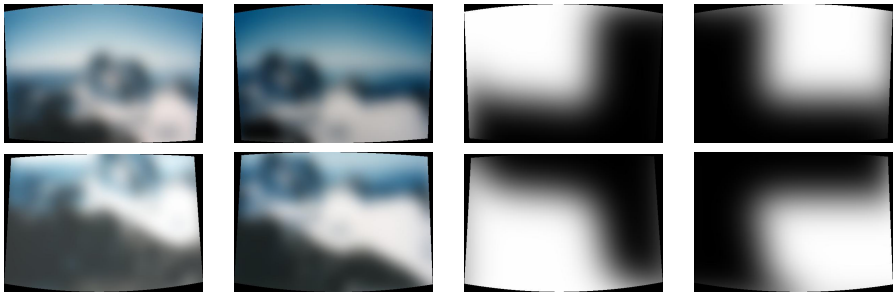
(a) Original images and blended result



(b) Band 1 (scale 0 to σ)



(c) Band 2 (scale σ to 2σ)



(d) Band 3 (scale lower than 2σ)

Figure 6. Multi-band blending. Bandpass images $B_{k\sigma}(\theta, \phi)$ for $k = 1, 2, 3$ are shown on the left, with the corresponding blending weights $W_{k\sigma}(\theta, \phi)$ shown on the right. Initial blending weights are assigned to 1 where each image has maximum weight. To obtain each blending function, the weights are blurred at spatial frequency σ and bandpass images of the same spatial frequency are formed. The bandpass images are blended together using weighted sums based on the blending weights (Note: the blending widths have been exaggerated for clarity in these figures).

(a) Linear blending (b) Multi-band blending

Figure 7. Comparison of linear and multi-band blending. The image on the right was blended using multi-band blending using 5 bands and $\sigma = 5$ pixels. The image on the left was linearly blended. In this case matches on the moving person have caused small misregistrations between the images, which cause blurring in the linearly blended result, but the multi-band blended image is clear.

(a) (b)

Figure 8. Stitching with rotation and zoom. Our use of invariant features make stitching possible despite rotation, zoom and illumination changes in the input images. Here the inset images at the base and tip of the tower are 4 times the scale of the other images.

9 Conclusions

This paper has presented a novel system for fully automatic panorama stitching. Our use of invariant local features and a probabilistic model to verify image matches allows us recognise multiple panoramas in unordered image sets, and stitch them fully automatically without user input. The system is robust to camera zoom, orientation of the input images, and changes in illumination due to flash and exposure/aperture settings. A multi-band blending scheme ensures smooth transitions between images despite illumination differences, whilst preserving high frequency details.

Future Work

Possible areas for future work include compensation for motion in the camera and scene, and more advanced modelling of the geometric and photometric properties of the camera:

Camera Motion Panoramas often suffer from parallax errors due to small motions of the optical centre. These could be removed by solving for camera translations and depths in the scene, before re-rendering from a central point. A good representation to use might be plane at infinity plus parallax [RC02]. Whilst gross camera motions cause parallax artifacts, small motions during shooting result in motion blur. Motion blurred images could be deblurred using nearby in-focus images as in [BBZ96]. Similar techniques can also be used to generate super-resolution images [CZ98].

Scene Motion Though our multi-band blending strategy works well in many cases, large motions of objects in the scene cause visible artifacts when blending between multiple images (see figure 10). Another approach would be to automatically find optimal seam lines based on regions of difference between the images [Dav98, UES01, ADA+04].

Advanced Camera Modelling An important characteristic of most cameras that is not included in the projective camera model (which preserves straight lines) is radial distortion [Bro71]. Whilst this is not explicitly modelled by our algorithm, we have tested the performance under moderate amounts of radial distortion (see figure 9). Although panorama recognition and approximate alignment is robust to radial distortion in our experiments, there are noticable artifacts in the rendered results. Hence high quality image stitching applications would need to include radial distortion parameters at least in the bundle adjustment and rendering stages. An ideal image stitcher would also support multiple motion models, for example, rotation about a point (e.g. panoramas), viewing a plane (e.g. whiteboards) and Euclidean transforms (e.g. aligning scanned images). One could also render to multiple surface types, e.g., spherical, cylindrical, planar.

Photometric Modelling In principle it should also be possible to estimate many of the photometric parameters of the camera. Vignetting (decrease in intensity towards image edges) is a common source of artifacts, particularly in uniform colour regions such as sky [GC05]. One could also acquire high-dynamic range [DM97, SHS+04] information from the overlapping image regions, and render tone mapped or synthetic exposure images.

We have developed a C++ implementation of the algorithm described in this paper, called Autostitch. A demo of this program can be downloaded from http://www.autostitch.net.

References

[ADA+04] A. Agarwala, M. Dontcheva, M. Agarwala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *ACM Transactions on Graphics (SIGGRAPH'04)*, 2004.

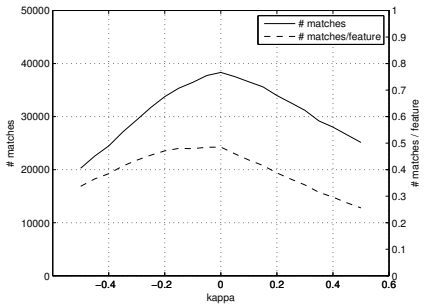
[BA83] P. Burt and E. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[BBZ96] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from and image sequence. In *Proceedings of the 4th European Conference on Computer Vision (ECCV96)*, pages 312–320. Springer-Verlag, 1996.

[BL97] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbor search in high-dimensional spaces. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR97)*, pages 1000–1006, 1997.

[BL03] M. Brown and D. Lowe. Recognising panoramas. In *Proceedings of the 9th International Conference on Computer Vision (ICCV03)*, volume 2, pages 1218–1225, Nice, October 2003.

[Bro71] D. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.



(a) Number of feature matches vs κ

(b) Example of stitching 2 images with $\kappa = -0.5$ (the full test set contains 44 images of this scene)

(c) Detail from the previous figure showing misregistration

(d) κ = -0.5

(e) κ = -0.25

(f) κ = 0

(g) κ = 0.25

(h) κ = 0.5

Figure 9. Stitching with radial distortion. This figure shows the effects on stitching of first order radial distortion $\mathbf{x}' = (1 + \kappa|\mathbf{x}|^2)\mathbf{x}$ with κ in the range $\kappa \in [-0.5, 0.5]$ (the image height is normalised to unit length). Note that radial distortion is *not* modelled in our algorithm. We used a test sequence of 44 images and applied radial distortion with 20 values of κ. Examples of the distorted images are given in figures (d)-(h). To evaluate the performance of stitching we counted the number of consistent matches after RANSAC, the results are shown in figure (a). Although the number of matches per feature dropped by around a third in the worst case, the number of correct feature matches was still high (around 500 per image), so the images could still be successfully matched. Nevertheless radial distortion causes visible artifacts in rendering as shown in figures (b)-(c), and correcting for this in the bundle adjustment and rendering stages would be important for high quality panorama stitching.

Figure 10. A difficult stitching problem. This example (from Times Square, New York) contains many moving objects and large changes in brightness between the images. Despite these challenges, our approach is able to find consistent sets of invariant features, and correctly register the images. Future automatic image stitchers could detect the moving objects, and compute high dynamic range radiance maps of the scene. This would enable the user to 're-photograph' the scene with different exposure settings and moving objects selected.

[BSW05] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR05)*, San Diego, June 2005.

[Che95] S. Chen. QuickTime VR – An image-based approach to virtual environment navigation. In *SIGGRAPH'95*, volume 29, pages 29–38, 1995.

[CZ98] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR98)*, pages 885–891, June 1998.

[Dav98] J. Davis. Mosaics of scenes with moving objects. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR98)*, pages 354–360, 1998.

[DM97] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. *Computer Graphics*, 31:369–378, 1997.

[FB81] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[GC05] D.B. Goldman and J.H. Chen. Vignette and exposure calibation and compensation. In *Proceedings of the 10th International Conference on Computer Vision (ICCV05)*, pages I: 899–906, 2005.

[Har92] C. Harris. Geometry from visual motion. In A. Blake and A. Yuille, editors, *Active Vision*, pages 263–284. MIT Press, 1992.

[Hub81] P.J. Huber. *Robust Statistics*. Wiley, 1981.

[HZ04] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[IA99] M. Irani and P. Anandan. About direct methods. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, number 1883 in LNCS, pages 267–277. Springer-Verlag, Corfu, Greece, September 1999.

[Low04] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[Mee90] J. Meehan. *Panoramic Photography*. Amphoto Books, September 1990.

[Mil75] D. Milgram. Computer methods for creating photomosaics. *IEEE Transactions on Computers*, C-24(11):1113–1119, November 1975.

[MJ02] P. McLauchlan and A. Jaenicke. Image mosaicing using sequential bundle adjustment. *Image and Vision Computing*, 20(9-10):751–759, August 2002.

[MSF] Microsoft Digital Image Pro. http://www.microsoft.com/products/imaging.

[RC02] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery using a reference plane. *International Journal of Computer Vision*, 49(2/3):117–141, 2002.

[REA] Realviz. http://www.realviz.com.

[SHS+04] H. Seetzen, W. Heidrich, W. Stuerzlinger, G. Ward, L. Whitehead, M. Trentacoste, A. Ghosh, and A. Vorozcovs. High dynamic range display systems. In *ACM Transactions on Graphics (SIGGRAPH'04)*, 2004.

[SK95] R. Szeliski and S. Kang. Direct methods for visual scene reconstruction. In *IEEE Workshop on Representations of Visual Scenes*, pages 26–33, Cambridge, MA, 1995.

[SK99] H. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):235–243, 1999.

[SS97] R. Szeliski and H. Shum. Creating full view panoramic image mosaics and environment maps. *Computer Graphics (SIGGRAPH'97)*, 31(Annual Conference Series):251–258, 1997.

[SS00] H. Shum and R. Szeliski. Construction of panoramic mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, February 2000.

[ST94] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR94)*, Seattle, June 1994.

[SZ03] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the 9th International Conference on Computer Vision (ICCV03)*, October 2003.

[Sze04] R. Szeliski. Image alignment and stitching: A tutorial. Technical Report MSR-TR-2004-92, Microsoft Research, December 2004.

[TMHF99] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment: A modern synthesis. In *Vision Algorithms: Theory and Practice*, number 1883 in LNCS, pages 298–373. Springer-Verlag, Corfu, Greece, September 1999.

[Tor02] P. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002.

[UES01] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition (CVPR01)*, volume 2, pages 509–516, Kauai, Hawaii, December 2001.

[ZFD97] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2D mosaic from a set of images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition, Puerto Rico*. IEEE, June 1997.