Recognising Panoramas

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

Abstract

The problem considered in this paper is the fully automatic construction of panoramas. Fundamentally, this problem requires recognition, as we need to know which parts of the panorama join up. Previous approaches have used human input or restrictions on the image sequence for the matching step. In this work we use object recognition techniques based on invariant local features to select matching images, and a probabilistic model for verification. Because of this our method is insensitive to the ordering, orientation, scale and illumination of the images. It is also insensitive to 'noise' images which are not part of the panorama at all, that is, it recognises panoramas. This suggests a useful application for photographers: the system takes as input the images on an entire flash card or film, recognises images that form part of a panorama, and stitches them with no user input whatsoever.

1. Introduction

Panoramic image mosaicing has an extensive research literature and there are several commercial offerings that come bundled with today's digital cameras. The geometry of the problem is well understood, and consists of estimating a 3×3 camera matrix or homography for each image [7, 15]. However, this estimation process needs an initialisation. In commercial applications this usually takes the form of user input, or a fixed ordering to the images. For example, the PhotoStitch software bundled with Canon digital cameras requires a horizontal or vertical sweep, or a square matrix of images. The REALVIZ Stitcher [1] has a user interface to roughly position the images with a mouse, before automatic registration proceeds.

In the research literature methods for automatic image matching / geometry estimation fall broadly into two camps: direct and feature based. Feature based methods [16, 6] begin by establishing correspondences between points, lines or other geometrical entities. For example, a typical approach would be to extract Harris corners and use a normalised cross-correlation of the local intensity values to match them. Direct methods [8, 14] attempt to iteratively estimate the camera parameters by minimising an error function based on the intensity difference in the area of overlap. Direct methods have the advantage that they use all of the available data and hence can provide very accurate registration, but they depend on the fragile 'brightness constancy' assumption, and being iterative require initialisation. However (assuming that conventional features are used) neither of these methods are robust to image zoom, change in illumination or to 'noise' images which are not part of the sequence.

Recently there has been great progress in the use of invariant features [12, 2, 4] for object recognition and matching. These features can be found more repeatably and matched more reliably than traditional methods such as cross-correlation using Harris corners. Harris corners are not invariant to scaling of the image, and cross-correlation of image patches is not invariant to rotation. However invariant features are designed to be invariant to these transformations. In this work we use Lowe's [9] Scale Invariant Feature Transform (SIFT features), which are geometrically invariant under similarity transforms and invariant under affine changes in intensity.

Although matching n images may seem to have quadratic complexity, it can be reduced to $O(n \log n)$. This is done by matching features using an approximate nearest neighbour algorithm [3]. A similar approach was taken by Schaffalitzky and Zisserman [11] in the context of structure from motion. Our work differs from this in our use of panoramic image geometry. In addition we introduce a probabilistic model for image match verification. This gives us a principled framework for rejecting noise images and recognising multiple panoramas in an unordered image set.

The process of optimising over the camera parameters to minimise the matching error is known as bundle adjustment [17]. We use as our objective function a robustified sum squared error of all feature matches. This yields a

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)
0-7695-1950-4/03 $ 17.00 © 2003 IEEE

non-linear least squares problem which we solve using the Levenberg-Marquadt algorithm. To seamlessly blend the panorama we use a multi-band blending strategy as proposed by Burt and Adelson [5]. This enables smooth transitions between the images with different overall intensities, whilst preserving sharp detail even in the presence of small mis-registrations.

2. Feature Matching

The first step in the panoramic recognition algorithm is to extract and match SIFT 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. The vector of gradients is normalised, and since it consists of differences of intensity values, it is invariant to affine changes in intensity.

Assuming that the camera rotates about it's optical centre, the group of transformations the images may undergo is a special group of homographies. We parameterise each camera by 3 rotation angles $\boldsymbol{\theta} = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix}$ and focal length f. This gives pairwise homographies $\tilde{\mathbf{u}}_i = \mathbf{H}_{ij}\tilde{\mathbf{u}}_j$

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

where

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

and (using the exponential representation for rotations)

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

However, for small changes in image position

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

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

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

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 it's k nearest neighbours (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[3].

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 neighbouring 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. 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

$$\begin{aligned} p(f^{(1:n_f)}|m = 1) &= B(n_i; n_f, p_1) \\ p(f^{(1:n_f)}|m = 0) &= B(n_i; n_f, p_0) \end{aligned}$$

where p_1 is the probability a feature is an inlier given a correct image match, and p_0 is the probability a feature

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)
0-7695-1950-4/03 $ 17.00 © 2003 IEEE

is an inlier given a false image match. The number of inliers $n_i = \sum_{i=1}^{n_f} f^{(i)}$. We choose values $p_1 = 0.7$ and $p_0 = 0.01$. Now we can evaluate the posterior probability that an image match is correct using Bayes' Rule

$$\begin{aligned} p(m=1|f^{(1:n_f)}) &= \frac{p(f^{(1:n_f)}|m=1)p(m=1)}{p(f^{(1:n_f)})} \\ &= \frac{1}{1+\frac{p(f^{(1:n_f)}|m=0)p(m=0)}{p(f^{(1:n_f)}|m=1)p(m=1)}} \end{aligned}$$

We accept an image match if $p(m=1|f^{(1:n_f)}) > p_{min}$. Assuming a uniform prior $p(m=1) = p(m=0)$, this reduces to a likelihood ratio test:

$$\frac{B(n_i; n_f, p_1)}{B(n_i; n_f, p_0)} \underset{reject}{\overset{accept}{\gtrless}} \frac{1}{\frac{1}{p_{min}} - 1}$$

Chosing a value $p_{min} = 0.97$ gives the condition

$$n_i > 5.9 + 0.22 n_f$$

for a correct image match.

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 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 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.

Given a correspondence $\mathbf{u}_i^k \leftrightarrow \mathbf{u}_j^l$ ($\mathbf{u}_i^k$ denotes the position of the kth feature in image i), the residual is

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

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

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

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

$$e = \sum_{i=1}^{n} \sum_{j \epsilon \mathcal{I}(i)} \sum_{k \epsilon \mathcal{F}(i,j)} f(\mathbf{r}_{ij}^k)^2$$

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, and $f(\mathbf{x})$ is a robust error function

$$f(\mathbf{x}) = \begin{cases} |\mathbf{x}|, & \text{if } |\mathbf{x}| < x_{max} \\ x_{max}, & \text{if } |\mathbf{x}| \geq x_{max} \end{cases}$$

We use $x_{max} = \infty$ during initialisation and $x_{max} = 1$ pixel 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

$$\mathbf{\Theta} = (\mathbf{J}^T \mathbf{J} + \sigma^2 \mathbf{C}_p^{-1})^{-1} \mathbf{J}^T \mathbf{r}$$

where $\mathbf{\Theta}$ are all the parameters, $\mathbf{r}$ the residuals and $\mathbf{J} = \partial \mathbf{r} / \partial \mathbf{\Theta}$. We encode our prior belief about the parameter changes in the covariance matrix $\mathbf{C}_p$. This is set such that the standard deviation of angles is $\sigma_\theta = \pi/16$ and focal lengths $\sigma_f = \bar{f}/10$. 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 focal length. Finally, σ represents the standard deviation of projection errors and is varied via Levenberg-Marquardt to ensure that the objective function does in fact decrease at each iteration.

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

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

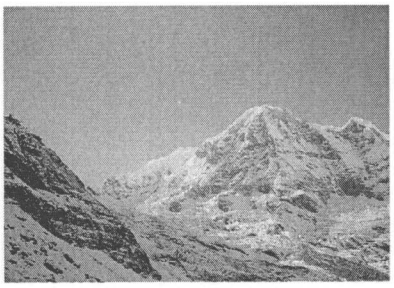
where

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

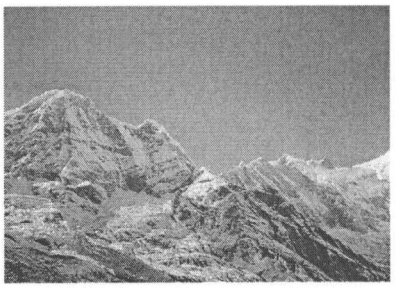
and

$$\begin{aligned} \frac{\partial \tilde{\mathbf{p}}_{ij}^k}{\partial \theta_{i1}} &= \mathbf{K}_i \frac{\partial \mathbf{R}_i}{\partial \theta_{i1}} \mathbf{R}_j \mathbf{K}_j^{-1} \tilde{\mathbf{u}}_j^l \\ \frac{\partial \mathbf{R}_i}{\partial \theta_{i1}} &= \frac{\partial}{\partial \theta_{i1}} e^{[\boldsymbol{\theta_i}]_\times} = e^{[\boldsymbol{\theta_i}]_\times} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned}$$

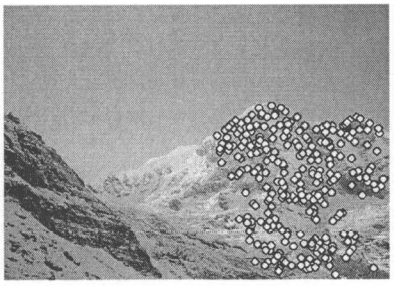
Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)
0-7695-1950-4/03 $ 17.00 © 2003 IEEE



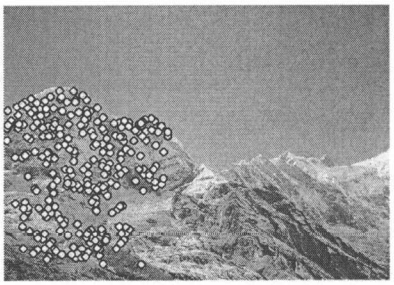
(a) Image 1



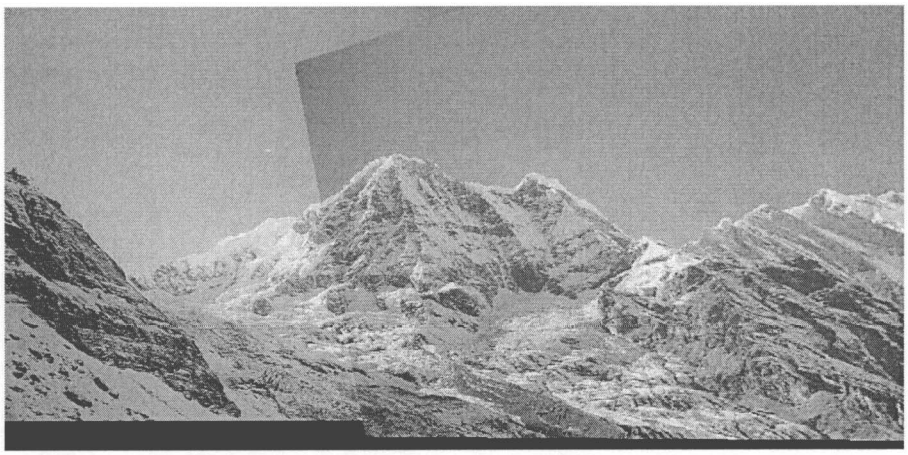
(b) Image 2



(c) SIFT matches 1



(d) SIFT matches 2



(e) 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.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)
0-7695-1950-4/03 $ 17.00 © 2003 IEEE

5. 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. There are a number of reasons for this: change in aperture/exposure time, vignetting (intensity decreases towards the edge of the image), parallax effects due unwanted motion of the optical centre, and any mis-registration errors due to mis-modelling of the camera, radial distortion etc. Because of this a good blending strategy is important.

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. A simple approach to blending is to perform a weighted sum of the image intensities along each ray using these weight functions. However, this can cause blurring of high frequency detail if there are small registration errors. To prevent this we have applied the multi-band blending strategy developed by Burt and Adelson [5]. The idea behind multi-band blending is to blend low frequencies over a large spatial range, and high frequencies over a short range. This can be performed over multiple frequency bands using a Laplacian Pyramid.

In our implementation we have used a simple 2 band scheme. A low pass image is formed with spatial frequencies of wavelength greater than 2 pixels relative to the rendered image, and a high pass image with spatial frequencies less than 2 pixels. We then blend the low frequency information using a linear weighted sum, and select the high frequency information from the image with the maximum weight.

Whilst it would be desirable to use more frequency bands in the blending scheme, an open problem is to design suitable spline functions for arbitrarily overlapping images.

6. Results

Figure (2) shows typical operation of the panoramic recognition algorithm. A set of images containing 2 panoramas and 5 noise images was input. The algorithm detected 2 connected components of image matches and 5 unmatched images, and output the 2 blended panoramas. The complete algorithm ran in 83 seconds on a 2GHz PC, with input images 525×375 pixels ($7'' \times 5''$ prints scanned at 75 dpi), and rendering the larger output panorama as a 300×3000 pixel image. The majority of computation time is spent in extracting the SIFT features from the images.

Figure (3) shows a larger example where 80 images were used to create a $360° \times 90°$ panorama. No user input is required: the object recognition system decides which images match, and the bundle adjustment algorithm optimises jointly for the $4 \times 80 = 320$ parameters of all the cameras. Finally, the multi-band blending scheme effectively hides the seams despite the illumination changes (camera flash and change in aperture/exposure).

Algorithm: Panoramic Recognition

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 (with the maximum number of feature matches to this image)
  (ii) Find geometrically conisistent feature matches using RANSAC to solve for the homography between pairs of images
  (iii) Verify image matches using 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)

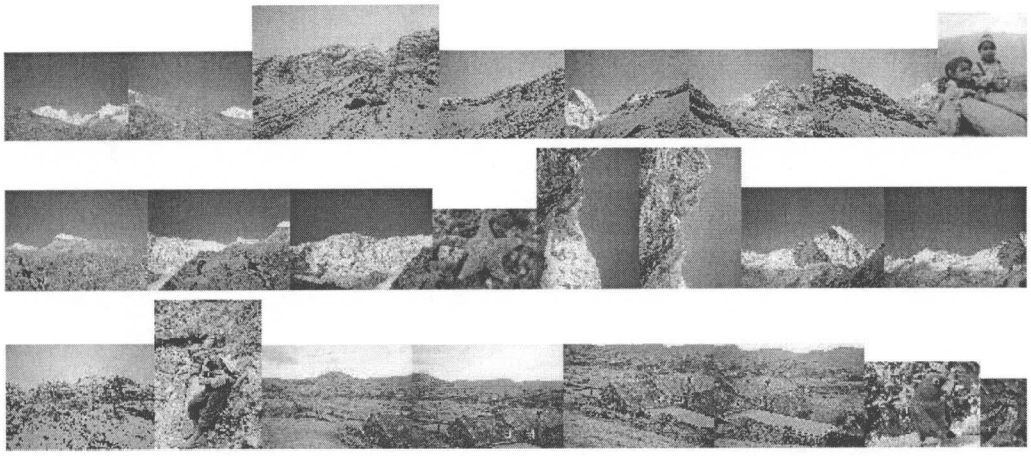
We have tested the system on many other image sets, for example full $360° \times 180°$ panoramas, and sequences where different cameras are used in the same panorama. Further examples can be found online at http://www.cs.ubc.ca/~mbrown/panorama/panorama.html.

7. Conclusions

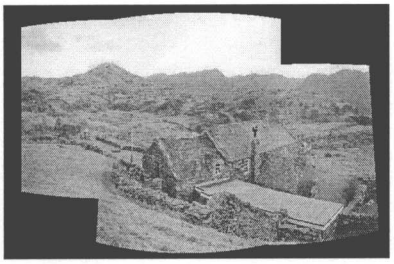
This paper has presented a novel system for fully automatic panorama stitching. Our use of invariant local features and a probabilistic model to verify image matches allows us recognise multiple panoramas in unordered image sets, and stitch them fully automatically without user input. The system is robust 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.

Possible future directions would be to attempt to distin-

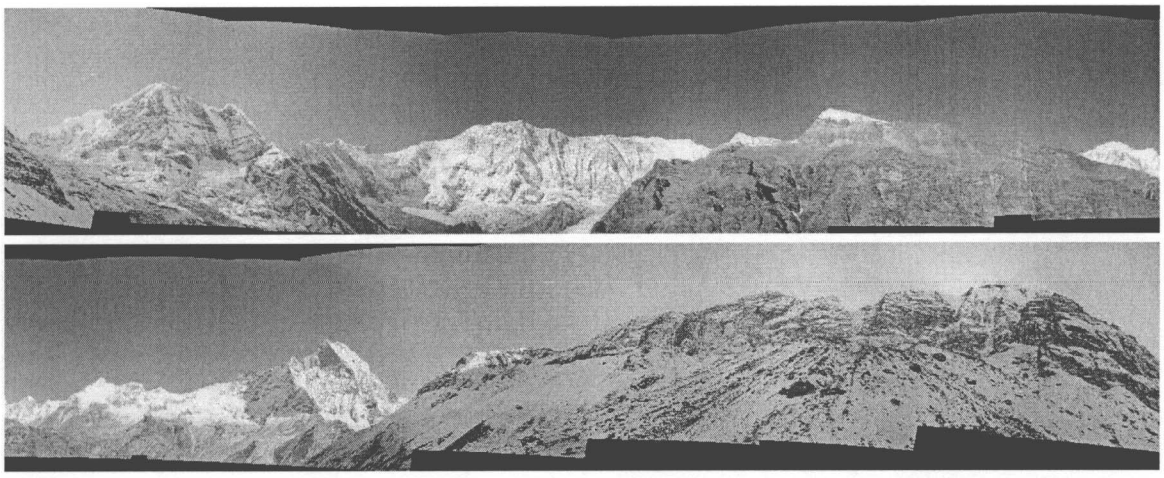
Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)
0-7695-1950-4/03 $ 17.00 © 2003 IEEE



(a) Input images



(b) Output panorama 1

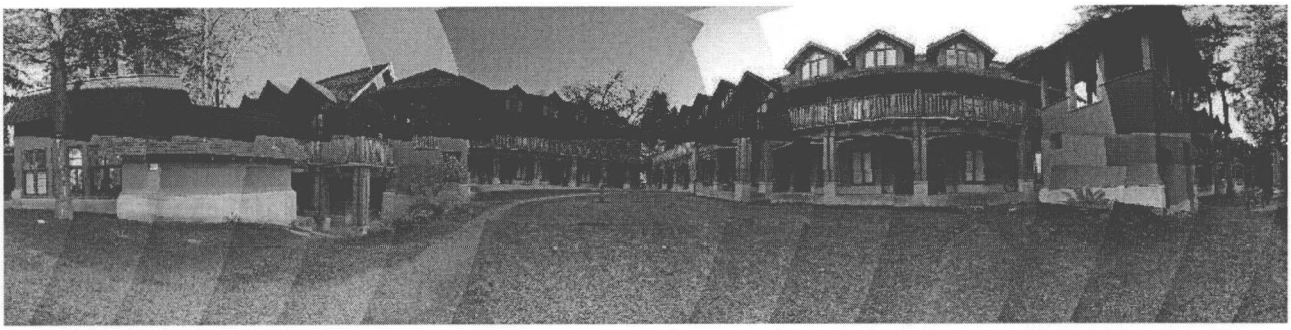


(c) Output panorama 2

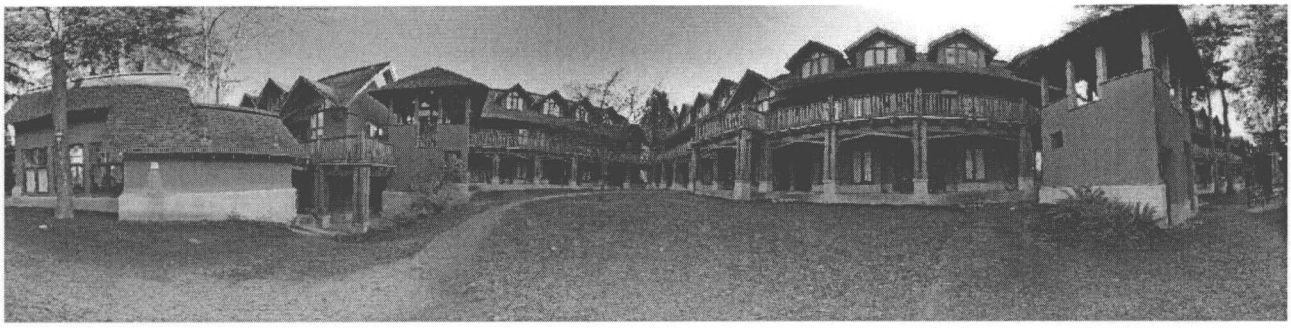
Figure 2. Typical operation of the panoramic recognition algorithm: an image set containing multiple panoramas and noise images is input, and panoramic sequences are recognised and rendered as output. The algorithm is insensitive to the ordering, scale and orientation of the images. It is also insensitive to noise images which are not part of a panorama. Note that the second output panorama is actually 360°, but has been split here for display purposes.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)
0-7695-1950-4/03 $ 17.00 © 2003 IEEE

(a) 40 of 80 images registered



(b) All 80 images registered



(c) Rendered with multi-band blending

Figure 3. Green College. 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. These $360° \times 90°$ images have been rendered in spherical coordinates (θ, ϕ). The sequence consisted of 80 images all of which were matched fully automatically with no user input, and a $4 \times 80 = 320$ parameter optimisation problem was solved for the final registration. With 400×300 pixel input images and a 500×2000 pixel output panorama, the whole process ran in 197 seconds on a 2GHz PC.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)
0-7695-1950-4/03 $ 17.00 © 2003 IEEE

guish between 3D imagery (camera translation) and panoramas (camera fixed), and to use more frequency bands in the blending scheme.

References

[1] REALVIZ. http://www.realviz.com.

[2] A. Baumberg. Reliable Feature Matching Across Widely Separated Views. In *Proceedings of the Interational Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000.

[3] 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*, pages 1000–1006, 1997.

[4] M. Brown and D. Lowe. Invariant Features from Interest Point Groups. In *Proceedings of the 13th British Machine Vision Conference*, pages 253–262, Cardiff, 2002.

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

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

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[8] 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.

[9] D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, September 1999.

[10] H. S. 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.

[11] F. Schaffalitzky and A. Zisserman. Multi-view Matching for Unordered Image Sets, or "How Do I Organise My Holiday Snaps?". In *Proceedings of the European Conference on Computer Vision*, pages 414–431, 2002.

[12] C. Schmid and R. Mohr. Local Grayvalue Invariants for Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997.

[13] 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.

[14] 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.

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

[16] P. Torr and A. Zisserman. Feature Based Methods for Structure and Motion Estimation. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, number 1883 in LNCS, pages 278–294. Springer-Verlag, Corfu, Greece, September 1999.

[17] 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.

[18] 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*, volume 2, pages 509–516, Kauai, Hawaii, December 2001.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)
0-7695-1950-4/03 $ 17.00 © 2003 IEEE