Video, Vision and Graphics (2003)
P. Hall, P. Willis (Editors)

Collaborative Vision and Interactive Mosaicing

John A. Robinson

Media Engineering, Dept. of Electronics, University of York, Heslington, York, YO10 5DD

Abstract

I propose criteria for collaborative vision applications where a camera user/operator and a computer work together to analyse a scene. An example of how these may be fulfilled is provided in IMP – an interactive mosaicing program. IMP generates mosaics in real-time, interacting with the user to cue camera movement and relay performance information.

1. Introduction

The development of fast methods in geometric computer vision makes it possible for a camera user and a computer to work together to analyse a scene. A *collaborative vision* application that constructs a 3D model of a space as a user videos it could, for example, provide advice about where the user should move to fill in gaps in the model, request higher resolution imagery (e.g. by advising the user to zoom into a distant corner of the ceiling) or signal where parts of the scene should be revisited if foreground motion has interfered with the 3D reconstruction. A long term goal is to realize such an application and embed it in a smart handheld camera so that on-the-spot 3D capture of film sets, crime scenes, homes for sale, etc. becomes possible. Such an application is probably some years away, but it is already possible to do more elementary kinds of geometric scene analysis in real time. Image mosaicing, for example, relies on estimation of projective transforms between frames of a video sequence, and fast algorithms are available. The object of this paper is to identify general characteristics for collaborative vision algorithms, and demonstrate their embedding in an interactive mosaicing application.

2. Criteria for Collaborative Vision

A *collaborative vision* system is one that leverages the user or operator's judgement and skills in a real-time vision task. To accomplish this, the system should fulfill various criteria.

1. It should be immediately responsive to a user's actions, and therefore fast,
2. It should be causal (That is, in the processing of an input frame, it can only use previous frames, not future ones. Of course, an acquired video sequence may be post-processed after the collaborative vision phase with a global method.),
3. It should evaluate the reliability of its inferences, so it can signal the user where there is uncertainty,
4. It should identify actions that the user could take to help in accomplishing the task, and signal these to the user,
5. It should allow easy user correction of its inferences.

In image mosaicing, the aim is to construct a single panoramic image from the video sequence generated as a user pans, tilts and zooms their camera. This is accomplished by estimating projective transforms[1,2] (or some simpler geometric transformation[3,4]) between the frames in the sequence, then generating a composite image made up of warped versions of the input frames. Various things can go wrong – foreground objects can appear and move, scene structure may be regular enough to confound the estimation process, camera movement between frames can result in blur. Interactive feedback is therefore useful while mosaicing, even if the final panorama is constructed post-capture using a global algorithm.

IMP is an interactive mosaicing program in which the five general requirements above are fulfilled as follows:

1. The system can mosaic 320x240 pel frames at 5 frames/s on a standard laptop (2GHz PIII).
2. It uses the immediately previous frames plus any earlier frames that are in the same region of the mosaic to make a causal estimation of the projective transform.
3. It relays a registration disparity measure to the user. The feedback mechanism changes for high disparities, to alert the user to take corrective action.
4. The system continuously displays the current state of the mosaic, so the user can determine how to move the camera.
5. The system is self-correcting when the user revisits areas of error.

The remainder of the paper is a description of *IMP* that expands these five points. First the underlying projective transform estimation algorithm is briefly reviewed. The speed of this algorithm is what allows the system to run at close to real time (point 1). Next, the forward/backward mosaicing method introduced in *IMP* is explained (point 2). Finally, interactive features are identified and illustrated (points 3, 4 and 5).

3. The SAM Estimator

The Simplex Adaptive Mesh (SAM) method used to estimate homographies in *IMP* is described elsewhere[5]. Here its operation is briefly summarized.

The projective transform from an image in $[x, y]^{\mathrm{T}}$ to an image in $[\hat{x}, \hat{y}]^{\mathrm{T}}$ is given by

$$\begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \end{bmatrix} = \frac{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}{\begin{bmatrix} c_1 & c_2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + 1} \qquad (1)$$

Estimation of the transform requires a search in eight-dimensional parameter space for a set $\{a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2, c_1, c_2\}$ that when applied to all the points in one image gives the closest possible match to the other image. Luminance values must be used to steer the search. A direct algorithm (like references 1,2 and SAM, as opposed to a feature-based algorithm[6]) works by minimizing some disparity measure between values in one image and values at corresponding transformed coordinates in the other. The usable domain and range of the transform depend on the image sizes and the transform parameters. Only the overlapping region of one image and a transformed version of the other can be used in a disparity calculation. In *IMP*, although estimation remains pairwise rather than global, the "previous" image against which the current image is compared is a backward mosaic rather than the previous input frame. It therefore contains information from earlier in the sequence (see below).

SAM has two stages – a translation estimator and a projective transform estimator. The first stage is conventional, using full search at the top level of an image pyramid, yielding estimates that are refined by projection down to successive levels with gradient descent minimization in each level. The second stage is a generate-and-test optimization procedure that works as follows.

SAM uses a mesh or grid of coordinates for which candidate transform values are calculated and which then sample the two images. A disparity value is computed from a weighted sum of the absolute differences in sample values. The weighting attenuates high absolute differences to limit the effect of localized moving objects. The weighted disparity is used as the criterion function for optimization by the Nelder-Mead simplex method[7]. The simplex requires the maintenance of 9 candidate transforms and their iterative adjustment. Each candidate transform is a particular set of the eight parameters a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2, c_1, c_2, in equation 1 and therefore corresponds to a point in 8-dimensional parameter space. These nine points form the vertices of the simplex. Geometrically, the simplex method involves changing the shape of this hypersolid by systematic movement of the vertices towards the minimum, until they are close enough together to meet a termination condition. SAM does the initialization of vertices in a systematic way by one-dimensional searches through parameter space according to the expected variation in each of the dimensions. If the initialization values are unsuitable for a particular case, the simplex changes shape to move quickly towards better vertices. In doing so, it grows in size, automatically lengthening the search time, but ensuring that, once values in the vicinity of the minimum are found, the simplex will converge slowly enough to avoid false minima.

4. Forward and Backward Mosaics for Estimation and Interactivity

Input video frames are processed sequentially by the Simplex Adaptive Mesh algorithm, then added, appropriately transformed, to two current mosaics. The *forward* mosaic is a representation anchored at the start frame of the sequence. That is, each frame's transformation is composed with previous transformations to render it relative to the unwarped first frame. The *backward* mosaic is anchored at the frame most recently added. So in the backward mosaic, the most recent frame is centred and untransformed. For each input frame, SAM estimates the projective transform between the current backward mosaic and the new frame. The new frame is transformed and added to the current forward mosaic. Now a window around the current frame in the forward mosaic is inverse transformed to produce the backward mosaic. This mosaic is shown to the user, and, as mentioned above, used as the base for estimating the transform of the next incoming frame. Figure 1 provides a schematic description of the method for the first three frames. All subsequent frames are handled similarly to frame 3 in the diagram.

Maintaining forward and backward mosaics is an efficient way to prevent the accumulation of error. *IMP* requires only one transform estimation per frame, in contrast to schemes that match each incoming frame to a plurality of others to achieve global registration. But because a mosaic (rather than just the previous frame) is used, all nearby earlier frames are included in the single pairwise estimate. An estimation error produces a distortion in the mosaic, which works to prevent further error accumulation. Figure 2 illustrates this schematically, showing how frames after an error are pulled back into alignment by the effect of earlier, correctly aligned, frames in the backward mosaic. Figures 3 and 4 demonstrate the effect for an input sequence due to Davis[3]. Figure 3 shows the accumulation of error with pairwise estimation between successive frames. Only part of the mosaic is included, because later frames are so badly warped the picture becomes unrecognisable. Davis (among others) justifies global estimation on the basis of results like this. As shown in figure 4, *IMP* corrects for the error shown in figure 3, and for later errors, without global estimation – simply by using the backward mosaic rather than the previous frame.

Because only two transformations are ever applied to any frame (the transformation to the forward mosaic, then the transformation to the current backward mosaic), roundoff does not accumulate as it would from maintenance of a single backward mosaic. The backward mosaic and the final mosaic, re-rendered with any projective transform (e.g. the one of the central frame in the sequence), are of high quality. Figures 5-7 illustrate this for *IMP* working in real time. Figure 5 shows every tenth frame of a sequence where a webcam is waved across a tower; figure 6 is the backward mosaic presented to the user halfway through the sequence; figure 7 is the final forward mosaic.

5. Other Interactive Features of *IMP*

IMP provides visual and aural feedback to the user. The current state of the backward mosaic is always displayed and the magnitude of the matching disparity is represented by an audible tone. When the disparity exceeds a threshold, SAM's estimate is not reliable, so the system displays the current input frame in monochrome and does not add it to the mosaic. Figure 8 shows an example for the tower sequence in figure 1, where false matching with the repeated structure of the tower was signalled. When an input frame yields a reliable transform estimate, the program goes back to accumulating mosaics. Indeed the final mosaic in figure 7 was constructed from the sequence of which figure 8 was a part. Minor estimation errors may not cause the disparity to exceed the threshold. These are revealed in the mosaic being displayed, and the user is able to sweep the camera back to the part of the scene that is in error. Finally, the user does not need to look at the backward mosaic if the cue tone remains low-pitched: in this case all the matches are good.

The ways in which *IMP* demonstrates the concepts of collaborative vision can be summarized with reference to the original five criteria.

1. All three major steps are fast. SAM is an efficient projective transform estimator. Pasting the current frame onto the forward mosaic requires an image warp, but no sophisticated stitching mechanism to hide image boundaries. The backward mosaic is only 3 frames high and 3 frames wide, so the final transformation from forward to backward mosaic is fast.
2. The system does not attempt global registration of frames. Instead the backward mosaic centred on the previous frame accumulates all previous frames in the region and provides reliable pairwise estimation.
3. The feedback mechanisms are simple because they rely on a single measured quantity – the disparity. The audio tone gives continuous feedback on how well *IMP* thinks it is doing, while the real-time mosaic reveals inaccuracies that the system has not detected. The switch to monochrome frames, together with a warning tone, asks the user to re-align the input video with the mosaic, which is simply done by moving the camera.
4. The system displays a window on the current mosaic canvas. It is easy to see areas that have not yet been visited. Moreover, the mosaic is displayed centred on the current frame, so by moving the camera it is possible to compare projections for the final mosaic.
5. The system is self-correcting when the user revisits areas of error. New frames simply overwrite the erroneous part of the mosaic.

6. Testing and Evaluation

IMP has been extensively tested with live video and stored sequences (such as that of figure 4). Further illustrative results are shown in figures 9 and 10. The first of these shows a mosaic constructed from a webcam input. The camera was panned, tilted and rolled with the centre of rotation about 10 cm from the optical centre. The sequence includes multiple revisits with the camera at different angles. The second is a particularly difficult test case because some of the middle frames of the sequence consist almost entirely of moving water. With ill-defined, dynamic features, this has caused mis-estimation, visible in the bending of the rail at the bottom. However, the buildings beyond the lake are well aligned, showing that the method is conservative in its estimations when there is little information to register. That is, it does not make erratic estimations in the absence of good features.

More formal testing of SAM is underway, but it is not yet clear what kinds of subjective tests are appropriate to the evaluation of *IMP*'s interactivity. We are liaising with human factors researchers to identify appropriate experiments.

7. Conclusion

IMP is an interactive mosaicing program that demonstrates how some of the characteristic of collaborative vision can be fulfilled.

We are now investigating the addition of depth estimation to the real-time mosaicing facility. The mechanisms being developed are based on those of References 8,9. The question of how to provide an informative, responsive, collaborative interface for these is central to our future work.

References

[1] R Szeliski, "Image Mosaicing for Tele-Reality Applications", Digital Equipment Corporation Cambridge Research Laboratory, Technical Report CRL 94/2, May 1994.
[2] S Mann, R Picard, "Video Orbits of the Projective Group: A simple approach to featureless estimation of parameters", IEEE Trans Image Proc, Vol 6, No 9, 1997, pp 1281-1295.
[3] J Davis, "Mosaics of Scenes with Moving Objects", Proc CVPR'98, June 1998.
[4] S Peleg, J Herman, "Panoramic Mosaics with VideoBrush, "DARPA Image Understanding Workshop, May 1997, pp 261-264.
[5] J A Robinson, "A Simplex-Based Projective Transform Estimator", Visual Information Engineering 2003, to appear, July 2003.
[6] P H S Torr, A Zisserman, "Feature Based Methods for Structure and Motion Estimation", and following discussion, in B Triggs, A Zisserman, R Szeliski (eds.), "Vision Algorithms: Theory and Practice. Proc Internat Workshop on Vision Algorithms, Corfu, Sept 1999, Springer-Verlag Lecture Notes in Computer Science, 1883, pp 267-297.
[7] J A Nelder, R Mead, "A Simplex Method for Function Minimization", The Computer Journal, Vol 7, 1965, pp 308-313.
[8] R Kumar, P Anandan, K Hanna, "Shape Recovery from Multiple Views: A Parallax Based Approach", Proc 12th Internat Conference on Pattern Recognition, 1994, pp685-688.
[9] M Irani, B Rousso, S Peleg, "Recovery of Ego-Motion using Region Alignment", IEEE Trans PAMI, Vol 19, No 3, 1997, pp 268-272.

[Figures on following pages]

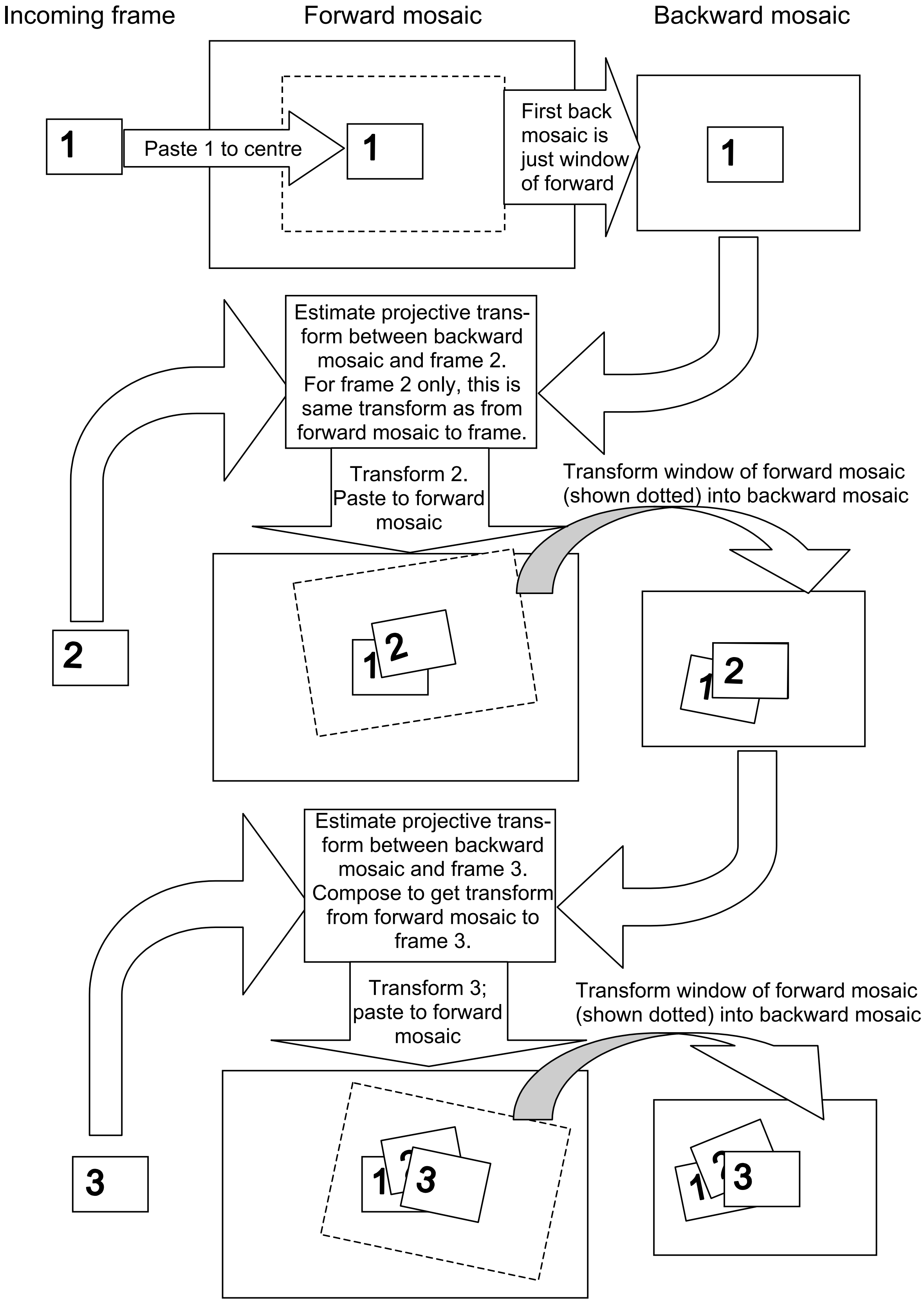
Incoming frame
Forward mosaic
Backward mosaic
1
Paste 1 to centre
1
First back mosaic is just window of forward
1
Estimate projective transform between backward mosaic and frame 2. For frame 2 only, this is same transform as from forward mosaic to frame.
Transform 2. Paste to forward mosaic
Transform window of forward mosaic (shown dotted) into backward mosaic
2
1
2
1
2
Estimate projective transform between backward mosaic and frame 3. Compose to get transform from forward mosaic to frame 3.
Transform 3; paste to forward mosaic
Transform window of forward mosaic (shown dotted) into backward mosaic
3
1
2
3
1
2
3

Fig 1. Schematic of the use of forward and backward mosaics

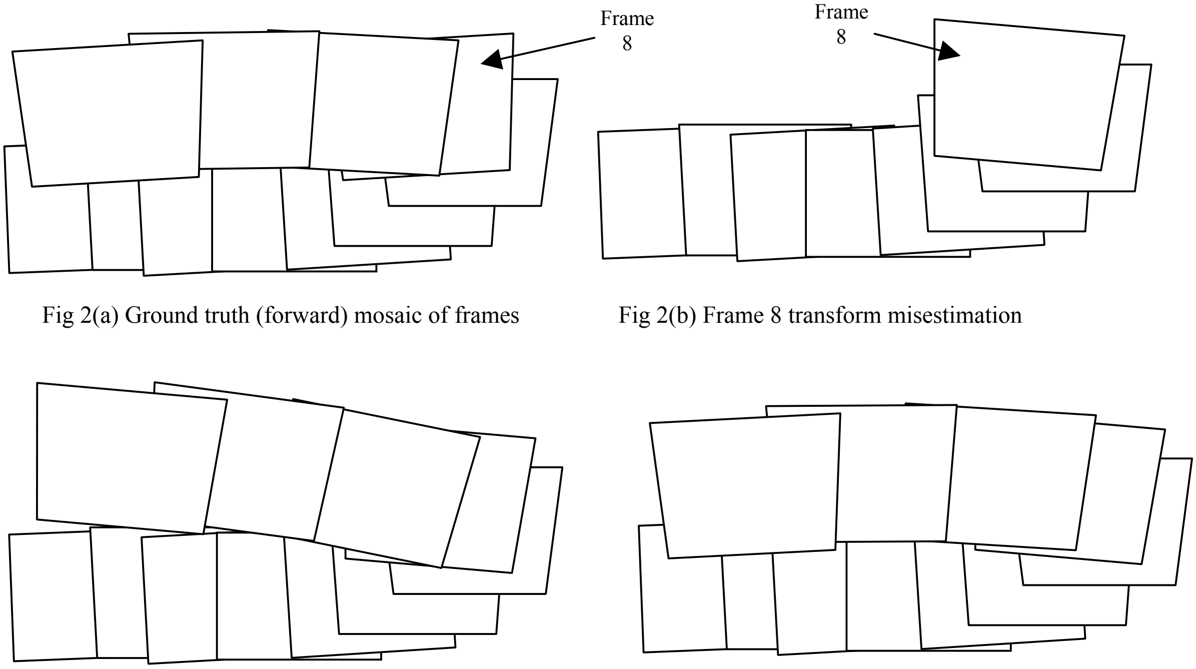

Fig 2(a) Ground truth (forward) mosaic of frames

Fig 2(b) Frame 8 transform misestimation

Fig 2(c) Consequences of the misestimation of frame 8 with correct subsequent frame-to-frame estimation of transforms: The error is magnified.

Fig 2(d) When subsequent frames are estimated with the backward mosaic, frame 8 remains incorrect, but later frames are pulled back into alignment by the effect of earlier (correctly aligned) frames in the backward mosaic.

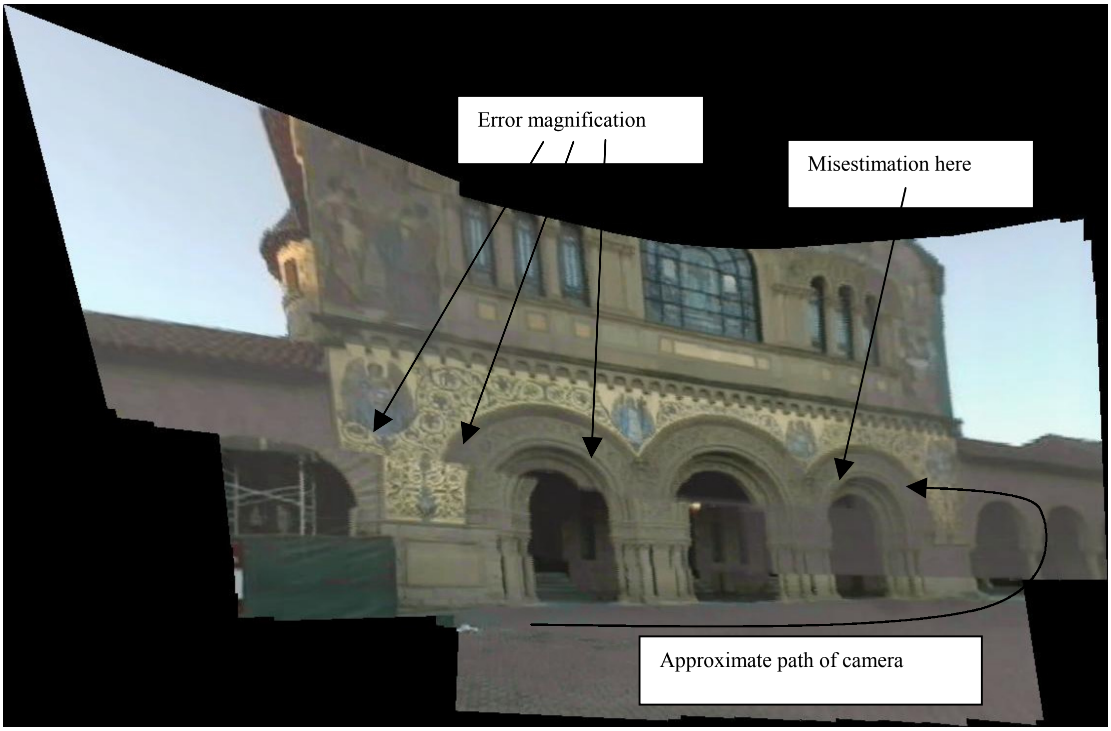

Fig 3. Error magnification on mosaic of Davis' "memchu" sequence following a single small transform misestimation error.

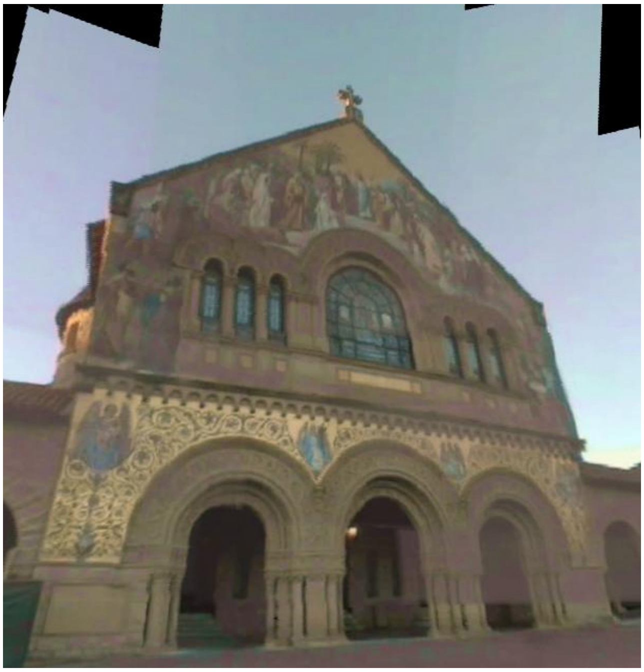

Fig 4: Mosaic generated from Davis' "memchu" sequence. The streaks in the bottom right occur because variation in camera gain is not compensated for. Using *IMP* on a 150-frame sequence has allowed good recovery of image geometry in a single pass with no global correction.

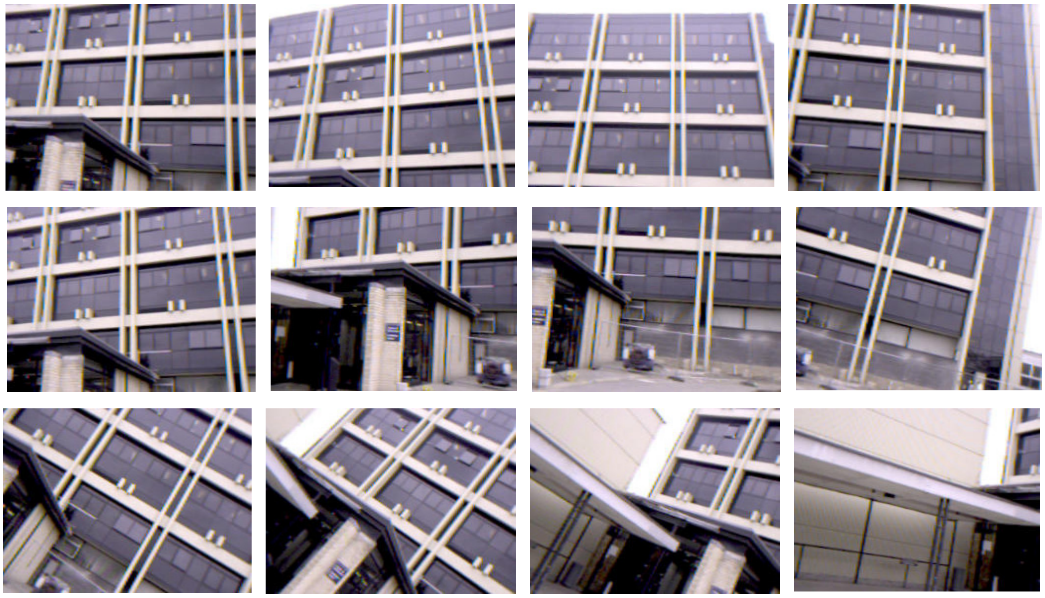

Fig 5: Every tenth frame from an input webcam video showing free camera movement and revisits.

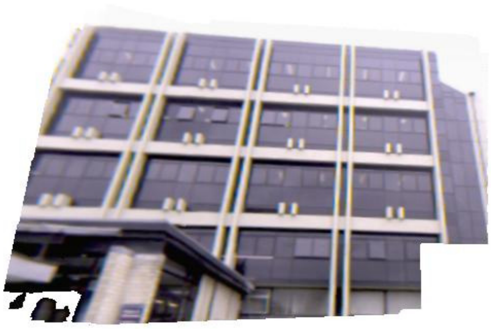

Fig 6: Backward mosaic as displayed to the user. This view is seen about half way through the capture of the video in fig 1.

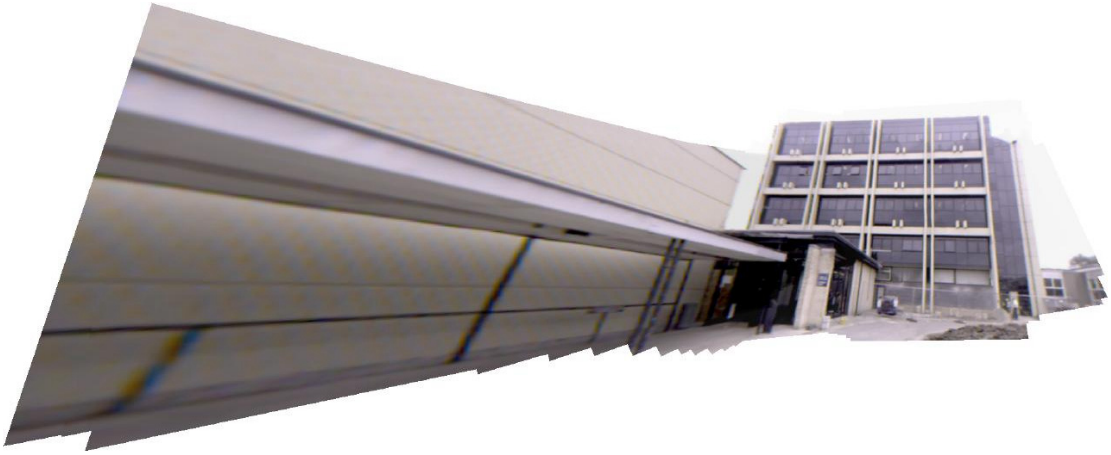

Fig 7: Final forward mosaic from the video sequence in fig 1.

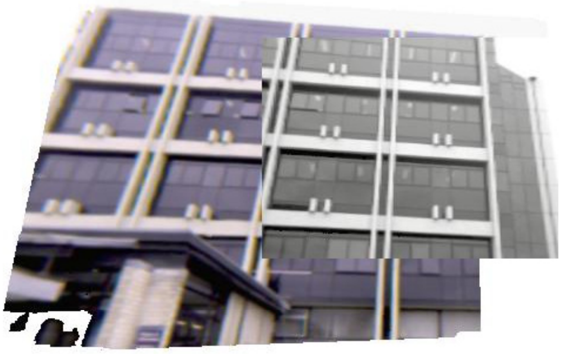

Fig 8: When a good match cannot be found, this is signalled in monochrome.

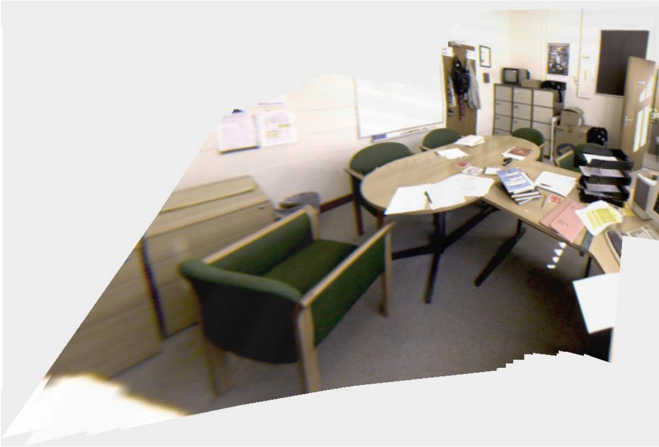

Fig 9: A mosaic generated in real time. The input video included multiple revisits.

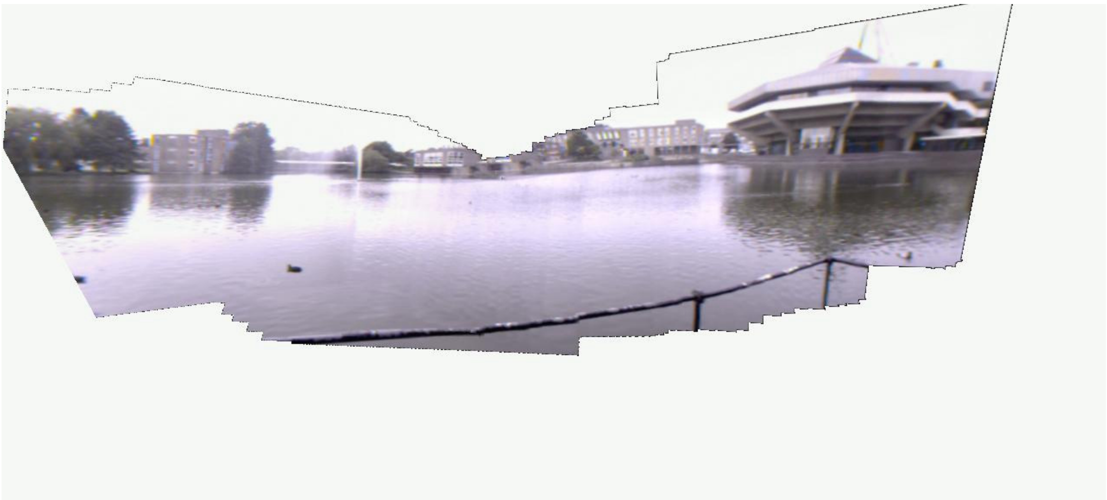

Fig 10: A particularly difficult real-time mosaic.