

Monitoring Human and Vehicle Activities Using Airborne Video

R. Cutler[†], C. Shekhar[†], B. Burns[‡], R. Chellappa[†], R. Bolles[‡], L.S. Davis[†]

[†]University of Maryland, College Park

[‡]SRI International

ABSTRACT

Ongoing work in Activity Monitoring (AM) for the Airborne Video Surveillance (AVS) project is described. The goal for AM is to recognize activities of interest involving humans and vehicles using airborne video. AM consists of three major components: (1) moving object detection, tracking, and classification; (2) image to site-model registration; (3) activity recognition. Detecting and tracking humans and vehicles from airborne video is a challenging problem due to image noise, low GSD, poor contrast, motion parallax, motion blur, and camera jitter. We use frame-to-frame affine-warping stabilization and temporally integrated intensity differences to detect independent motion. Moving objects are initially tracked using nearest-neighbor correspondence, followed by a greedy method that favors long track lengths and assumes locally constant velocity. Object classification is based on object size, velocity, and periodicity of motion. Site-model registration uses GPS information and camera/airplane orientations to provide an initial geolocation with +/- 100m accuracy at an elevation of 1000m. A semi-automatic procedure is utilized to improve the accuracy to +/- 5m. The activity recognition component uses the geolocated tracked objects and the site-model to detect pre-specified activities, such as people entering a forbidden area and a group of vehicles leaving a staging area.

Keywords: Activity recognition, event recognition, motion segmentation, detecting people, periodic motion

1. INTRODUCTION

The extremely high spatiotemporal resolution of a video camera makes it the sensor of choice for surveillance applications. However, use of video data for these purposes has been limited. A video sensor generates a tremendous volume of data, which can quickly overwhelm the operator unless it is preprocessed to retain only those segments that contain significant information, such as the movements of vehicles and people. Methods such as frame differencing are unable to distinguish between independently moving targets and false alarms arising from the motion of the sensor platform or natural phenomena such as trees swaying in the wind. Appearance-based techniques yield poor results on EO video data due to image variations resulting from weather, illumination, and viewpoint changes. Thus until recently, algorithms for video data have been characterized by a lack of robustness, which has precluded their use in a practical application. A further drawback of video sensors has been their limited field of view.

The situation has changed dramatically in the past few years, as a number of important technical challenges related to video-based surveillance have been addressed. First, the availability of reliable and fast image stabilization algorithms have made it possible to compensate accurately for platform motion, and thus eliminate a prime source of false alarms. Second, the extensive use of contextual information made possible by the site model based image exploitation paradigm has reduced the computational load of image understanding algorithms by orders of magnitude. Site models provide accurate topographic and geometric information about areas being monitored, focusing attention on portions of the image relevant to the activity being detected. A further benefit of using site models is a reduction in the number of false alarms, since much less data is processed, and many potential false alarms (due to trees, shadows, etc.) can be predicted and filtered out. And finally, the limited field of view of a video sensor has been expanded significantly into a virtual field of regard using sensor scheduling and video mosaicing algorithms. This work demonstrates a functional surveillance monitoring system using modern technology involving image stabilization, site-model based image exploitation, sensor scheduling/mosaicing, object classification, and activity recognition.

The system consists of three major components: (1) moving object detection, tracking, and classification; (2) image to site-model registration; (3) activity recognition. We use frame-to-frame affine-warping stabilization and temporally integrated intensity differences to detect independent motion. Moving objects are initially tracked using nearest-neighbor correspondence, followed by a greedy method that favors long track lengths and assumes locally constant velocity. Object classification is based on object size, velocity, and periodicity of motion. Site-model registration uses GPS information and camera/airplane orientations to provide an initial geolocation with +/- 100m

accuracy at an elevation of 1000m. A semi-automatic procedure is utilized to improve the accuracy to +/- 5m. The activity recognition component uses the geolocated tracked objects and the site-model to detect scripted activities.

The AM system was implemented on a dual processor Pentium III PC running Windows NT. The SIMD instructions available in the Pentium III (MMX, SSE) were utilized to greatly increase the speed of image processing operations. AM currently processes 360x240 grayscale images at 20 FPS.

2. MOVING OBJECT DETECTION AND TRACKING

Given an image sequence I_t from a moving camera, we segment regions of independent motion. The images I_t are first spatially Gaussian filtered to reduce noise, resulting in I_t^* . The image I_t^* is then stabilized¹ with respect to image $I_{t-\tau}^*$, resulting in $V_{t,t-\tau}$. The images $V_{t,t-\tau}$ and I_t^* are differenced and thresholded to detect regions of motion, resulting in a binary motion image:

$$\mathcal{M}_{t,-\tau} = \begin{cases} 1 & \text{if } |I_t^* - V_{t,t-\tau}| > T_M \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where T_M is a threshold. In order to eliminate false motion at occlusion boundaries (and help filter spurious noise), the motion images $\mathcal{M}_{t,\tau}$ and $\mathcal{M}_{t,-\tau}$ are logically *and*'ed together:

$$\mathcal{M}_t = \mathcal{M}_{t,-\tau} \wedge \mathcal{M}_{t,\tau}. \quad (2)$$

Note that for large values of τ , motion parallax and a violation of the affine motion model will cause false motion in \mathcal{M}_t . In our system, $\tau=500$ ms was typically used. This technique has been extended to integrate multiple images differences, which provides a more robust motion estimate than using only 3 images as in Equation (2).

In many surveillance applications, images are acquired using a camera with automatic gain, shutter, and exposure. In these cases, normalizing the image mean before comparing images I_{t_1} and I_{t_2} will help minimize false motion due to a change in the gain, shutter, or exposure.

A morphological open operation is performed on \mathcal{M}_t (yielding \mathcal{M}_t^*), which reduces motion due to image noise. The connected components for \mathcal{M}_t^* are computed, and small components are eliminated (further reducing image noise). The connected components which are spatially similar (in distance) are then merged, and the merged connected components are added to a list of objects O_t to be tracked. An object has the following attributes: area, centroid, bounding box, velocity, ID number, and age (in frames). Objects in O_t and O_{t+k} , $k > 0$, are corresponded using spatial and temporal coherency.

3. OBJECT CLASSIFICATION

Object classification from airborne video is particularly difficult due to image noise, low GSD, poor contrast, motion parallax, motion blur, and camera jitter. An example image is shown in Figure 2, which shows a person running across a parking lot. Figure 3 shows this person at three different times, which demonstrates the typical image quality input to the classifier. Tracked objects are classified as people, vehicle, or other, using the object's size, ground speed, and periodicity of motion. An object's size is measured in ground area (computed using the image area and the estimated image to site-model registration). Since a detected object may be fragmented into many blobs, the object size is only used to help identify vehicles, but not people. Similarly, the ground speed is also only used to help identify vehicles, but not people, since vehicles can move slowly (like people), but people cannot typically run faster than 20 MPH. The periodicity of the object's motion is the only attribute utilized to classify people. We use an extension of the technique by Cutler and Davis,² which is more robust to changes in object appearance.

3.1. Detecting Periodic Motion

We define the motion of a point $\vec{X}(t)$, at time t , periodic if it repeats itself with a constant period p , i.e.:

$$\vec{X}(t+p) = \vec{X}(t) + \vec{T}(t), \quad (3)$$

where $\vec{T}(t)$ is a translation of the point. When point correspondences are not available, as is the case in our low resolution images of people (see Figure 3), we utilize the periodicity of an object's image similarity.

The output of the motion segmentation and tracking algorithm is a set of foreground objects, each of which has a centroid and size. To detect periodicity for each object, we first align the segmented object (for each frame) using the object's centroid, and resize the objects (using a Mitchell filter³) so that they all have the same dimensions. The scaling is required to account for apparent size change due to change in distance from the object to the camera. Because the object segmentation can be noisy, the object dimensions are estimated using the median of N frames (where N is the number of frames we analyze the object over). The object O_t 's self-similarity is then computed at times t_1 and t_2 . While many image similarity metrics can be defined (e.g., normalized cross-correlation, Hausdroff distance,⁴ color indexing⁵), perhaps the simplest is absolute correlation:

$$S_{t_1, t_2} = \sum_{(x, y) \in B_{t_1}} |O_{t_1}(x, y) - O_{t_2}(x, y)|, \quad (4)$$

where B_{t_1} is the bounding box of object O_{t_1} . In order to account for tracking errors, the minimal S is found by translating over a small search radius r :

$$S'_{t_1, t_2} = \min_{|dx, dy| < r} \sum_{(x, y) \in B_{t_1}} |O_{t_1}(x + dx, y + dy) - O_{t_2}(x, y)|. \quad (5)$$

For periodic motions, S' will also be periodic. For example, Figure 4 shows a plot of S' for all combinations of t_1 and t_2 for a person running (the similarity values have been linearly scaled to the grayscale intensity range $[0, 255]$; dark regions show more similarity). Note that a similarity plot should be symmetric along the main diagonal; however, if substantial image scaling is required, this will not be the case. In addition, there will always be a dark line on the main diagonal (since an object is similar to itself at any given time), and periodic motions will have dark lines (or curves if the period is not constant) parallel to the diagonal.

Let A be the autocorrelation of S' . If S' is periodic, then A will have peaks regularly spaced in a planar lattice M_d , where d is the distance between the lattice points. In our examples, we will consider a 45° rotated square lattice M_d (Figure 1). The peaks \mathcal{P} in A are matched to M_d using the match measure e :

$$B_i = \{\mathcal{P}_i \mid |M_{d,i} - \mathcal{P}_i| \leq \min_{j \neq i} |M_{d,i} - \mathcal{P}_j|, T_D\} \quad (6)$$

$$e(M_d) = \sum_i |M_{d,i} - B_i|, \quad (7)$$

where B_i is the closest peak to the lattice point $M_{d,i}$, and T_D is a distance threshold ($T_D < d/2$). M_d matches \mathcal{P} if all the following are satisfied:

$$\min_{d_1 \leq d \leq d_2} e(M_d) < T_e, \quad (8)$$

$$|B| \geq T_M, \quad (9)$$

where T_e is a match thresholds; $[d_1, d_2]$ is the range of d ; T_M is the minimum number of points in M_d to match. In practice, we let $T_D = 1$, $T_e = 2|M_d|$, $T_M = 0.9|M_d|$. The range $[d_1, d_2]$ determines the possible range of the expected period, with the requirement $0 < d_1 < d_2 < L$, where L is the maximum lag used in computing A . The number

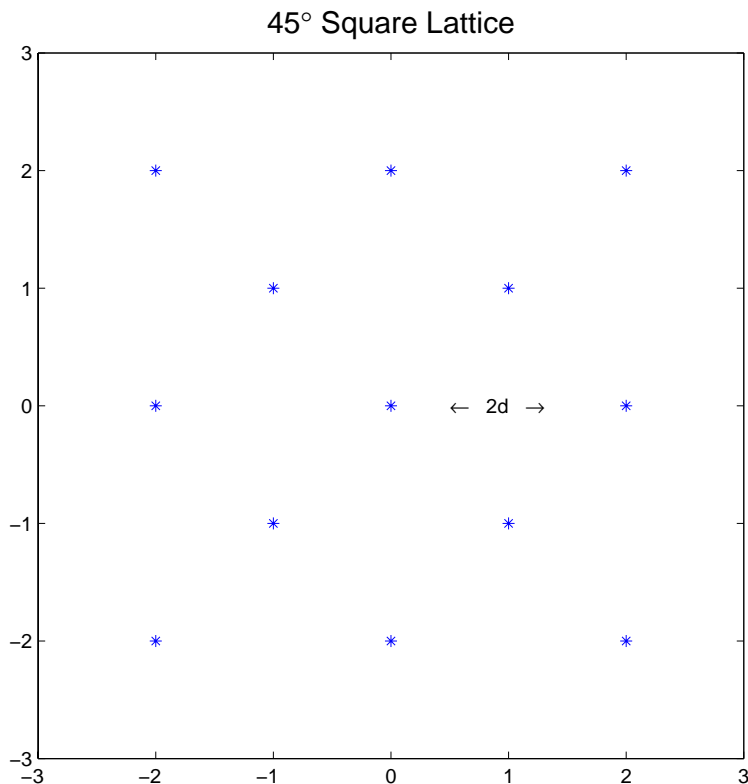


Figure 1. 45° rotated square lattice used to match the peaks of the autocorrelation of S' .

of points in M_d is based on the period of the expected periodicity, and frame-rate of the camera. The period is estimated by $p = 2d\tau$, where τ is the sampling interval (e.g., $\tau = 33$ ms for NTSC video).

Peaks in A are determined by first smoothing A with Gaussian filter G , yielding A^* . $A^*(i, j)$ is a peak if $A^*(i, j)$ is a strict maximum in a local neighborhood with radius N . In our examples, G is a 5×5 filter with $\sigma = 1$, and $N = 5$. Lin et. al⁶ provides an automatic method for determining the optimal size of G . Figure 5 shows the result of the autocorrelation of S' shown in Figure 4, with the detected peaks.

4. IMAGE TO SITE-MODEL REGISTRATION

The Twin Otter aircraft used in the AVS project is equipped with a variety of secondary sensors that provide “metadata” in the form of measurements of the position and orientation of the gimbal as well as the settings of the cameras. Based on these measurements, it is possible to compute a 3x4 “camera” matrix that (in homogeneous coordinates) maps points in the 3D world to points in the image.⁷ The reverse mapping is ill-posed in the general case, but if the scene consists of a flat plane, “geolocation” of points in the image can be accomplished using straightforward linear algebra (see Szeliski⁸). In either case, the camera matrix computed using raw metadata needs to be refined to improve accuracy.

4.1. Registration refinement

The camera matrix computed using the metadata is an approximate one. There are a number of different sources of error, but the ones that affect us the most are the errors in the gimbal azimuth and elevation. Typically, these inaccuracies result in geolocation errors of the order of 100m, which are unacceptably high for activity monitoring. One obvious solution is to use the approximate camera matrix as an initial condition, and refine it using known site features in the image. It would then be necessary to detect, identify and track these features in real time. While this is our ultimate objective, in our present system we employ simplifications to make it more practical given the limited computing power and real-time requirements.



Figure 2. Person running across a parking lot.

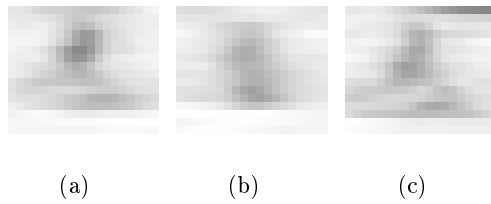


Figure 3. Zoomed images of the person in Figure 2. The person is 12x7 pixels in size.

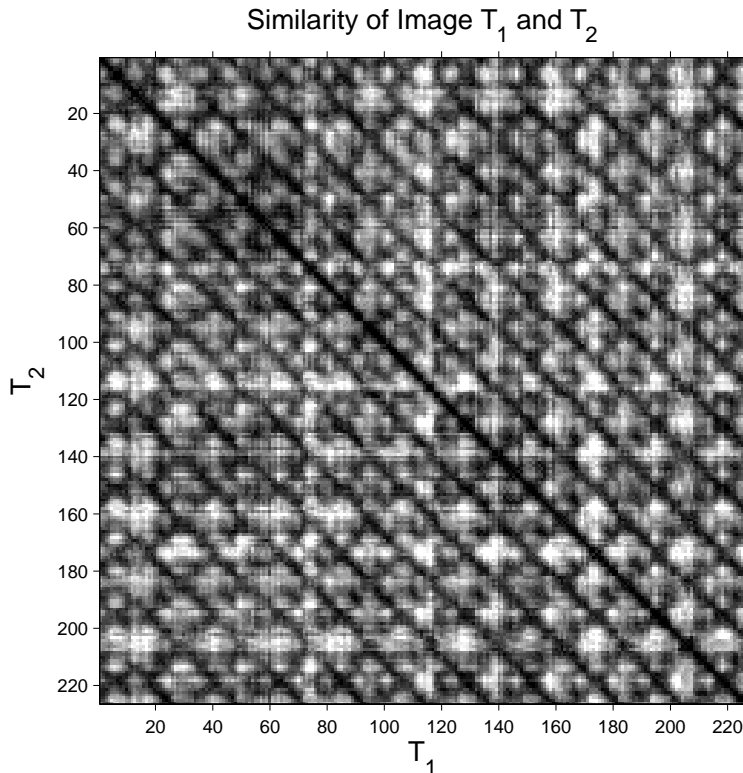


Figure 4. Similarity plot of the running person in Figure 2. Dark pixels correspond to greater similarity.

The first simplification is to assume that the geolocation errors in any given image can be corrected by a pure translational shift. This assumption is valid if the camera is sufficiently far from the scene, and no camera roll is present. Both these conditions are met in the image sequences used in the AVS project. Based on this simplification, we only need a single site feature correspondence to refine the registration.

The second simplification is to permit the operator to interactively select a feature to initialize the semi-automatic registration. This involves using the mouse to click on a point in the live video. It would be impractical for the operator to do this for every frame, so we use the frame stabilization parameters (FSP's) provided by the frame-to-frame stabilization module to track the operator-selected site feature in the live video. When this pseudo-tracked point drifts too far away from the true position in the video image (after a minute or so), the operator re-locates the feature point by clicking on its true location. It is then tracked using FSP's. When a selected feature point leaves the field-of-view, the operator selects a new feature in the site model to track.

5. ACTIVITY RECOGNITION

The goals of the activity recognition portion of this research are (1) to develop techniques for representing activities, such as a person walking to a vehicle, entering it, and driving away and (2) to develop techniques for using these representations to recognize occurrences of them in video data taken from an aircraft. There are several challenging problems to be solved to achieve these goals. First, the representation scheme needs to cover the behaviors that could be interpreted as an occurrence of an activity without including other behaviors. Second, the recognition system needs to work with incomplete and erroneous tracking and classification results. And third, the system needs to be able to represent an interesting collection of behaviors, such as people transferring material from one vehicle to another.

We have developed an initial representation scheme for activities, which we call Activity Templates. Activity templates encode an activity as an augmented finite state machine, which explicitly lists the expected sequences of events that make up the activity. For example, taking-a-vehicle-out-of-a-parking-area is represented by the sequence:

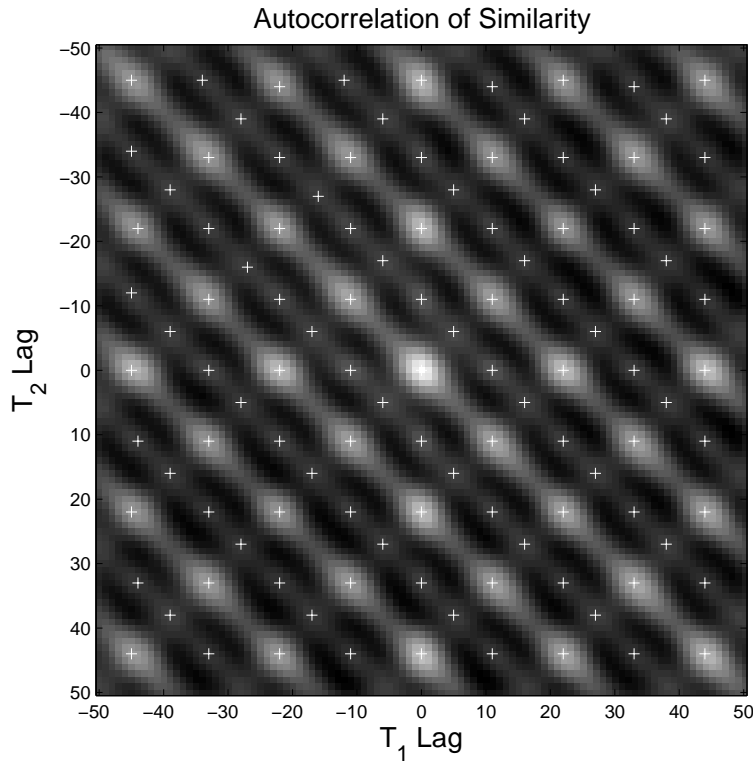


Figure 5. Autocorrelation of S in Figure 4. The peaks (denoted by plus symbols) are used to fit the 45° rotated square lattice in Figure 1. Whiter pixels show higher autocorrelation.

person-approaches-parking-area, person-enters-parking-area, person-moves-inside-parking-area, person-approaches-vehicle, person-enters-vehicle, vehicle-moves-within-parking-area, vehicle-exits-parking-area.

As implied by the example above, activities are stated in terms of a site model that consists of geometric features, such as points, lines, and areas. These features are described in world coordinates versus image coordinates so that they can be recognized from any viewpoint. To do work in world coordinates, however, we rely on video registration and a terrain model to map from image coordinates to world coordinates.

The primitive events that we currently can recognize include approaching/leaving an area (or line or point), entering/exiting an area, and moving inside an area. In addition, the representation scheme supports optional events, alternatives, and time limits.

Figure 6 is an example of an activity template. The states in the finite state machine represent progress in the recognition process. The transitions are taken when the specified events are detected. In addition, camera control, such as scanning along a fence, can be specified along the arcs.

The activity recognition module identifies activities by stepping through the finite state machine, recognizing events and performing the indicated camera control. One limitation of this approach is that alternative interpretations have to be explicitly combined in the finite state machine. We are currently exploring a version of dynamic belief networks that will be able to entertain multiple hypotheses in a more straightforward way.

6. CONCLUSIONS

We have described an airborne surveillance monitoring system that performs robustly under realistic operating conditions. Future enhancements includes (1) improved motion segmentation techniques, such as by Pless et. al,⁹ (2) fully automated registration, and (3) improved activity recognition using belief networks to provide multiple hypotheses and probabilistic reasoning.

Acknowledgments: This work was supported by the DARPA Airborne Video Surveillance project.

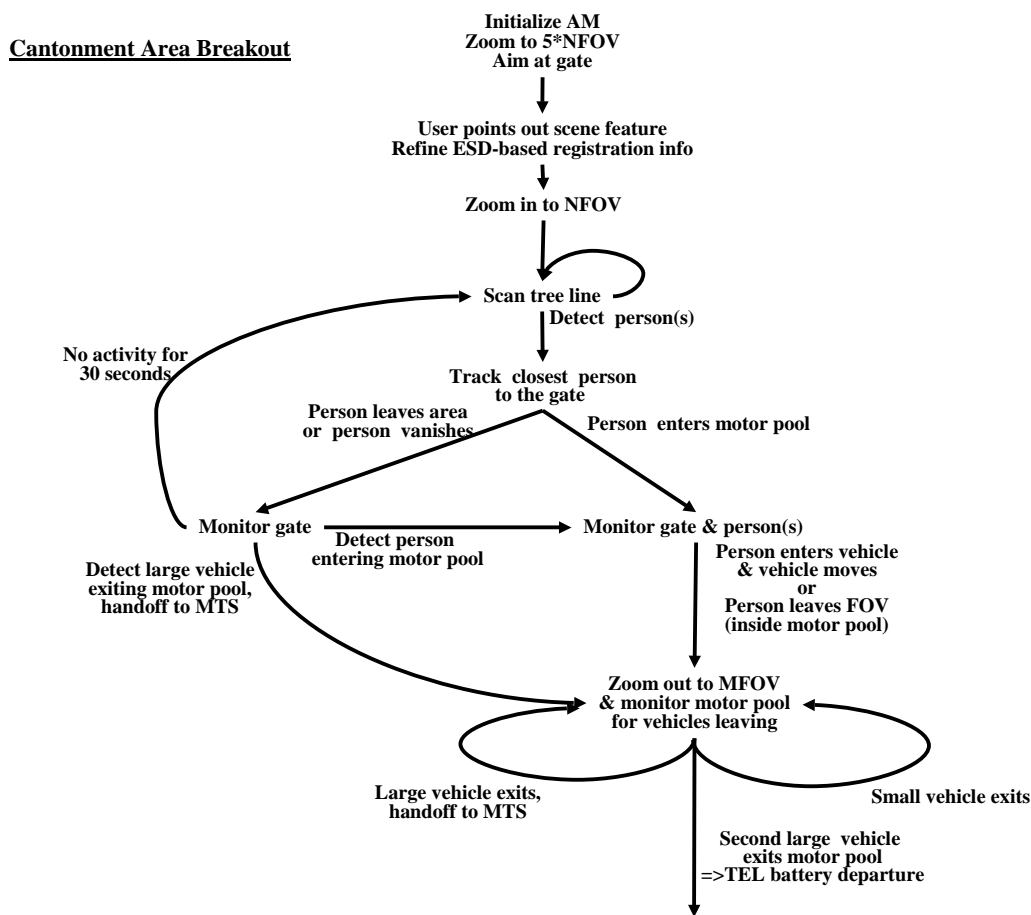


Figure 6. Cantonment Area Breakout vignette

REFERENCES

1. M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, "Real-time scene stabilization and mosaic construction," in *DARPA Image Understanding Workshop*, (Monterrey, CA), Nov. 1994.
2. R. Cutler and L. Davis, "View-based detection and analysis of periodic motion," in *International Conference on Pattern Recognition*, (Brisbane, Australia), August 1998.
3. D. Schumacher, "General filtered image rescaling," in *Graphics Gems III*, D. Kirk, ed., Harcourt Brace Jovanovich, 1992.
4. D. Huttenlocher, G. A. Klanderman, and W. Rucklidge, "Comparing images using the hausdroff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(9), pp. 805–863, 1993.
5. D. H. Ballard and M. J. Swain, "Color indexing," *Int. Journal of Computer Vision* **7-1**, pp. 11–32, 1991.
6. H.-C. Lin, L.-L. Wang, and S.-N. Yang, "Extracting periodicity of a regular texture based on autocorrelation functions," *Pattern Recognition Letters* **18**, pp. 433–443, 1997.
7. L. Wixson, J. Eledath, M. Hansen, R. Mandelbaum, and D. Mishra, "Image alignment for precise camera fixation and aim," in *Proceedings of the Computer Vision and Pattern Recognition*, 1998.
8. R. Szeliski, "Image mosaicing for tele-reality applications," Tech. Rep. 94/2, DEC, 1994.
9. R. Pless, T. Brodsky, and Y. Aloimonos, "Independent motion: the importance of history," in *Proceedings of the Computer Vision and Pattern Recognition*, 1999.