

Aerial Video Surveillance and Exploitation

RAKESH KUMAR, MEMBER, IEEE, HARPREET SAWHNEY, MEMBER, IEEE,
SUPUN SAMARASEKERA, STEVE HSU, MEMBER, IEEE, HAI TAO, MEMBER, IEEE,
YANLIN GUO, MEMBER, IEEE, KEITH HANNA, ARTHUR POPE, RICHARD WILDES, MEMBER, IEEE,
DAVID HIRVONEN, MICHAEL HANSEN, AND PETER BURT, MEMBER, IEEE

Invited Paper

There is growing interest in performing aerial surveillance using video cameras. Compared to traditional framing cameras, videos provide the capability to observe ongoing activity within a scene and to automatically control the camera to track the activity. However, the high data rates and relatively small field of view of videos present new technical challenges that must be overcome before videos can be widely used.

In this paper, we present a framework and details of the key components for real-time, automatic exploitation of aerial video for surveillance applications. The framework involves separating an aerial video into the natural components corresponding to the scene. Three major components of the scene are the static background geometry, moving objects, and appearance of the static and dynamic components of the scene. In order to delineate videos into these scene components, we have developed real time, image-processing techniques for 2-D/3-D frame-to-frame alignment, change detection, camera control, and tracking of independently moving objects in cluttered scenes. The geo-location of video and tracked objects is estimated by registration of the video to controlled reference imagery, elevation maps, and site models. Finally static, dynamic and reprojected mosaics may be constructed for compression, enhanced visualization, and mapping applications.

Keywords—Aerial images, camera control, change detection, computer vision, geo-location, image processing, mosaicing, registration, tracking, video surveillance, visualization.

I. INTRODUCTION

Aerial surveillance has a long history in the military for observing enemy activities and in the commercial world

Manuscript received November 5, 2000; revised May 8, 2001. This work was supported by funding from various Defense Advanced Research Projects Agency (DARPA) programs: Visual Surveillance and Monitoring (VSAM), Warfighter Visualization (TIGER), Image Based Modeling, Aerial Visual Surveillance (AVS), and Battlefield Awareness and Data Dissemination (BADD).

The authors are with the Vision Technologies Laboratory, Sarnoff Corporation, Princeton, NJ 08543 USA (e-mail: rkumar@sarnoff.com; hsawhney@sarnoff.com; ssamarasekera@sarnoff.com; shsu@sarnoff.com; tao@soe.ucsc.edu; yguo@sarnoff.com; khanna@sarnoff.com; apope@sarnoff.com; wildes@cs.yorku.ca; dhirvonen@sarnoff.com; mhansen@sarnoff.com; pburt@sarnoff.com).

Publisher Item Identifier S 0018-9219(01)08435-3.

for monitoring resources such as forests and crops. Similar imaging techniques are used in aerial news gathering and search and rescue. Aerial imagery was used for topographic mapping in 1849 by Colonel Aime Laussedat of the French Army Corps of Engineers [1]. Kites and balloons were used to fly the cameras. In 1913, airplanes were used to obtain photographs for mapping purposes. Aerial images were used extensively in World War I, primarily in reconnaissance and intelligence.

Until recently, aerial surveillance has been performed primarily using film or electronic framing cameras. The objective has been to gather high-resolution still images of an area under surveillance that could later be examined by human or machine analysts to derive information of interest. Currently, there is growing interest in using video cameras for these tasks. Video captures dynamic events that cannot be understood from aerial still images. It enables feedback and triggering of actions based on dynamic events and provides crucial and timely intelligence and understanding that is not otherwise available. Video observations can be used to detect and geo-locate moving objects in real time and to control the camera, for example, to follow detected vehicles or constantly monitor a site. However, video also brings new technical challenges. Video cameras have lower resolution than framing cameras. In order to get the resolution required to identify objects on the ground, it is generally necessary to use a telephoto lens, with a narrow field of view. This leads to the most serious shortcoming of video in surveillance—it provides only a “soda straw” view of the scene. The camera must then be scanned to cover extended regions of interest. An observer watching this video must pay constant attention, as objects of interest move rapidly in and out of the camera field of view. The video also lacks a larger visual context—the observer has difficulty perceiving the relative locations of objects seen at one point in time to objects seen moments before. In addition, geodetic coordinates for objects of interest seen in the video are not available.

Further challenges of video relate to control and storage. A camera operator can have difficulty manually controlling the camera to scan a scene or to hold an object of interest in view because of the soda straw view of the world provided by video. Video contains much more data than traditional surveillance imagery, so it is expensive to store. Once stored in a database, surveillance video is difficult and tedious to search during subsequent analysis.

Before video can be generally employed in aerial surveillance, new video technologies must be developed that make it much easier for human operators to use and interpret video data. Technologies are needed to automatically control the camera and to detect and geo-locate objects of interest. New methods are needed to annotate and present video imagery to humans to provide an immediate, in-depth understanding of the observed scene. Technologies are also needed to compress and store surveillance video and to give users easy access to archived video.

In order to serve a variety of different needs for surveillance applications, it is important to provide an underlying framework for spatio-temporal aerial video analysis. In the past decade or so, we have developed such a framework based on image alignment with progressively complex models of motion and scene structure. The framework involves delineation of video imagery into components that correspond to the static scene geometry, dynamic objects in the scene, and the appearance of both the static and dynamic parts of the scene. Progressive complexity in this framework provides us a handle on the model selection problem since always applying the most complex model (say, a 3-D alignment model) may lead to unstable and unpredictable results. In essence, such instabilities arise as the employed model results in over fitting the data at hand. For example, 3-D estimation may produce unpredictable results when the scene may be largely flat. Furthermore, the alignment framework also includes situating the video components in a geo-referenced coordinate system within a reference imagery and model database. By aligning the static scene components in video frames to a reference database, static and dynamic objects, entities and locations in the video can be geo-referenced and annotated with respect to the information in the reference database.

The video analysis framework has been used to develop a number of key capabilities, some of which have been put together in a system for aerial video surveillance. These key capabilities include:

- frame-to-frame alignment and decomposition of video frames into motion (foreground/background) layers;
- mosaicing static background layers to form panoramas as compact representations of the static scene;
- detecting and tracking independently moving objects (foreground layers) in the presence of 2-D/3-D backgrounds, occlusion and clutter;
- geo-locating the video and tracked objects by registering it to controlled reference imagery, digital terrain maps and 3-D site models;
- enhanced visualization of the video by re-projecting and merging it with reference imagery, terrain and/or maps to provide a larger context.

Over the past several years, the Defense Advanced Research Projects Agency (DARPA) and other government agencies have sponsored a series of research programs aimed at developing these new technologies. The Vision Technologies Laboratory at Sarnoff Corporation has participated in many of these programs and has contributed to almost all aspects of the technology, from automated search and detection to compression, display, storage, and access.

II. SYSTEM OVERVIEW

In this section, we provide a brief overview of an integrated aerial video surveillance (AVS) system. In the following sections, we examine each component of this system in more detail. For this general overview, we assume the AVS system is intended for military surveillance from an unmanned aerial vehicle (UAV). This application requires the greatest degree of automation and performs the widest range of tasks. For the most part, systems for other applications will use a subset of the components described here. The UAV system is shown in Fig. 1. The system includes sensors and processing components on board the aircraft and additional processing and displays at an operator control station on the ground.

A. Sensors: Cameras, Gimbals and Telemetry

Video surveillance is typically performed with one or two video cameras that are mounted within gimbals on board an aerial platform. The cameras may be standard visible light (electrooptical) or infrared (IR). Gimbals allow the camera to actively scan a scene. The gimbals also provide mechanical stabilization to isolate the camera from aircraft vibration. Telemetry meta-data associated with video are physical measurements of the camera's pose and calibration, which include location and orientation of the sensor in a world coordinate system and intrinsic camera parameters such as focal length and optical center.

B. Front-End Processing

We divide image processing performed on board the aircraft into two stages: front-end processing and scene analysis. Front-end processing includes "signal level" operations that are applied to the source video to enhance its quality and to isolate signal components of interest. Examples of such operations are electronic stabilization, noise and clutter reduction, mosaic construction, image fusion, motion estimation, and the computation of "attribute" images such as change and texture energy.

C. Scene Analysis

Scene analysis includes operations that interpret the source video in terms of objects and activities in the scene. Moving objects are detected and tracked over the cluttered scene. Terrain is recovered for the purposes of moving object detection in the presence of 3-D parallax and also for aerial mapping purposes.

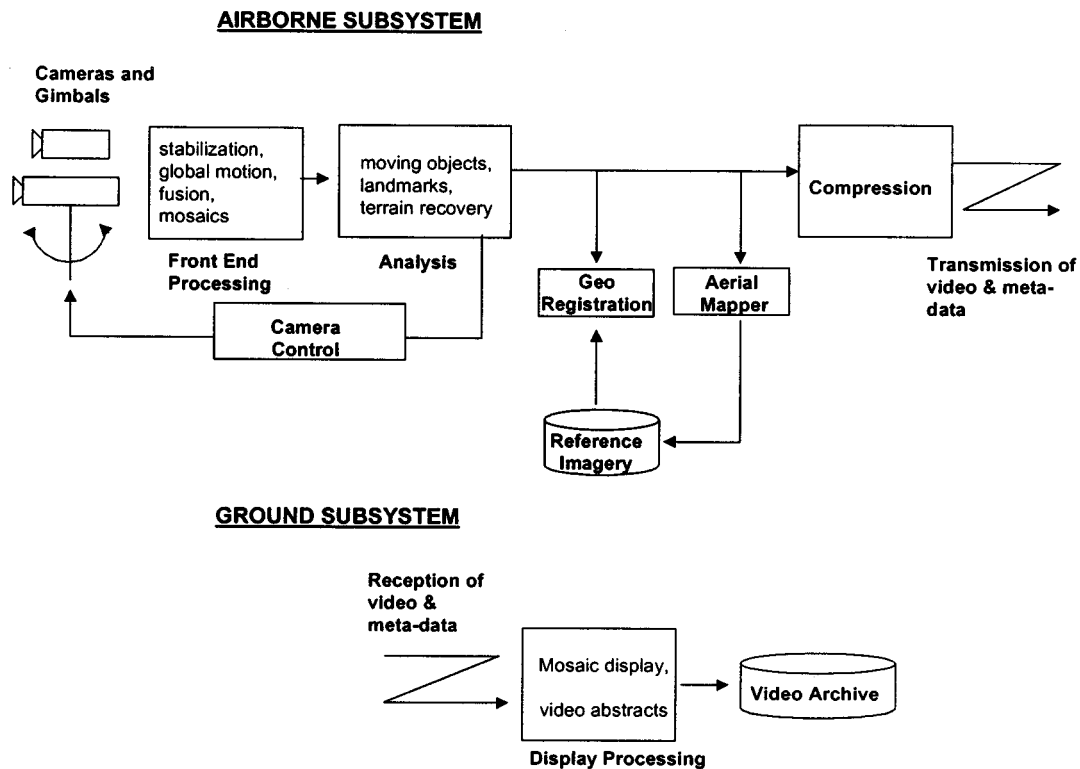


Fig. 1. Components of an aerial video surveillance system for use with an unmanned aerial vehicle. Surveillance systems for other military and civilian applications would generally use a subset of these components.

D. Camera Control

Scene information obtained in the analysis stage provides a basis for automatic camera control. Task-oriented control modes include fixating points on the ground, tracking moving objects, and scanning regions of interest. Other functions include automated control of focus and control of the camera exposure to obtain better images of selected objects.

E. Geo-Location

The AVS system must also determine the geodetic coordinates of objects within the camera's field of view. Rough geo-locations of points on the ground can be estimated from aircraft coordinates and camera orientation. However, as altitude increases, small errors in measured camera orientation result in larger geo-location errors. More precise geo-locations can be estimated by aligning video frames to calibrated reference images. Further, image based geo-location allows video to be precisely localized in the presence of telemetry dropouts. Geo-locations are used in turn for precision targeting, target handoff between aircraft, updating imagery in geo-spatial databases, and indexing video databases.

F. Aerial Mapping

A common objective of aerial surveillance is that of obtaining images of extended regions of the landscape. Such images are constructed as mosaics from the source video as the camera sweeps over the scene. These mosaics are ortho-

rectified (projected to nadir view) and registered to map coordinates. Subsequently, they can be used to update imagery in a geo-spatial database.

G. Compression and Transmission

Video is transmitted to the ground station where the operator views it. This transmission often must occur over a low-bandwidth channel, so significant image compression is required. The results of scene analysis on board the aircraft provide a basis for high-performance object-based video compression such as MPEG4, in which the stationary background is represented as a mosaic and foreground moving objects are represented as video "chips."

H. Display

Additional video processing is performed at the display to format video information so it is easy for a human to interpret. Video is stabilized and accumulated into an extended mosaic. Target tracks and landmark names are overlaid on the mosaic. In addition, this "mosaic based display" allows the user to scroll and zoom into regions of interest, independent of the camera scan.

I. Archiving

Finally, surveillance video is stored for future reference and analysis. Video abstracts are stored with the video to support content-based access. These include ortho-mosaics and activity records.

Although our concern in this paper is with image processing for aerial surveillance, it is appropriate to begin with a brief description of the camera, gimbal, and pose sensors, since these determine characteristics of the video and associated meta-data that will be processed.

Both IR and visible cameras are used in AVS. The image formats that are used cover a wide range, but commercial standards are typical: e.g., 720×480 pixel resolution, 30 frames per second for NTSC based systems. Interlace cameras with 60 fields per second are often used but these introduce aliasing artifacts—progressive scan cameras are to be preferred. The camera optical system is chosen to resolve objects of interest on the ground from the intended operational altitude of the aircraft. In practice, the camera field of view is often quite narrow; 0.3° to 10° is common.

Camera gimbals provide mechanical stabilization, isolating the camera from vibration of the aircraft. Gimbals differ in sophistication and quality. A high performance gimbal for military surveillance may provide stabilization to $10 \mu\text{rad}$. This corresponds to 0.4 horizontal pixels for a 720×480 camera with a 1° field of view. However, such gimbals are extremely expensive (typically \$300 000–\$500 000). Less expensive (\$30 000–50 000) gimbals typically provide stabilization to $100 \mu\text{rad}$, corresponding to 4.1 horizontal pixels. There is a lower limit to the field of view that can be supported by a given camera and gimbals system. This limit is set by motion blur. Motion blur occurs when the image moves by more than the width of a pixel; during the time the camera integrates light for a given video frame. For example, suppose 0.01-s integration time is required for a given camera and camera stability from frame to frame is $100 \mu\text{rad}$. If this camera has 720 horizontal pixels, then motion blur would occur for an optic system that provides less than a 1.23° field of view.

An important adjunct to video information is time-synchronized meta-data or engineering support data (ESD) containing physical measurements of camera pose and calibration. Translation may be derived from altimeters, Global Positioning System (GPS), and other radio navigation signals. Rotation is derived from a combination of aircraft and gimbal orientations. Aircraft orientation may be derived from an inertial navigation system (INS), gravity sensor, magnetic compass, radio navigation signals, or multiple GPS antennas. Both gimbal orientation and camera calibration (e.g., focal length of a variable zoom lens) can be derived from control signals. All such measurements, which relate the image pixels to rays in 3-D world coordinates, are referred to as telemetry. Position measurements using survey-grade differential GPS with on-the-fly carrier ambiguity resolution can achieve 5-cm accuracy in practice. High-quality INS can achieve $500 \mu\text{rad}$ orientation accuracy. However, instrumentation currently deployed on today's operational UAVs are commonly one to three orders of magnitude worse. Synchronization of meta-data with video is also a critical issue since the pose of a surveillance camera may be changing rapidly.

Our decomposition of processing on board the aircraft into front end and analysis stages is based on the nature of the computations themselves and hence on the types of processing architectures best suited to perform them. Front-end processes are those that are performed in the “signal domain” as filtering and resampling operations on the video frames. These operations are performed uniformly over extended image regions and lend themselves to a flow-through, pipelined computing architecture. Front-end processes include signal enhancement, electronic stabilization, mosaic construction energy, and detection of moving objects. Front-end processing also generates local attribute images such as motion flow and texture.

A. Computational Framework

Sarnoff has developed a special purpose image processing architecture for front end processing [2]. This features multiresolution and parallel pipeline computation flow. It is tailored to perform filtering and resampling operations very efficiently and to adapt from frame to frame to scene activity and changing analysis requirements. Within a given image frame, the filters are applied uniformly over defined regions of interest.

Basic sequences of front-end operations that serve multiple AVS functions are as follows.

- 1) Pyramid Transform. Convert each source video frame $I(t)$ into a multiresolution, pyramid representation. This includes low pass or Gaussian and bandpass or Laplacian components.
- 2) Global Motion Estimation. Estimate global frame-to-frame image motion $d(t)$ that is due, for example, to camera pan, zoom and general 3-D motion of the sensor.
- 3) Video Mosaic. Combine images within the temporal window into a mosaic.
- 4) Electronic Stabilization. Warp image frames to remove unwanted global motion.
- 5) Detection of scene changes using 2-D and 3-D techniques to compensate for image sensor motion.

B. Pyramid Transform

In order to achieve the best performance at the front-end signal processing stage of aerial video surveillance, it is essential to provide means for controlling the scale of spatial and temporal operations so they can be matched to salient image structure in the source video. This can be achieved in the spatial domain by representing the source video at multiple scales. Operators of fixed scale can then be matched to the appropriate image scale.

The first front-end processing step is to perform a pyramid transform on each source video frame [3]. A pyramid is a sequence of images of decreasing resolution obtained by repeatedly convolving an initial image with a set of filters “ W .” In the basic Gaussian pyramid, “ W ” acts as a low-pass filter. Successive application of “ W ” implies that the band limit is

reduced in octave steps from image to image in the pyramid sequence. As the band limit is reduced, the sample density may also be decreased by a factor of two.

A Laplacian pyramid is a sequence of images obtained by computing the differences between successive levels of the Gaussian pyramid. In effect, this is tantamount to filtering the original image by a sequence of bandpass filters. Laplacian pyramids highlight edge information in images and are often used for matching images of the scene acquired under different illumination conditions.

However, Gaussian and Laplacian pyramids are isotropic in nature and do not provide any information on the orientation of features present in the image at different scales. Orientation energy pyramids are used to represent an image at multiple scales and orientations. Many different schemes have been proposed in the literature to represent and compute orientation in images. One popular scheme to compute orientations was proposed by Freeman and Adelson [4]. They propose computation of orientation information via application of a bank of filters that are tuned for spatial orientation and scale. In particular, the filtering is implemented in terms of second derivative Gaussian filters, at different orientations, and their Hilbert transforms [4]. The filters are taken in quadrature to eliminate phase variation by producing a measure of local energy, within an orientation band. This filtering is typically applied at a set of four orientations—vertical, horizontal, and two diagonals—to yield a corresponding set of “oriented energy images.” Further, the entire representation is defined over a Gaussian pyramid [5] to support multiresolution analysis. Orientation pyramids are often used to solve difficult alignment problems where there is a great deal of appearance change between two images. An example of an orientation energy representation of an image can be seen in Fig. 17(c). Significantly, image energy measures can be extended to account explicitly for spatio-temporal orientation by defining the filtering over $x - y - t$ video volumes for added representational power to support the analysis of time varying imagery [6].

C. Global Motion Estimation

The displacement of pixels between subsequent frames in a video sequence may occur because of the motion of the video sensor, independent motion of objects in the scene, motion of the source of illumination, and other factors. At the global motion estimation step, the displacement of pixels due to the motion of the sensor is computed.

Let $\vec{d}(x, y, t_1, t_2)$ be a displacement field describing the motion between frames $I(x, y, t_1)$ and $I(x, y, t_2)$. Let $\hat{I}(x, y, t_1, t_2)$ be image $I(x, y, t_1)$ warped into alignment with image $I(x, y, t_2)$ based on the following displacement field:

$$\hat{I}(x, y, t_1, t_2) = I((x, y) - \vec{d}(x, y, t_1, t_2), t_1). \quad (1)$$

The displacement field due to motion of the sensor can be modeled by a set of global and local parameters. This modeling is simple for simple camera motions such as panning, zooming or for general motion with respect to a distant

scene and is more complicated for arbitrary 3-D motions and scenes. A parametric model, such as an affine, quadratic, or projective transform [7]–[9] may represent pixel motion due to camera pan or zoom or translation relative to a distant surface. The displacement field modeled as a quadratic function is shown in (2)

$$\begin{aligned} d_x^q(x, y, t_1, t_2) &= a_1 + a_2x + a_3y + a_7x^2 + a_8xy \\ d_y^q(x, y, t_1, t_2) &= a_4 + a_5x + a_6y + a_7xy + a_8y^2. \end{aligned} \quad (2)$$

Note that for discrete pairs of views with significant camera rotation, a full eight-parameter projective transformation is required to align a planar surface or the distant scene. However, for closely related views such as those obtained from a video sequence, the above quadratic transformation is a good approximation to the motion model and is more stable to compute. Furthermore, as the center of projection recedes from the distant surface of interest, an affine transform may suffice.

For general 3-D scenes, where the camera is not distant from the scene, the quadratic model does not account for pixel motion due to parallax. In this case, the model for pixel motion is also a function of the distance of the scene point from the camera (i.e., its depth). Traditionally, in dynamic image analysis the 3-D displacement field has been parameterized in terms of rotational and translation motion and depth fields [10]. However, aligning the images using this parameterization requires knowledge of the camera interior orientation parameters such as focal length and optical center. In many applications, this information is typically not available. In [11]–[13], the alternate parameterization described does not require this knowledge. The parameterization is based on the image motion of a 3-D plane and the residual parallax field [14].

In [11], it was shown that, given two views (under perspective projection) of a scene (possibly from two distinct uncalibrated cameras), if the image motion corresponding to an arbitrary planar surface is compensated (by applying an appropriate 2-D parametric warping transformation to one of the images) then the residual parallax displacement field on the reference image plane is an epipolar field. That is

$$\begin{aligned} d_x &= d_x^q + d_x^r \\ d_y &= d_y^q + d_y^r. \end{aligned} \quad (3)$$

The total motion vector (d_x, d_y) of a point can be written as the sum of the motion vector due to the planar surface modeled as a quadratic (d_x^q, d_y^q) as represented in (2) and the residual parallax motion vector (d_x^r, d_y^r) . The residual vector can be represented as

$$\begin{aligned} d_x^r &= \gamma * (fT_x - xT_z) \\ d_y^r &= \gamma * (fT_y - yT_z) \\ \gamma &= \frac{H}{(Z * T_p)} \end{aligned} \quad (4)$$

where $\vec{T} = (T_x, T_y, T_z)$ is the translation, γ is the parallax at a point, H is the perpendicular distance of the point of

interest from the plane, Z is its depth, T_p is the perpendicular distance from the center of the first camera to the plane, and f is the focal length. At each point in the image, the parallax varies directly with the height of the corresponding 3-D point from the reference surface and inversely with the depth of the point.

Traditionally, motion parameter estimation is done in a two-step process. First, correspondences are established between frames [15], [16] and then a structure from motion algorithm is applied to solve for the motion parameters [17], [18]. The robustness of the parameter estimation obtained by this two-step method is very much dependent on the accuracy of the correspondences established in the first step. As an alternative, [9] and [19] propose direct methods for estimating the motion parameters and the correspondences between images simultaneously. This approach has proved to be very stable for computing motion parameters between consecutive frames in a video sequence.

In order to align two images (an “inspection” image and a “reference” image), [9] describes a family of hierarchical direct registration techniques with different image motion models. This technique first constructs a pyramid from each of the two input images and then estimates the motion parameters in a coarse-to-fine manner. Within each pyramid level, the sum of squared difference (SSD) measure integrated over regions of interest (which is initially the entire image region) is used as a match measure

$$E(\{A\}) = \sum_x (I(x, y, t_1) - I((x, y) - \vec{d}(x, y, t_1, t_2)))^2 \quad (5)$$

where “ A ” is the set of motion parameters, which need to be estimated. These parameters may be global 2-D parameters such as the quadratic parameterization described in (2) or 3-D quasi-parametric transform such as the plane + parallax transform described in (3) and (4).

The general framework for all the direct algorithms is the same. The objective function in (5) is minimized to find the parameter values that yield a best match between the pair of images. Levenberg-Marquardt minimization [20] is applied to the objective function to estimate the unknown motion parameters and the resulting displacement field. Starting with some initial values (typically zero), the hierarchical estimation algorithm iteratively estimates the parameters in order to minimize the SSD error at a coarse resolution, then successively at finer resolutions. After each step of the iteration, the transformation based on the current set of parameters is applied to the inspection images, in order to reduce the residual displacement between the images. The reference and inspection images are registered so that the desired image region is aligned. The above estimation technique is a least-squares-based approach and, hence, sensitive to outliers. However, as reported in [9], doing the least-squares estimation over a pyramid minimizes this sensitivity. The pyramid-based approach locks on to the dominant image motion in the scene. To further improve rejection of noise and unmatched structure, [21] developed robust versions of

the above least-squares algorithms using robust statistics. In practice, image alignment is often computed between band-pass Laplacian pyramid levels to reduce effects of illumination variation, or between images that have been otherwise preprocessed to enhance salient pattern structure (e.g., oriented energy pyramids).

D. Video Mosaics

A further function of front end processing is mosaic construction [1]. Images are accumulated into the mosaic as the camera pans, effectively extending the field of view of the camera. We will define several types of image mosaics at successive stages of the AVS processing stream. The “image domain” mosaics formed in the front-end stage are the simplest and are defined in a 2-D image coordinate system. Later mosaics will be defined in the world coordinates and will have 3-D structure.

There are two basic steps in constructing image domain mosaics: alignment and merging [22]–[25]. Construction of a 2-D mosaic requires computation of alignment parameters that relate all of the images in the collection to a common world (or mosaic) coordinate system. For image collections corresponding to video sequences, the computation of alignment parameters may be structured in one of the following ways. 1) Successive images are aligned, the parameters cascaded to determine the alignment parameters between any frame and the mosaic coordinate system. 2) Each image is aligned directly to the current composite mosaic image using the mosaic coordinate system as the reference. 3) The current composite mosaic image is aligned to the new image, using the new image as the reference. The result of any of these processes is a set of transformations linking (directly or indirectly) coordinate systems of all of the relevant images. These transformations are used to warp each video frame to the mosaic coordinate system. These warped images are then combined to form a mosaic. In order to avoid seams between images in the mosaic, the warped video frames are merged in the Laplacian pyramid domain [26]. This blends images along the seams such that image features are merged over a distance comparable to their size—large features over a long distance and small features over a small distance. An example of a mosaic constructed using the above procedure is shown in Fig. 2. In this case, the images form a sequence in which the camera (mounted on an aerial vehicle) moves over an extended segment of terrain while also rotating and zooming. The input frames of the video sequence are of size 360×240 pixels. The output mosaic is of size 855×741 pixels and is constructed using 11 key frames automatically picked from a long sequence.

E. Electronic Stabilization

Each of the front-end processes described thus far have used the current frame to define the space/time coordinate system; all frames used in processing at time t are in the set $S(t)$ that are aligned to the $I(t)$. This ensures the processed data is in registration to current physical reality. However, the source video often contains unwanted components of global image motion that are due to camera vibration or instability

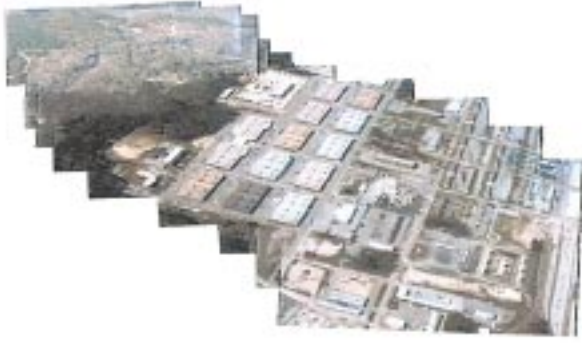


Fig. 2. Image domain mosaic from video over Camp Lejeune, NC.

in camera control. This motion can be removed through electronic stabilization, as a final front-end processing step [22]. The observed frame-to-frame offsets, $d(t)$, may be assumed to combine a desired component, $d_d(t)$, due, for example, to intended camera pan and zoom and an undesired component, $d_u(t)$, due to camera vibration and control instability. For electronic stabilization, each output frame needs to be warped by $d_u(t)$ [equal to $d(t) - d_d(t)$] to compensate for this motion component.

The desired component of motion combines intended camera pan and zoom and motion relative to the scene as the aircraft flies. The components due to pan and zoom may be known from the camera controller. The component due to aircraft motion depends on the distance to objects in the scene so they will often not be known. Absent other information, $d(t)$ is decomposed into high and low temporal frequency components and stabilization is used to damp high frequency components. Alternatively, the desired component of camera motion may be modeled as a sequence of linear segments separated by break points. The electronic stabilization unit attempts to estimate these break points and the linear motions as a continuous causal process as frames are received.

F. Change Detection and Moving Object Identification

Video cameras image the scene at the rate of 30 frames per second. As a result of this rapid imaging, it is possible to detect moving objects from the static background and identify other changes between subsequent video frames even when the camera sensor is itself moving. The motion of the pixels due to sensor motion must be compensated to detect real changes in the scene and identify moving objects. As noted in Section IV-C, pixel displacements due to sensor motion may be modeled by 2-D parametric motion for distant scenes or panning cameras. When the camera is close to the imaged scene, the pixel displacements of the background must be modeled by a quasi-parametric transform such as the plane + parallax model for 3-D motion discussed earlier. We discuss change detection for these two cases in the next two sections.

1) *Estimation of Change Energy Using 2-D Motion Compensation:* The procedure for the computation of change energy is shown in Fig. 3 [27], [28]. This provides a basis for moving object detection. The first four processing steps generate a space/time difference video sequence: 1) the source

video frames $I(t)$ are transformed spatially into their band-pass Laplacian pyramid representation $I_L(k, t)$, where k represents different pyramid levels; 2) global motion parameters $d(t - n, t)$ between frames “ $t - n$ ” and “ t ” are estimated from $I_L(k, t)$; 3) a temporal window $S(t)$ is formed by warping frames into alignment with $I_L(k, t)$; and 4) a temporal bandpass filter is applied to $S(t)$ to form each difference video frame $D_L(k, \tau, t)$. The temporal filter has the effect of forming the difference between the current frame, $I_L(k, t)$ and a frame at some time τ earlier in the source sequence [and within $S(t)$]. These steps tend to eliminate the stationary background and highlight moving image features of spatial scale k (the Laplacian pyramid level) that move a significant distance relative to this scale over the time τ .

The space/time difference sequence is converted to change energy through four additional steps: 1) the samples of $D_L(k, \tau, t)$ are squared; 2) squared values are integrated locally through Gaussian pyramid construction to level l ; 3) $I_L(k, t)$ is similarly squared and integrated to Gaussian level l ; and 4) normalized change energy $E(k, l, \tau, t)$ is formed as the ratio of these quantities at each sample position. Specifically, let $G_l[I(t)]$ be the Gaussian pyramid constructed from image $I(t)$ to level l . The change energy is given by

$$E(x, k, l, \tau, t) = \frac{G_l[D_L(x, k, \tau, t)]^2}{G_l[I_L(x, k, t)]^2}. \quad (6)$$

Scale parameters k , l , and τ determine space/time characteristics of this change energy image and allow the process to be tuned to targets of interest in the scene. A related change measure has been described in [29]. Analysis of spatio-temporal orientation provides another refinement of these ideas for change detection with robustness to image clutter [30]. An example change energy image is shown in Fig. 4.

Change energy computed between two frames can give rise to two change regions for a single moving object: one at the object’s location in the current frame and one at the object’s location in the previous frame [31]. A single change image does not show which region corresponds to the object’s location in the current frame. This ambiguity can be resolved by computing change images from three frames. Given the current image $I(t)$ and previous image at time τ_1 and next image at time τ_2 , change energy images $E(x, k, l, \tau_1, t)$ and $E(x, k, l, \tau_2, t)$ are constructed as described above, then multiplied together [32]. Forming the product of this pair of change energy images has the effect of suppressing regions where there is energy in only one. Since both images will have energy at the location of the object in the current image, it is only this location that remains highlighted.

2) *Change Detection in 3-D:* As the aircraft flies, stationary objects in the camera field of view that stand above the ground, such as treetops, appear to move relative to the ground itself. As long as this parallax motion is sufficiently small, it can be ignored and moving object detection can be based on 2-D analysis, as outlined above. However, parallax becomes significant when Aerial Video Surveillance is conducted from a low or fast aircraft, or the motion of targets of

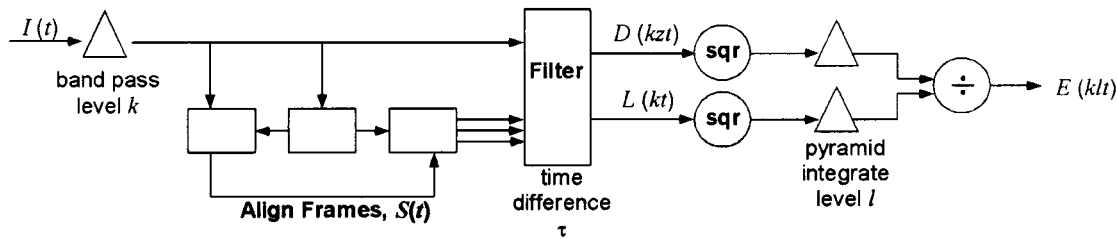


Fig. 3. Computation of change energy. Scale parameters k , l , and τ tune the process to objects and motions of interest in the scene.

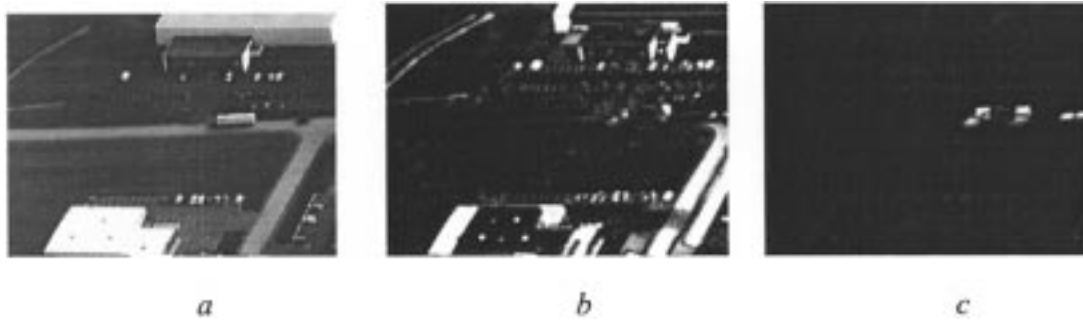


Fig. 4. Example change energy image. (a) Source frame, (b) change energy without alignment, and (c) change energy with alignment, moving truck appears as bright pixels in the change energy image.

interest is small. In these cases, the analysis has to be generalized to three dimensions.

A number of approaches to the problem have either employed only the epipolar constraints or have assumed that correspondences and/or image flow are available or can be reliably computed. Adiv [7] assumed the availability of optical flow and used the flow to group regions on the basis of the rigidity constraint over two frames. Irani and Anandan [33] presented constraints on image parallax arising from constancy of projective structure under the plane-plus-parallax model. However, the constraints may be employed only to test given point correspondences or plane + parallax flow. No algorithm was given to reliably establish dense correspondences that are required by their approach. Fejes and Davis [34] used only the epipolar structure of rigid 3-D motion flow fields and developed a low-dimensional projection-based algorithm to separate independent motions. Their constraint can be fooled in situations where the direction of object motion and the epipolar flow direction are the same or similar; such is the case in some of our example sequences. Lourakis *et al.* [35] presented an algorithm that exploits both epipolar and shape constraints but the computation is based on a precomputation of normal flow that will be unreliable without additional constraints. Torr [36] uses model selection and segmentation for separating multiple 3-D motions. However, the dense separation of the scene and independent motions is not fully developed, and it is not clear how the technique will handle sparse 3-D scenes.

An alternative method to detect changes in presence of 3-D motion may be to compensate for the frame-to-frame background motion using the plane + parallax alignment algorithm and then apply the techniques described in the previous section for detecting change energy. However, this approach fails when the motion of the independently moving object is

in the same direction as the motion of the sensor. In this case, it is not possible to distinguish independent motion from parallax motion. Such a situation can be seen in Fig. 5(e).

There are two fundamental constraints that apply to the static 3-D scene and not to any independent motions. First, between two frames, all points on the fixed scene should satisfy the epipolar geometry constraint. However, there are particular situations where the epipolar constraint may be satisfied by a moving object too, for example, when an object is being tracked by a camera that is moving in the direction of the object motion. Second, the shape of the fixed scene with respect to a reference coordinate system should remain invariant to camera motions. This constraint can be employed over three or more frames. In general, for reliable separation of moving objects from the fixed background, both constraints need to be employed.

In [37], both the above fundamental constraints are used in an algorithm to solve the problem of independent motion detection when the parallax (or the “3-Dness” of the scene) is not dense, which is typically the case in AVS. It is not assumed that correspondences, optical flow or normal flow are available. The plane-plus-parallax decomposition [11]–[13] of multiview geometry and invariant projective shape is used to progressively introduce the constraints while solving the correspondence problem too. Multiple images are aligned together by progressively solving for the planar and parallax parameters. Regions of the images that cannot be aligned with either of the two fundamental constraints are labeled as independent motions. The algorithm tracks a dominant plane using homography transformations over a sequence. Subsequently, the mutually consistent epipoles over the whole sequence are robustly solved for using the depth constancy constraint over three or more frames. The homographies and epipoles are used to enforce depth

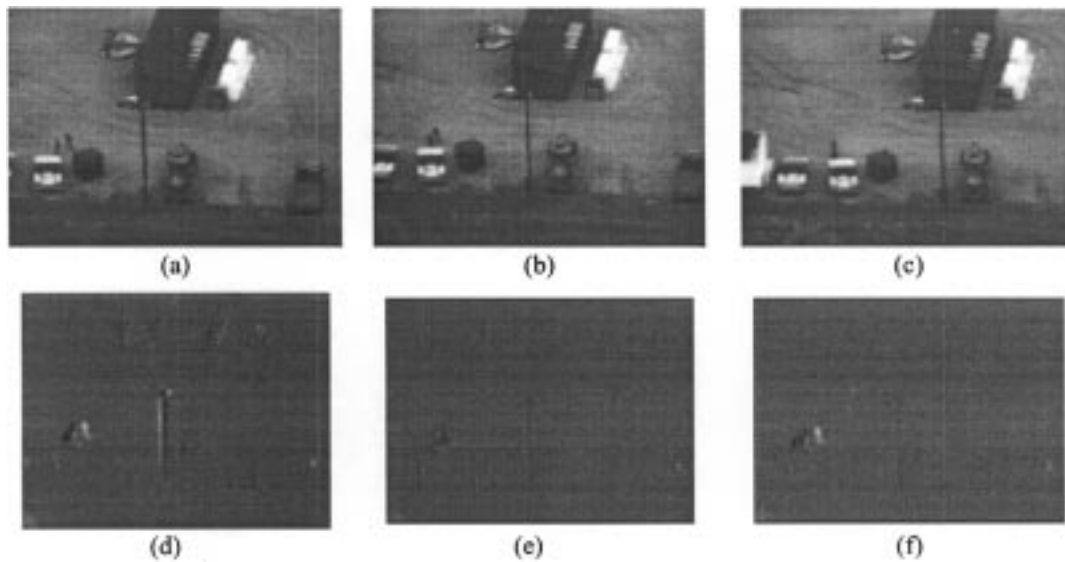


Fig. 5. (a), (b), (c) Frames from a source video that contains camera motion-pan and parallax motion—the telephone pole relative to the background, and object motion—two people walking, (d) difference images after 2-D scene stabilization removes the background but highlights both parallax (pole, building) and object motion, (e) difference image after compensation by plane + parallax alignment, almost everything is matched including the two moving objects and nothing is highlighted, and (f) difference image after compensation by three-frame 3-D algorithm, only the moving objects are highlighted.

constancy while aligning images within a coarse-to-fine framework. Finally, misaligned regions are detected and labeled as independent motions.

A comparison of 2-D and 3-D analysis is shown in Fig. 5. Fig. 5(a), (b), (c) shows three frames from an aerial sequence of two people running. Fig. 5(d) shows the difference image after compensation by a quadratic global motion. Much of the background is aligned. However both static objects such as the pole and the people are highlighted. Fig. 5(e) shows the difference image after compensation by the plane + parallax algorithm. In this case, there is almost perfect compensation and nothing is highlighted. This is because the motions of the camera and of the people are in a similar direction. Finally, Fig. 5(f) shows the difference image after compensation by the 3-D algorithm; note only the running people are highlighted.

G. Front-End Processor

Sarnoff has developed a family of special-purpose vision processors specifically tailored for front-end vision processing [2], [38], [39]. These processors use an optimized, parallel-pipelined architecture to perform fundamental front-end processing functions including pyramid transforms, geometric image warping with bilinear and bicubic image interpolation, space/time filtering, and global and local motion estimation. The latest processor in this family, known as the Acadia I integrated circuit, is a system-on-a-chip front-end vision processor capable of performing over 80 billion operations per second [39]. The Acadia I is provided on a PCI board form factor, as shown in Fig. 6, and is currently being used by Sarnoff to implement front-end processing functions described above at real-time video rates.

V. MOVING OBJECT TRACKING

A common requirement in aerial video surveillance is the capability to automatically detect and track moving objects in the scene. This can be challenging in AVS because targets appear small and their motions can be small compared to the camera induced scene motions. In Section IV-F, we discussed how to detect moving objects using change energy. Here, we discuss how to track moving objects.

In order to reliably track and maintain identity of objects over time, it is desirable for the Object State to contain representations of **motion, appearance, and ownership or shape in the image**. This is called an object *layer* [24], [40], [41]. With an object state represented as a layer, maximum *a posteriori* estimation (MAP) in a temporally incremental mode can be applied to update the state for tracking. Tracking with complete state representation is useful for applications that require segmented object appearance (for example, indexing and object insertion/removal) in addition to the traditional applications that require maintenance of only position and geometric transformations.

Most traditional trackers either use or estimate a partial representation only. For example, change-based trackers ignore the appearance information and, thus, have difficulty dealing with close-by or stationary objects. Template trackers typically only update motion and, hence, can drift off or get attached to other objects if there are instants of time when other similar templates are in close proximity [42]. Some template trackers use parametric motion (affine/similarity, etc.) to update both the motion and the shape of the template [43]; however, since no explicit updating of template ownership is done, even these may drift.

In [41], multiobject tracking is formulated as a 2-D layer estimation and tracking problem with a view toward

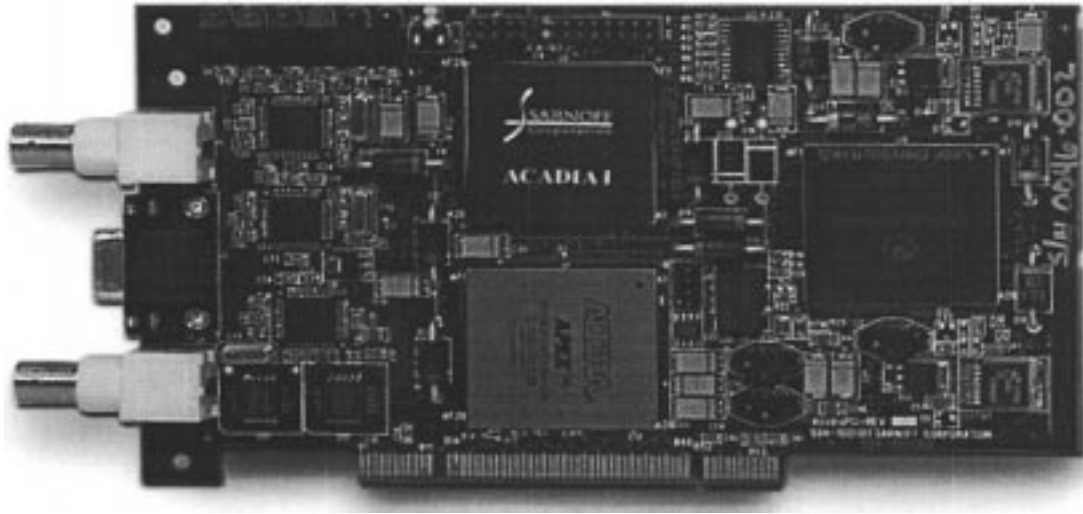


Fig. 6. The ACADIA chip on a PCI board is used to implement front-end AVS processing operations in real time.

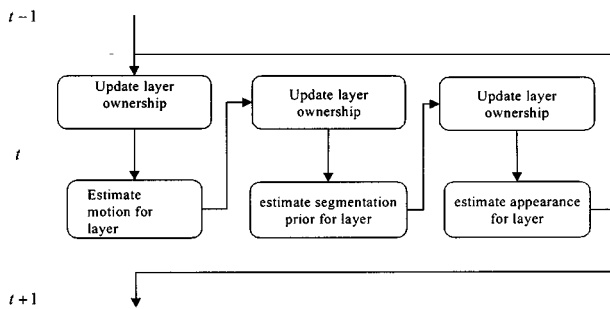


Fig. 7. Dynamic layer tracking algorithm.

achieving completeness of representation and, thereby, providing robust tracking in aerial videos when ambiguous or cluttered measurements occur. The layer tracker includes motion, appearance, and shape as the state representation. The state is updated at each instant of time using the Expectation-Maximization (EM) algorithm for MAP estimation. A dynamic Gaussian segmentation prior is introduced to encode the domain knowledge that the foreground objects have compact shapes. The dynamics of the segmentation prior are also modeled so that gradual changes over time are allowed. The motivation for employing such a global parametric shape prior is twofold. First, the prior imposes a preference on the shape of a foreground layer and prevents the layer from evolving into an arbitrary shape in the course of tracking. As a result, it assists in tracking when ambiguous or cluttered measurements occur. Second, only the compact parametric form of the prior function needs to be estimated, which makes the estimation process computationally efficient.

For each tracked object, at each time instant, its motion, appearance, and shape are estimated. Because it is difficult to optimize all three simultaneously, [41] adopts the strategy of improving one with the other two fixed. This is a generalized EM algorithm and it can be proven that this converges to a locally optimal solution. The diagram in Fig. 7 summarizes the optimization process.

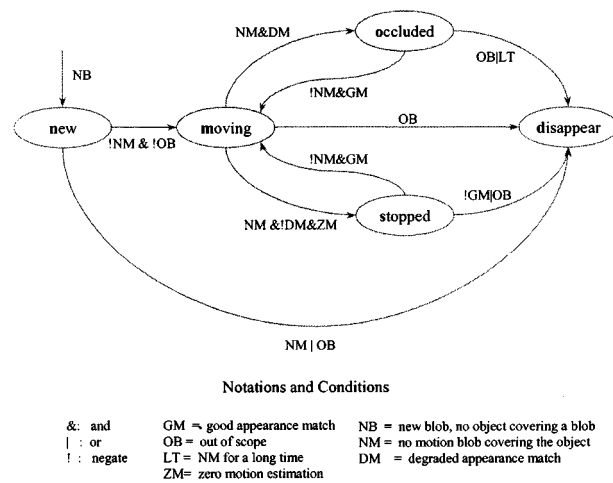


Fig. 8. State transition diagram for the dynamic layer tracker.

Besides the core tracking algorithm described above, additional issues which need to be handled are: 1) initialization of layers; 2) deletion and addition of layers; and 3) determination of object status such as stationary and occluded, which are important for some applications. These tasks are accomplished through a state machine (Fig. 8). In this state transition graph, there are five states and each directed edge represents a transition. The condition for transition is also marked along the edge. For example, a new object is initialized if a new change blob is detected far away from existing objects. An object is deleted if it is out of the field of view. An object is marked as stationary, if its motion blob disappears, there is no significant decrease of correlation score and the estimated motion is zero. When a new vehicle, or a layer, is added, an initialization step estimates the three components of a layer (motion, appearance, segmentation, or shape) from the change blob and the image. More specifically, the position of the object is located at the center of the blob. A zero velocity is assigned. The segmentation prior is estimated from the second-order moments of the blob. The appearance

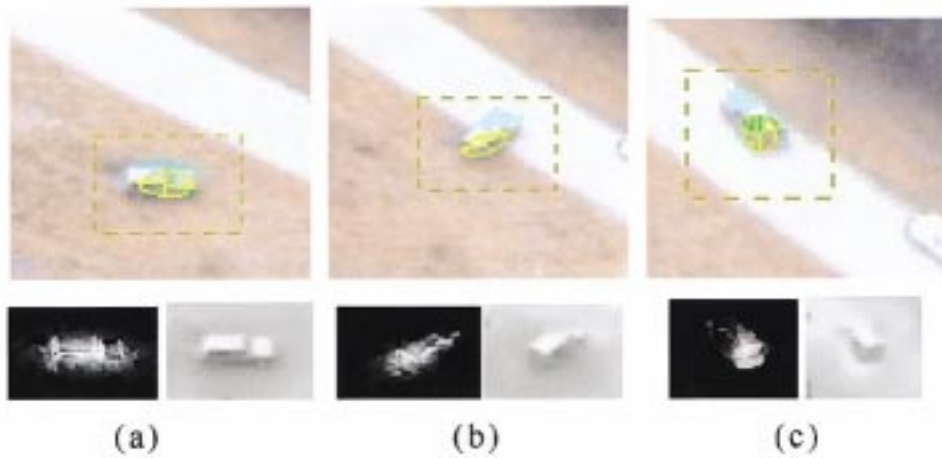


Fig. 9. Vehicle turning. The first row shows the cutouts of the original video frames and the Gaussian shape priors. The second row shows the segmentation and the appearance of the vehicles.

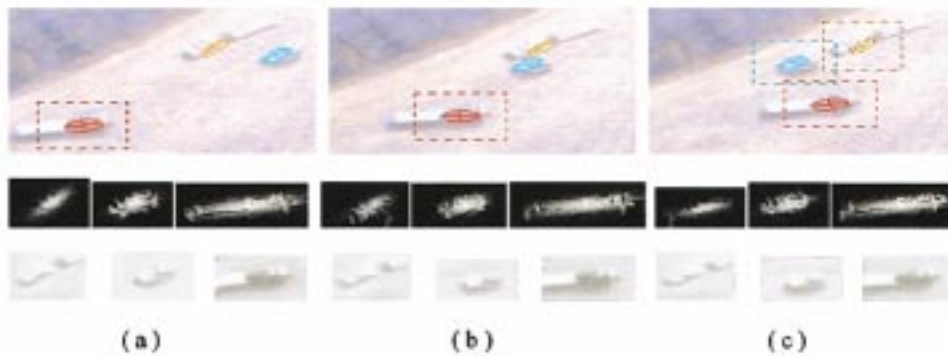


Fig. 10. Vehicle passing and stationary vehicles. The first row shows the cutouts of original video frames and the Gaussian shape priors. The second and the third rows show the segmentation and the appearance of the three vehicles.

is obtained from the original image. The initial segmentation image is a scaled version of the blob image.

The tracking system is designed to handle complex motions and complex interactions, such as passing and stop. In Fig. 9, the tracking result on a clip with a turning vehicle is demonstrated. In this example, the appearance, shape, and the motion of the vehicle changes dramatically. The layer tracker, however, has estimated them correctly and maintains the track.

Tracking vehicle interactions is difficult for change-based trackers because the change blobs merge. After they split, motion is the only cue to distinguish them, which is not reliable when the merge is not brief. A template tracker that does not keep track of ownership would lose track too when the other vehicle is treated as the background context. The layer tracker however, maintains the appearance information and reliable tracking can still be achieved. In Fig. 10, three vehicles are successfully tracked. One of them eventually becomes stationary. A change-based tracker cannot handle this scenario because appearance information is needed for tracking stationary objects.

VI. CAMERA CONTROL

A requirement of aerial video surveillance is that the camera be controlled in response to observations, in real

time. For example, once a target of interest is located, the operator may want to dwell on that target while he examines ongoing activity. If the target is stationary, then the camera must be systematically panned, as the aircraft flies by, in order to keep fixation on the target. If the target is moving, the camera must be systematically panned to follow its motion. At other times the camera must be systematically scanned over an extended scene region of interest. An AVS system should provide automatic camera control for each of these modes.

A camera can be directed to point at a specified geographic location, or sweep a defined geographic region, based on aircraft location, camera telemetry and terrain elevation data. In practice, such open loop control has limited precision due to errors in telemetry and terrain data.

Precise, stable camera control can be achieved by incorporating an image-based closed loop element in the control system. The scene itself provides a reference coordinate system that can be tracked precisely through image alignment. Discrepancy between the expected and observed image motions is fed back to refine camera control [44]. For example, when a stationary ground point is being fixated, there should be zero global image translation. Feedback is used to adjust camera control until residual motion is

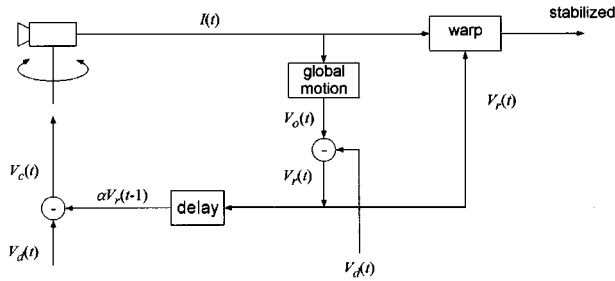


Fig. 11. Flow diagram for image-based camera control.

cancelled. If the target is moving, then feedback is used to adjust camera-tracking velocity until the target remains centered in the camera field of view. Likewise, when the camera is swept in an “S” pattern to scan a region of interest, successive scans should overlap one another by some prescribed amount. Observed errors in the extent of this overlap are used to adjust the camera control.

An overall camera control system is shown in Fig. 11. Assume $V_d(t)$ is a desired image scan velocity expressed in image coordinates. An initial estimate of the camera pan velocity needed to achieve this image velocity is computed from camera telemetry, aircraft velocity, and estimated distance from the aircraft to the scene.

Let $V_0(t)$ be the image velocity actually observed, based on global image to image alignment. The difference between the desired and the observed is the residual: $V_r(t) = V_0(t) - V_d(t)$. At the next frame time, the camera control velocity, $V_c(t)$, is then adjusted based on the observed residual

$$V_c(t) = V_d(t) - \alpha V_r(t-1) \quad (7)$$

where α is a feedback gain factor. At the same time, the observed residual can be used to electronically stabilize the output video to follow the desired image velocity $V_d(t)$ through a warp operation.

This type of feedback is also used to track moving objects. In this case $V_0(t)$ is an estimate of observed residual object velocity in the image domain. $V_r(t)$ is fed back to adjust camera pan so that it matches object motion. The velocity $V_d(t)$ is a small adjustment to the camera pan velocity needed to bring the target gradually to the center of the field of view.

VII. GEO-LOCATION

The utility of aerial video depends in large part on knowing precisely where the camera is pointing on the ground. The ideal objective is to know geographic coordinates—latitude, longitude, and elevation—for each pixel in the scene. This geo-coding information enables a wide range of functions. Ground structures and moving objects seen in video can be associated with real-world positions. Video can be superimposed on imagery, maps, and 3-D models from a geographic database to provide spatial context and collateral information for enhanced understanding of the video. Conversely, such database information can be transformed to, and overlaid on, video coordinates. Finally, the video can be used to revise the database itself, enhancing its spatial breadth and resolution or temporal fidelity.

Geo-coordinates can be provided by instruments on board the aircraft along with knowledge of the terrain over which the aircraft is flying. Geo-coordinates can also be provided by the registration of current video frames to previously obtained calibrated reference imagery.

The geo-coordinates of a point on the ground can be computed from the location of that point in the video frame and the geometry of the imaging process [44] (see Fig. 12). It may be assumed the location of the aircraft is known from GPS and other on-board navigational equipment. The orientation of the camera is known from sensors on the gimbals. The altitude of the aircraft above the ground plane is known from altimeters and terrain data. Data from these sensors constitute the engineering support data (ESD); (see also Section III) stream, which is synchronized with the video. An estimate of ground location can be obtained from a digital terrain map (DEM). The intersection of the image ray with the terrain profile gives the geo-coordinates of the point.

In practice, telemetry (ESD) based geo-location often does not provide coordinates of image points to a desired precision. Measurements of aircraft location and camera pointing angles have limited accuracy. Also, digital terrain data may not exist for the area under surveillance, or may be of relatively low resolution. The resulting geo-coordinates can be in error by tens or hundreds of meters on the ground.

The precision of geo-coordinates can often be significantly improved through image processing by aligning current video frames to previously obtained calibrated reference images [45]–[47]. Reference images have been collected for many regions of the world from satellites. Pixel positions for these images are known to a precision of few meters. Aerial video images inherit this precision when they are aligned to the reference images/models.

In the next two sections, we present solutions for two different cases of geo-registration. First, we discuss the registration of video to reference imagery and DEM data. Here, it is assumed the appearance between video and reference imagery can be matched. In the second case, we discuss alignment of video to 3-D site models of urban and suburban areas. In this case, there may be very large appearance differences between video and reference imagery. There can also be quite significant occlusion effects due to the large number of buildings present in the scene.

A. Geo-Registration to Reference Imagery and DEM Data

Fig. 13 shows the main components of a video based geo-registration system [46]. A reference image database is created in geo-coordinates along with the associated digital elevation maps (DEMs) and annotations [48], [49]. The visual features available in the reference imagery database are correlated with those in video imagery to achieve an order of magnitude improvement in alignment in comparison to purely telemetry (telemetry include location, orientation of the sensor, etc.) based alignment. To achieve this, a section of the reference image is first warped to the perspective of the UAV sensor based on the telemetry data alone. Subsequently, precise subpixel alignment between the video frame and reference imagery corresponding to the relevant locale

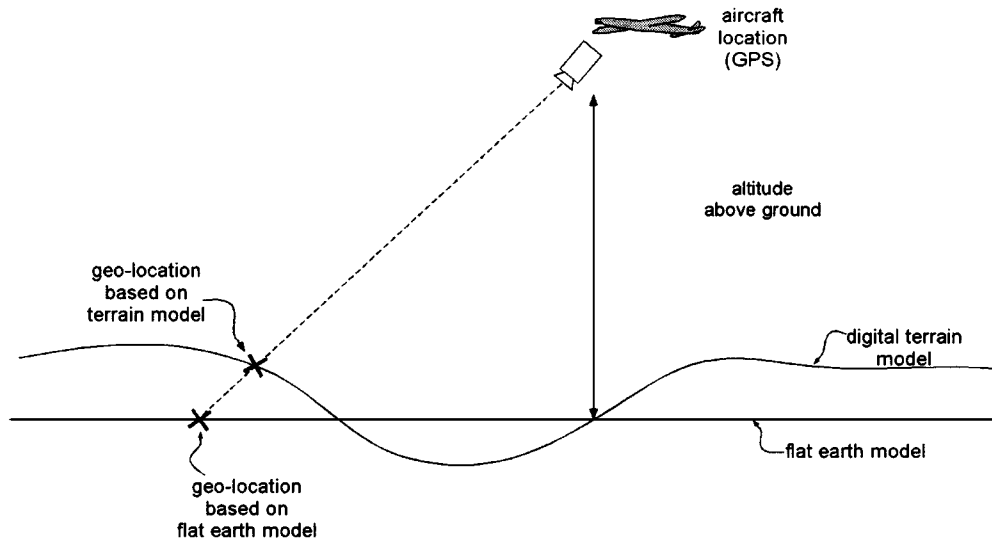


Fig. 12. Geometry of geo-location.

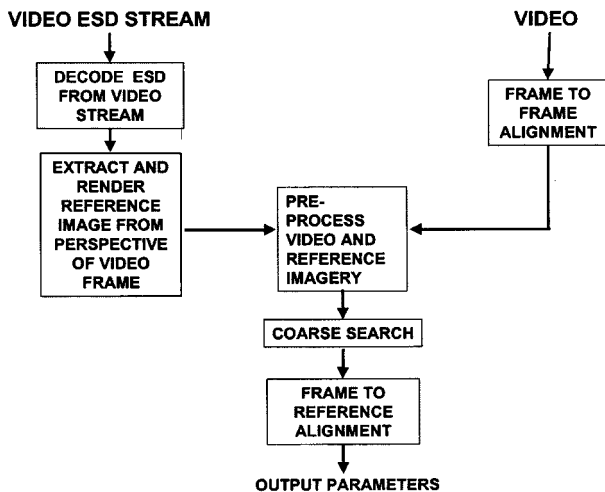


Fig. 13. Geo-registration system.

is used for the accurate geo-location of the video frame. The process of alignment of video to reference imagery is divided into the following steps.

1) *Frame-to-Frame Alignment*: Video frames at typically 30 frames per second contain significant frame-to-frame overlap. In order to meet the real-time constraints for the geo-registration system, as a first step in the front-end processing is to reduce redundancy by identifying *key* video frames based on computing frame to frame motion using the techniques presented in Section IV.

2) *ESD and Rendering Engine*: The engineering support data (ESD) supplied with the video is decoded to define the initial estimate of the camera model (position and attitude) with respect to the reference database. In the case of nadir imagery, video frames may be matched to reference imagery in the form of an orthophoto. However, in the case of oblique video, due to the large change in viewpoint, the appearance between the orthophoto and the video images can be quite different and it is difficult to find matching features. Instead, the video frames may be matched to rendered reference im-

ages which have approximately the same perspective as the video frames. This also minimizes occlusion effects, since the rendering process naturally does hidden surface removal. The camera model provided by the ESD is used to apply an image perspective transformation to reference imagery obtained from the database to render a set of synthetic reference images from the perspective of the sensor.

3) *Preprocessing*: There can be a great deal of appearance change between a video and its corresponding reference orthophoto, even following projection to a common coordinate frame. Many sources contribute to this change, including, variation in sensor characteristics, diurnal and seasonal cycles, and scene structure (e.g., new construction). To ameliorate such difficulties, it is desirable to choose a representation of image intensity that highlights pattern structure that is common to the two image sources that are to be brought into alignment.

In [46], the video and reference images are represented by Laplacian pyramids; these mitigate some changes in appearance such as constant offsets in gray scale and small-scale changes. However, Laplacian pyramids are not adequate to deal with changes such as orientation reversal and are not orientation selective. Features with the potential to serve in the desired fashion are those that exhibit a local dominant orientation or well-localized point-like structures that can be thought of as capturing a range of orientations, e.g., roads, tree lines, edges of buildings, compact isolated structures and the like. Similarly, the local distribution of orientations that are present in an image can be indicative of texture structure. Correspondingly, the image representation employed in [47] decomposes intensity information according to local spatial orientation. This representation is derived via application of a bank of filters that are tuned for spatial orientation and scale to both the video imagery as well as the (projected) reference image. Filters are applied at a set of four orientations—vertical, horizontal and two diagonals—to yield a corresponding set of “oriented energy images” for both the video and reference imagery.

4) *Coarse Search*: A coarse indexing module then locates the video imagery more precisely in the rendered reference image. However, an individual video frame may not contain sufficient information to perform robust matching and, therefore, results are combined across multiple frames using the results of frame-to-frame alignment. In [45], matching is performed by correlating image features, present in multiple video frames, to the reference image. The correlation surfaces for individual image features often have multiple peaks. Disambiguation is obtained by imposing global consistency by combining the video frame-to-frame motion information with the correlation surfaces across multiple frames. The correlation-based search is done across a range of rotation, translation, and zoom motion parameters. The net result is that the sequence of frames is located to within a few pixels in the reference frame. The final correlation score is used as a measure of accuracy in the coarse search step.

5) *Fine Registration*: A fine geo-registration module refines the coarse estimate further using the relative information between frames to constrain the solution. In general, the transformation between two views of a scene can be modeled by: 1) an external coordinate transformation that specifies the 3-D alignment parameters between the reference and the camera coordinate systems and 2) an internal camera coordinate system to image transformation that typically involves a linear (affine) transformation and nonlinear lens distortion parameters. In [46] and [47], the precise alignment formulation combines the external coordinate transformation and the linear internal transformation into a single 3-D projective view transformation. Twelve parameters (a_1 to a_{12}) are used to specify the transformation

$$\begin{aligned} X_I &= \frac{a_1 * X_r + a_2 * Y_r + a_3 * k(X_r, Y_r) + a_{10}}{a_7 * X_r + a_8 * Y_r + a_9 * k(X_r, Y_r) + a_{12}} \\ Y_I &= \frac{a_4 * X_r + a_5 * Y_r + a_6 * k(X_r, Y_r) + a_{11}}{a_7 * X_r + a_8 * Y_r + a_9 * k(X_r, Y_r) + a_{12}}. \end{aligned} \quad (8)$$

The 3-D depth of each scene point (X_r, Y_r) is represented by the parameter $k(X_r, Y_r)$. The reference image coordinates (X_r, Y_r) are mapped to the ideal video coordinates (X_I, Y_I) by (8).

This transformation together with the DEM data and any nonlinear lens distortion parameters completely specifies the mapping between the video pixels and those in the reference imagery. One major advantage of this approach is that camera calibration need not be known. This increases the applicability of our proposed system to arbitrary video camera platforms. Note in many aerial imaging instances, (8) can be reduced to be a projective transform (where the terms $a_3 = a_6 = a_9 = 0$). This approximation is valid when there is only a small translational viewpoint difference between the rendered reference image and the video frame or the distance between camera to ground is large as compared to the height of objects in the scene.

In [45], the fine alignment of each frame is done separately. However, two problems occur using this approach: 1)

the individual video frames may be quite different from the reference imagery and 2) certain video frames may not have sufficient distinguishing information to match them to the reference imagery. The appearance differences can be due to multiple reasons such as changes in the world, different image acquisition times, different imaging sensors used, etc. To mitigate against these effects, [46] and [47] match a block of frames simultaneously to the reference imagery. The block of frames would provide a larger context for robust matching. Note that the frame to frame alignment within the block can be stable because of local features.

In the bundle-based approach (Fig. 14), the results from the frame-to-frame alignment processing are used to constrain the simultaneous alignment of several sets of frames to a set of rendered reference images. As noted earlier, the video frames are matched to rendered reference images whose perspective is close to the video frame. In the block alignment scheme for an oblique video sequence, different video frames are matched to different rendered reference images (see Fig. 14). However, since the reference images are rendered by the system, the relationship between them is completely known. Tie-points are established between the reference frames and video frames and also between the video frames. Kumar *et al.* [46] estimates tie-points by computing optic flow [9] between the Laplacian pyramid representations of the images. However, this process is susceptible to errors, as noted earlier, because of large appearance changes between reference and video imagery. Wildes *et al.* [47] improve on the process by using oriented energy pyramids to represent the images. These provide features that are more stable to match under gross appearance changes. Wildes *et al.* [47] use robust statistics to test and remove outliers in the frame-to-reference tie-point estimation. The frame-to-reference parameters are solved by minimizing the error term in (9) with respect to the frame-to-reference parameters (a_1 to a_{12}) in (8)

$$E = \sum_{i=1}^k E_{f2f}(i, i+1) + \sum_{j=1}^{j=m} \sum_{i=1}^k E_{r2f}(j, i). \quad (9)$$

The energy term E_{f2f} is the sum of squares of the geometric displacements between corresponding tie-points belonging to a pair of neighboring video frames. The energy term E_{r2f} is a similar geometric displacement error between corresponding tie-points belonging to a video frame and a reference frame. E is the sum across all video frames of the frame-to-frame matching errors (E_{f2f}) and rendered reference frame to video frame matching errors (E_{r2f}). In Fig. 14, we show three rendered reference images with video frames tied to each of them. In practice, the number of rendered reference frames depends on the motion of the video in the bundle of video frames. The number of video frames used in a bundle depends on there being enough multidirectional features present across the bundle to be able to tie them robustly to the reference imagery. For real-time applications, we use a sliding sub-bundle scheme, where a new set of frames is

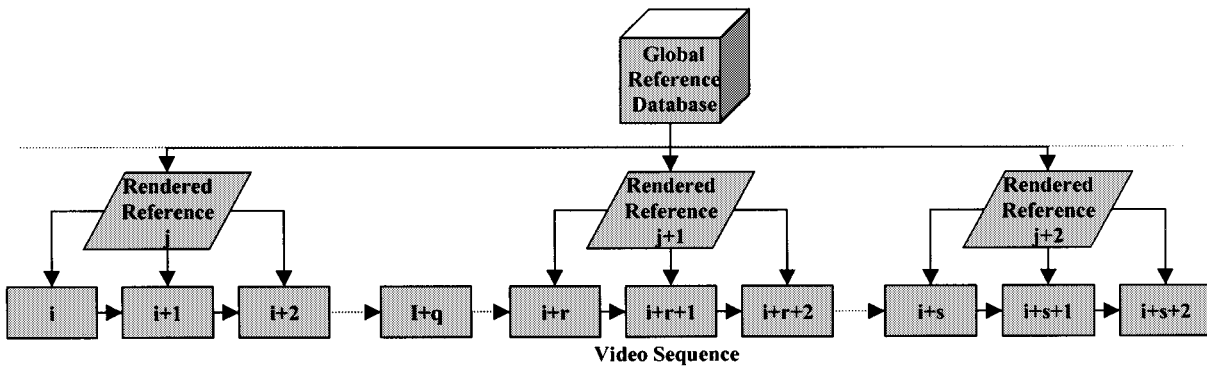


Fig. 14. Multibundle alignment of oblique video frames to reference models.



Fig. 15. Geo-registration of video. (a) Overlay of video mosaic over ortho-photo using ESD information alone. (b) Overlay of video-mosaic over ortho-photo after coarse search step. (c) Overlay of video mosaic over ortho-photo after fine alignment step.

added at each time instant and an earlier sub-bundle of frames is removed from the estimation.

a) *Results:* Fig. 15 shows geo-registration results that were obtained from video captured from an X-drone UAV flying over Webster Field, MD. Key frames obtained at 3 Hz were used to geo-register the sequences. Fig. 15(a) shows the overlay of the video mosaic over the reference imagery using ESD information alone. The video mosaic is not aligned and the geo-location error is about 1000 feet (hundreds of pixels). Fig. 15(b) shows the geo-registration result after the coarse search step. The video mosaic is now aligned to within a few pixels from the reference image. Finally, Fig. 15(c) shows the result after the fine alignment step [46], where it can be noted there is quite precise alignment.

Fig. 16 shows a more difficult but successfully solved case of geo-registration when large appearance changes are present [47]. The video imagery in this case was acquired in the winter while the reference imagery was obtained a few years ago in the summer. Further, the video is imaged at a moderately oblique angle and the terrain is hilly.

B. Registration to Site Models

Urban and suburban regions with a large number of buildings are not sufficiently well represented by extant reference imagery and digital elevation models. In this case, the representation is augmented with the use of site models, where each building is modeled using polygonal structures and overlaid on the reference imagery and terrain [48].

The problem of registration of video to site models is most related to work on object recognition and tracking.

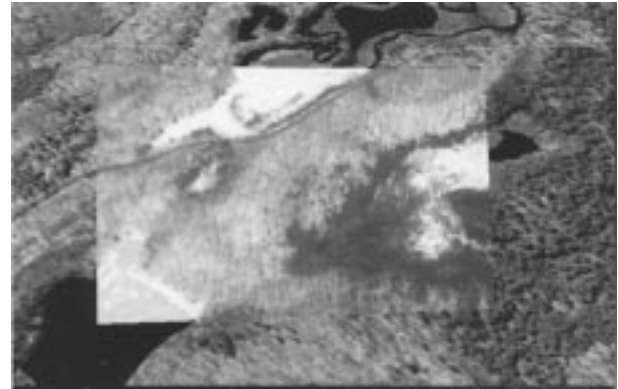


Fig. 16. Overlay of video image of Fort Drum over rendered reference image. Reference image was taken in summer, video was taken after snowfall in winter.

Model-based alignment by explicitly extracting features from the image, identifying their correspondence to model features, then solving for absolute orientation, is one of the standard approaches [50]–[52], whose drawbacks include unreliability of feature detectors and combinatorial complexity of matching. Hsu *et al.* [53] follow the alternative correspondence-less approach, which has been used in deformable template-based object recognition [54] and object tracking applications [55], [56]. In aerial video surveillance applications, any single video frame captures only a small part of the overall scene. Not only does the “object” occupy a large field of view, unlike in the above cited works, but there are often pose ambiguities in single images that should be resolved by combining information across frames.

Hsu *et al.* [53] presents a potentially real-time “direct” method for pose refinement, which simultaneously estimates pose parameters and the correspondence between features. Tracked features are used to predict the pose from frame to frame and the predicted poses are refined by a coarse to fine process of aligning projected 3-D model line segments to oriented energy pyramids. The existing site model is considered to be a collection of untextured polygonal faces. Face edges in the given model imply discontinuities in surface normal and/or material properties in the actual 3-D scene, which generally induce brightness edges in the image. The alignment method selects 3-D line segments from the model and projects them onto the video frame using the current pose

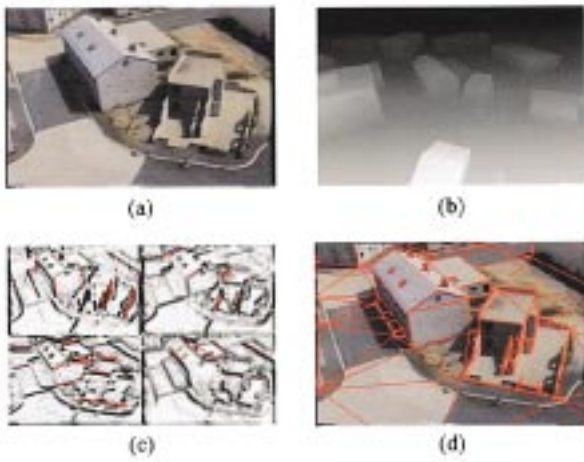


Fig. 17. Geo-registration of video to site models: (a) original video frame, (b) 3-D site model, (c) oriented energy image with four major orientations: 0° , 45° , 90° , and 135° . Model lines used for alignment are overlaid on top in red and (d) model lines are projected on to image using estimated pose.

estimate. The local edge strength in the image itself is represented as an oriented energy field. Model lines are projected to the orientation image which is nearest their orientation.

Fig. 17(a) shows a video frame and Fig. 17(c) shows model lines projected (red color lines) for over an oriented pyramid representation of the video frame in Fig. 17(a). The optimization process varies the pose to maximize the integral of this oriented energy field along the projected 3-D line segments, causing projected line segments to move toward loci of high energy. This procedure is used to estimate the pose for any frame. Fig. 17(b) shows the 3-D model and Fig. 17(d) shows the model lines rendered and overlaid over the image using the estimated pose parameters.

An initial estimate or prediction of pose is needed to bootstrap the above registration procedure. Various prediction functions may be used. In an interactive system, a user can set the pose for the first frame. Physical measurements from position/attitude sensors mounted on the camera platform may be another source. Finally, when processing a sequence of frames, the pose can be predicted from the previous frame's estimated pose. Hsu *et al.* [53] use the interframe tracking of features between frames i and $i - 1$, plus the depth of features in $i - 1$, to predict the 3-D pose of image i . Features are tracked using an optic flow technique [9]. The predicted pose parameters are computed using a robust statistics technique, which is able to handle outliers in the matching procedure [52], [57].

Once pose has been estimated, the initially untextured rough site model can be refined. The same approach used for pose estimation can also be used to refine the placement and shape of parameterized object models. Estimation of a dense parallax field can be used to refine the shape of known objects and to represent objects that have newly appeared or were otherwise previously unmodeled. The appearance of model surfaces can be recovered by mapping texture from the video frames to the faces of the model.



Fig. 18. Video flashlight: video frame is warped and overlaid over model; scene may be visualized from any viewpoint.

The foregoing process updates a static model of the scene, but does not capture temporal information. Alternatively, the pose recovery process can also enhance the visualization of dynamic information in live video. Aerial surveillance video is often unstable and narrow field of view, hence difficult to comprehend. Aligning it to a geometric model allows us to reproject and visualize the video from different points of view with a larger context of the static scene embedded as background (Fig. 18). Each video frame is like a flashlight, which illuminates the static 3-D model with the latest imagery. Alignment of video to true world coordinates also allows us to annotate the video, insert synthetic objects and find the 3-D geo-location of image points.

VIII. AERIAL MAPPING

One common objective of aerial video surveillance is "aerial mapping," the collection of extended images of the landscape as geodetically indexed video mosaics. As an aircraft flies, its camera is systematically swept over a region of interest. The video frames are aligned and merged into a mosaic. The mosaic is ortho-rectified and aligned to map coordinates. The resulting image is similar to a panoramic satellite photograph. In Section IV, a front-end mosaicing process was described. Two refinements of the mosaicing process are needed to construct extended mosaics: 1) ortho-rectification and geo-coding to project the component image frames into a common geo-coordinate frame and 2) global alignment of the image frames. Fig. 19 illustrates this procedure.

Alignment of consecutive video frames alone cannot be used to construct extended mosaics in which frames are arrayed in a two dimensional pattern. When the scene is scanned with a back and forth motion as the aircraft flies, small errors that occur in aligning successive image frames gradually accumulate, so that successive scans do not align precisely.

The frame-to-frame alignment process can be extended to create mosaics of frames that are captured using 2-D scans of the camera. This "local to global" procedure has three steps, as shown in Fig. 20 [58]. First, alignment parameters are

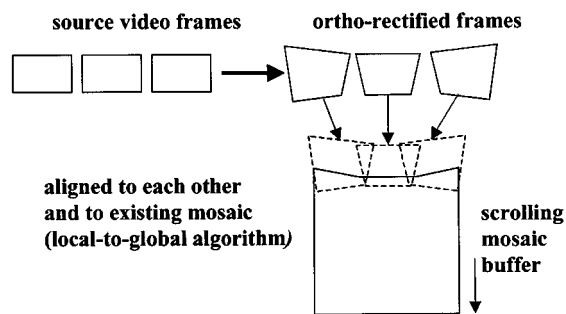


Fig. 19. Steps in constructing an ortho-rectified mosaic.

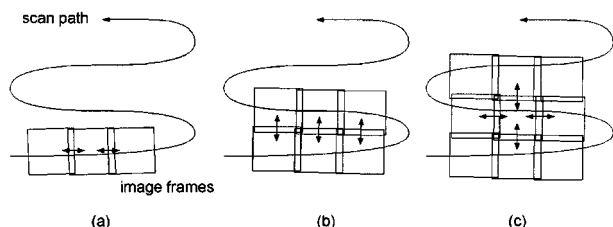


Fig. 20. Local to global alignment for constructing extended mosaics. (a) Compute alignment between successive video frames, (b) compute alignment between frames in successive scans, and (c) solve for the best simultaneous alignment of each frame to all its neighbors.

computed between successive video frames as video is received (the front-end processing step). Second, similar alignment parameters are computed between overlapping frames in successive left-right scans. Third, a global alignment of the frames is computed from the local alignment that represents a best simultaneous alignment of each frame with all its neighbors and geodetic information provided in the ESD stream accompanying the video. The aligned frames are then merged using multiresolution methods to form a seamless mosaic.

When the spatial layout of the video frames is not known *a priori*, an automatic topology inference algorithm can be used to choose the spatially, but not temporally, adjacent frames to be aligned. The collection of frames and pairs of registered frames forms a graph. An iterative process alternates between refining the simultaneous alignment given the current graph and refining the graph (discovering overlapping frames) given the current alignment [58].

A further refinement to the mosaicing process is needed to construct extended aerial mosaics: ortho-rectification and geo-coding. As the camera is swept over the scene, the “footprint” of the camera’s view on the ground changes size and shape. The footprint will be roughly rectilinear when the camera points straight down, but will become larger and more key-stoned as the viewing direction is made more oblique. The image footprint will be further distorted by hilly or mountainous terrain.

In order to construct a consistent mosaic from such images, it is necessary to reproject each image frame to a common viewing direction and scale. The appropriate common direction is nadir, so that the resulting images can be aligned to map coordinates. Reprojection again makes

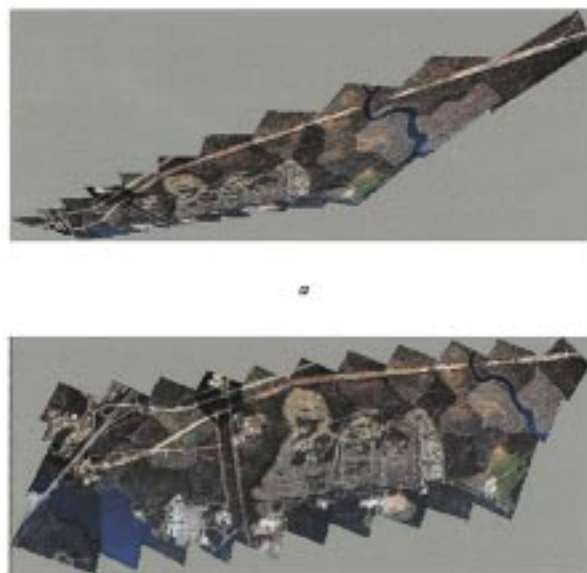


Fig. 21 (a) Mosaic constructed using frame to frame constraints alone and (b) ortho-mosaic constructed using local to global constraints and warping imagery to nadir viewpoint.

use of a terrain model and camera coordinates. If the terrain is not too rough, a flat earth model can be used. If the terrain is rough, an elevation model needs to be used to warp the image to be an orthographic projection from a nadir viewpoint. The terrain may be provided by other sensors such as laser range finders or recovered from the video stream.

In order for the scale, orientation and position of the mosaic to be correlated with true geography, world to image geo-referencing constraints should also be imposed. Such constraints may be derived from telemetry, point correspondences with respect to a map or ground survey, or geo-registration to reference images (Section VII).

Fig. 21 shows construction of image-domain and ortho-mosaics constructed from a highly oblique aerial video sequence obtained from an airplane flying 1 km above ground, where the periodic side-to-side scan is effected by a 23° camera rotation. Fig. 21(a) shows the mosaic constructed using 50 such frames using the image domain methods presented in Section IV. Because the central frame was chosen as the reference coordinate system, the mosaic is distorted with respect to a nadir view and the scale changes greatly across the mosaic. Fig. 21(b) shows the ortho-mosaic for the same region constructed using the topology inference and the local-to-global alignment technique with telemetry-based geo-referencing. Note that the scene appears to have the same scale throughout the ortho-mosaic.

IX. COMPRESSION AND TRANSMISSION

In AVS systems such as that shown in Fig. 1 video is transmitted to the ground for viewing by an operator. Often this transmission must be done over a very low-bandwidth communications channel and significant compression is required. The image processing already done on board the

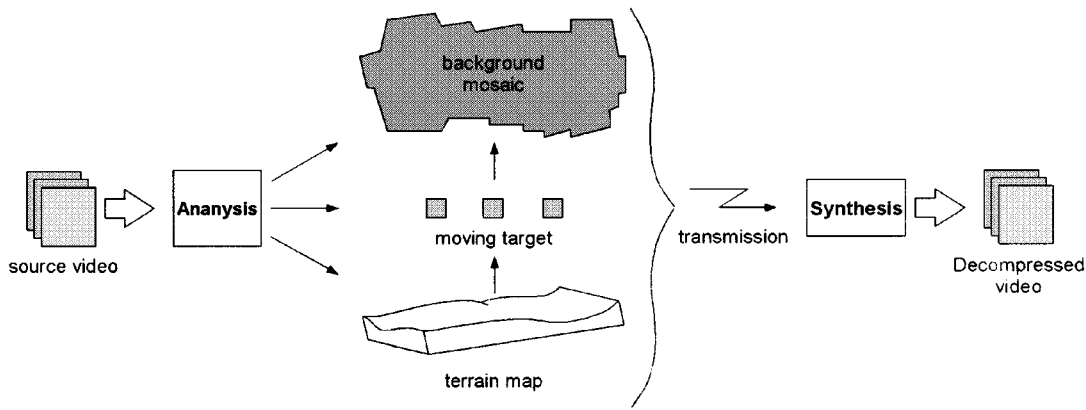


Fig. 22. Object-based compression of aerial video. Video is represented compactly by a background mosaic and foreground moving objects. A terrain map may be transmitted to account for parallax motion. The video is resynthesized at the receiver.

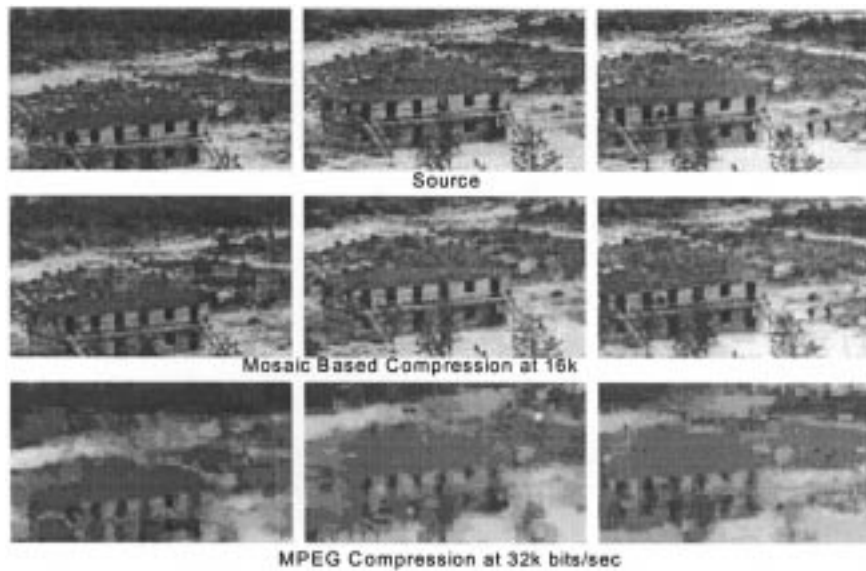


Fig. 23. Comparison of mosaic-based compression to MPEG-2 compression. Top row contains example source frames from a sequence in which people enter a building. (The images are at 360×240 resolution and temporally sampled by four, 7.5 frames/s). Mosaic based compression is shown in the middle row, at an extremely low data rate, 16 kb/s. MPEG compression at twice this data rate is shown in the bottom row. Despite its lower bit rate, mosaic compression provides significantly better quality.

aircraft provides a basis for high performance “object-oriented” compression. An example AVS compression scheme is shown in Fig. 22 [59]. The source video is divided into two components: the static background and foreground moving objects. The background scene is represented as an extended mosaic and, optionally, a digital elevation map. This achieves compression by removing redundancy present in consecutive video frames using frame-to-frame alignment and motion parallax computation. The background mosaic is further compressed using standard intraframe coding techniques, such as wavelet compression. The mosaic is transmitted progressively as it is constructed.

Foreground objects are segmented from the background and are transmitted separately as regions of change. Because these objects tend to be small in the image, the data required for transmission is also small. At the ground, the background

and foreground components are decompressed and recombined at the receiving end to regenerate the video sequence.

Further compression can be achieved by adjusting mosaic resolution and fidelity based on content. While targets and their surrounding regions are sent at the highest resolution and free of compression artifacts, surrounding countryside can be transmitted at reduced resolution and quality. The performance gains enabled by this mosaic-based compression are substantial. A typical example comparing this method with MPEG-2 is shown in Fig. 23.

X. DISPLAY

One of the most challenging aspects of aerial video surveillance is formatting video imagery for effective presentation to an operator. The soda straw nature of aerial video makes direct observation tedious and disorienting.

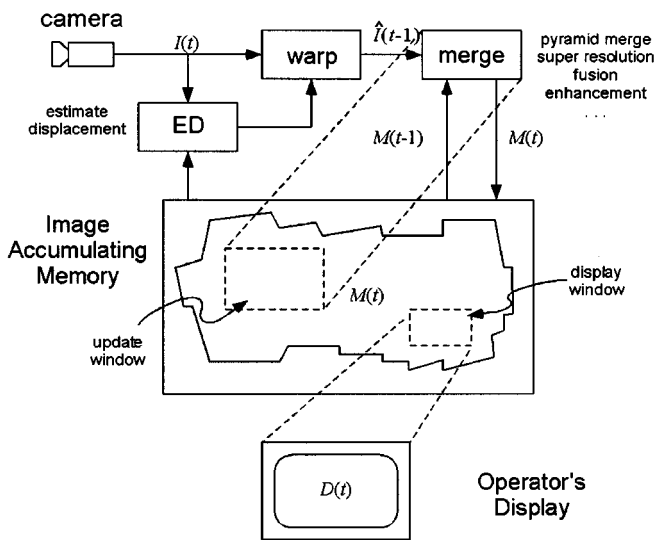


Fig. 24. Elements of a mosaic display.

These shortcomings of video can be overcome, to a large extent, through the use of a “mosaic-based display.” Again, this makes use of the image processing functions and image representations that serve automated video analysis on board the aircraft, but now to support human interpretation on the ground.

Elements of the mosaic display are shown in Fig. 24. The mosaic display decouples the observer’s display from the camera. An operator may scroll or zoom to examine one region of the mosaic even as the camera is updating another region of the mosaic.

As a further step, the mosaic is reprojected to nadir view and aligned with map coordinates. This provides a link between current video and known geographic information about the scene, such as maps, prior imagery of the area, and names and classification data for specific cultural features in the scene such as roads and building and natural features such as rivers and mountains. The current images can be overlaid on prior images or maps for direct comparison. The imagery can be annotated automatically with the names of landmarks of interest.

As a further generalization, the mosaic can be overlaid on a terrain model of the scene, then rendered from an arbitrary user selected viewing direction. In this way, the observer can “fly through” the scene independently of the aircraft’s own motion.

The techniques described thus far provide the operator with an improved representation of the static background areas of a scene under surveillance. Extensions of these techniques provide improved representations of foreground scene activity as well. The locations of moving objects can be highlighted on the mosaic display. As they move, their tracks can be represented symbolically as a sequence of lines or dots, or as images of the moving object itself inserted at regular time intervals, as shown in Fig. 25. This “synopsis mosaic” provides the observer with a summary of extended scene activities that can be interpreted at a glance [60], [61]



Fig. 25. Synopsis mosaic obtained while tracking a white truck along a road. The appearance of the truck at multiple locations is overlaid on the mosaic.

The operator can also be given the ability to replay scene activities as “dynamic mosaics” as he or she would replay a recorded video. However, now as video frames are replayed, they are overlaid on the full background mosaic, so objects appear to move over the mosaic. Such dynamic mosaics enhance perception by presenting activities in a larger scene context.

Finally, the mosaic display can provide the operator with a convenient means for directing the camera and the overall surveillance operation. The operator can use a pointing device, such as a mouse, to designate targets on the display to be fixated or tracked. If the mosaic is overlaid on a map or reference image, the operator can use the pointing device to indicate regions of the scene to be scanned.

XI. ARCHIVING

Aerial video often needs to be recorded and stored for later analysis. Raw video contains prodigious amounts of data, however, so it is costly to store. At the same time, very little of this video contains information that could be of later interest. Simply storing video would make later access impractical, as a human operator would need to search the database by replaying video that was already tedious to watch when it was viewed live. Thus, a practical archiving system needs to provide effective video compression and indexing. The image processing techniques and image representations used in the AVS systems for real-time analysis and display also provide solutions to compression and indexing for archiving.

A system for automatically generating video abstracts for archiving and browsing is shown in Fig. 26 [61]. Source video and associated geo-location data are processed continuously in real time. The video is first decomposed into scene cuts, then each cut is represented as a mosaic for compact storage. Targets and target tracks are stored with each mosaic, as are geo-locations. The video may be indexed using attributes such as time, geo-location and spatial coverage, appearance of background or foreground objects, target tracks,

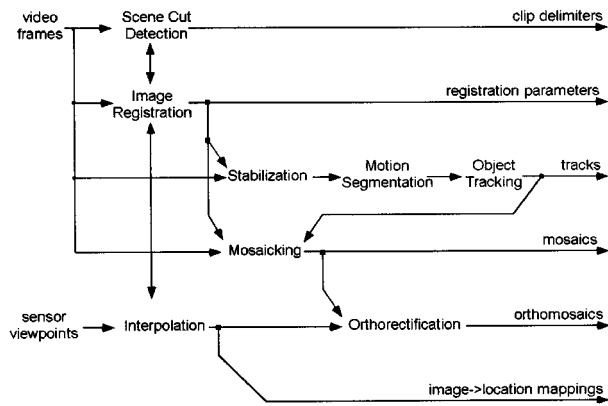


Fig. 26. Video abstraction for storage.

etc. Synopsis mosaics and key frames may be used to effectively browse through the video.

XII. CONCLUSION

We have outlined an integrated systems approach to aerial video surveillance. A small set of basic image processing functions and image and meta-data representations have been shown to serve all stages of AVS processing, from the camera to onboard analysis, video compression, display, and archiving.

We have divided AVS image processing on board the aircraft into front-end and analysis stages. Roughly speaking, front-end processing is at the signal level and characterized by two spatial dimensions and time. Analysis, on the other hand, is at an object and scene level and characterized by three spatial dimensions and time.

The most fundamental AVS image processing function is image alignment—alignment of successive image frames and alignment of current frames to previously collected imagery. At the front-end stage of AVS processing, a moving window of image frames is aligned to the current frame, then a set of space/time filters applied to this window serves to reduce noise and background clutter, to generate attribute and mosaic images and to detect moving objects. At the analysis stage, image alignment serves to track moving objects and to locate landmarks in the scene. Alignment provides visual feedback to implement precise camera control. Alignment to reference imagery is the basis for geo-registration and provides the bridge between current imagery and all previously collected information about a scene that may be stored in a geographic information database.

3-D scene analyses in AVS can be represented as parallax maps, digital elevation maps, and site models. Generalized image alignment is used to recover these elevation maps and site models from video. Pattern search and parallax compensation are then converted from 3-D to 2-D processing by reprojecting images through the 3-D representations. Frame-to-frame alignment and image integration is used throughout the AVS system to generate mosaics. These provide a basis for video compression, visualization, and archiving.

ACKNOWLEDGMENT

The research reported in this paper is the work of a large number of current and past members of the Vision Technologies Laboratory at Sarnoff Corporation. Only a few of them are listed here as co-authors.

REFERENCES

- [1] P. R. Wolf and B. A. Dewitt, *Elements of Photogrammetry with Applications in GIS*, 3rd ed. New York: McGraw Hill, 2000.
- [2] P. Burt, P. Anandan, G. van der Wal, and R. Bassman, "A front-end vision processor for vehicle navigation," in *Proc. Int. Conf. Intelligent Autonomous Systems*, 1993.
- [3] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532–540, Apr. 1983.
- [4] W. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 891–906, Sept. 1991.
- [5] B. Jahne, *Digital Image Processing*, Berlin, Germany: Springer-Verlag, 1988.
- [6] R. Wildes and J. Bergen, "Qualitative spatio-temporal analysis with an oriented energy representation," in *Proc. Eur. Conf. Computer Vision*, Ireland, 2000, pp. 768–774.
- [7] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, July 1985.
- [8] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA: MIT Press, 1993.
- [9] J. R. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proc. Eur. Conf. Computer Vision*, 1992.
- [10] K. J. Hanna and N. E. Okamoto, "Combining stereo and motion analysis for direct estimation of scene structure," in *Proc. IEEE Int. Conf. Computer Vision*, Berlin, 1993, pp. 357–365.
- [11] R. Kumar, P. Anandan, and K. Hanna, "Direct recovery of shape from multiple views: A parallax based approach," in *Proc. Int. Conf. Pattern Recognition*, Jerusalem, Israel, 1994.
- [12] H. Sawhney, "3D geometry from planar parallax," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1994.
- [13] A. Shashua and N. Navab, "Relative affine structure: Theory and application to 3D reconstruction from perspective views," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1994.
- [14] J. J. Koenderink and A. J. van Doorn, "Affine structure from motion," *J. Opt. Soc. Amer. A*, vol. 8, no. 2, pp. 377–385, 1991.
- [15] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *Int. J. Comput. Vis.*, vol. 2, pp. 283–310, Jan. 1989.
- [16] B. D. Lucas and T. Kanade, "An iterative technique of image registration and its application to stereo," in *Proc. 7th Int. Joint Conf. Artificial Intelligence*, 1981, pp. 674–679.
- [17] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artif. Intell. J.*, vol. 78, pp. 87–119, Oct. 1995.
- [18] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [19] B. K. P. Horn and E. J. Weldon, Jr., "Direct methods for recovering motion," *Int. J. Comput. Vis.*, vol. 2, pp. 51–76, June 1988.
- [20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1993.
- [21] M. J. Black and P. Anandan, "The robust estimation of multiple motions: parametric and piecewise smooth flow fields," *Comput. Vis. Image Understand.*, vol. 63, pp. 75–104, Jan. 1996.
- [22] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, "Real-time scene stabilization and mosaic construction," in *Proc. IEEE Workshop on Applications of Computer Vision*, 1994.
- [23] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna, "Representation of scenes from collections of images," in *Proc. IEEE Workshop on Visual Representations*, Cambridge, MA, 1995.
- [24] H. S. Sawhney, S. Ayer, and M. Gorkani, "Model-based 2D and 3D dominant motion estimation for mosaicing and video representation," in *Proc. Int. Conf. Computer Vision*, 1995, pp. 583–590.

- [25] R. Szeliski, "Video mosaics for virtual environments," *IEEE Comput. Graph. Appl.*, vol. 16, pp. 22–30, Mar. 1996.
- [26] P. Burt and T. Adelson, "A multi-resolution spline with application to image mosaics," *ACM Trans. Graphics*, vol. 2, pp. 217–236, 1983.
- [27] C. H. Anderson, P. J. Burt, and G. S. van der Wal, "Change detection and tracking using pyramid transform techniques," *SPIE Intell. Robots Comput. Vis.*, vol. 579, pp. 72–78, 1985.
- [28] P. Burt, J. Bergen, R. Hingorani, R. Kolczynski, W. Lee, A. Leung, J. Lubin, and H. Shvaytser, "Object tracking with a moving camera: An application of dynamic motion analysis," in *Proc. IEEE Workshop on Motion*, 1989.
- [29] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions," *Int. J. Comput. Vis.*, vol. 12, pp. 5–15, Feb. 1994.
- [30] R. Wildes, "A measure of motion salience for surveillance applications," in *Proc. IEEE Conf. Image Processing*, 1998, pp. 183–187.
- [31] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving target classification and tracking from real-time video," in *Proc. IEEE Workshop on Applications of Computer Vision*, 1998.
- [32] A. Selinger and L. Wixson, "Classifying moving objects as rigid or nonrigid without correspondences," in *Proc. DARPA Image Understanding Workshop*, 1998.
- [33] M. Irani and P. Anandan, "A unified approach to moving object detection in 2D and 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 577–589, June 1998.
- [34] S. Fejes and L. Davis, "What can projections of flow fields tell us about visual motion," in *Proc. Int. Conf. Computer Vision*, 1998.
- [35] M. I. A. Lourakis, A. A. Argyros, and S. C. Orphanoudakis, "Independent 3D motion detection using residual parallax normal flow fields," in *Proc. Int. Conf. Computer Vision*, 1998.
- [36] P. H. S. Torr, A. Zisserman, and S. Maybank, "Robust detection of degenerate configurations for the fundamental matrix," *J. Comput. Vis. Image Understand.*, vol. 71, pp. 312–333, Sept. 1998.
- [37] H. S. Sawhney, Y. Guo, and R. Kumar, "Independent motion detection in 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 1191–1199, Oct. 2000.
- [38] M. Piacentino, G. van der Wal, and M. Hansen, "Reconfigurable elements for a video pipeline processor," in *Proc. IEEE Symp. Field-Programmable Custom Computing (FCCM99)*, Apr. 1999.
- [39] —, "The ACADIA vision processor," in *Proc. IEEE Int. Workshop on Computer Architecture for Machine Perception*, Padua, Italy, Sept. 2000.
- [40] J. Y. A. Wang and E. H. Adelson, "Layered representation for motion analysis," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1993, pp. 361–366.
- [41] H. Tao, H. S. Sawhney, and R. Kumar, "Dynamic layer representation with applications to tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head, SC, 2000.
- [42] G. Hager and P. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1996, pp. 403–410.
- [43] M. J. Black and Y. Yacoob, "Tracking and recognizing rigid and nonrigid facial motions using local parametric models of image motion," in *Proc. 5th Int. Conf. Computer Vision, ICCV'95*, 1995, pp. 374–381.
- [44] L. Wixson, J. Eledath, M. Hansen, R. Mandelbaum, and D. Mishra, "Image alignment for precise camera fixation and aim," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998.
- [45] R. Kumar, H. S. Sawhney, J. C. Asmuth, A. Pope, and S. Hsu, "Registration of video to geo-referenced imagery," in *Proc. Int. Conf. Pattern Recognition, ICPR'98*, Brisbane, Australia, Aug. 1998.
- [46] R. Kumar, S. Samarasekera, S. Hsu, and K. Hanna, "Registration of highly-oblique and zoomed in aerial video to reference imagery," in *Proc. Int. Conf. Pattern Recognition, Barcelona*, Spain, 2000.
- [47] R. Wildes, D. Hirvonen, S. Hsu, T. Klinedinst, R. Kumar, B. Lehman, B. Matei, and W. Zhao, "Video georegistration: Algorithm and quantitative evaluation," in *Proc. IEEE Int. Conf. Computer Vision*, Vancouver, July 2001.
- [48] "Rapid generation and use of 3D site models to aid imagery analysts/systems performing image exploitation," in *Proc. SPIE Conf. Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*, vol. 1944, Apr. 1993.
- [49] A. J. Heller, M. A. Fischler, R. C. Bolles, C. I. Connolly, R. Wilson, and J. Pearson, "An integrated feasibility demonstration for automatic population of geo-spatial databases," in *1998 Image Understanding Workshop*, Monterey, CA, Nov. 20–23, 1998.
- [50] T. Drummond and R. Cipolla, "Real-time tracking of complex structures for visual serving," in *Proc. Vision Algorithms Workshop in Conjunction with ICCV '99*, 1999, pp. 91–98.
- [51] D. Lowe, "Robust model-based motion tracking through the integration of search and estimation," *Int. J. Comput. Vis.*, vol. 8, no. 2, pp. 113–122, 1992.
- [52] R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vis. Graphics Image Process.*, vol. 60, pp. 313–342, Nov. 1994.
- [53] S. Hsu, S. Samarasekera, R. Kumar, and H. S. Sawhney, "Pose estimation, model refinement and enhanced visualization using video," in *IEEE Proc. Computer Vision and Pattern Recognition*, Hilton Head, SC, 2000, pp. 488–495.
- [54] J. Coughlan, D. Snow, C. English, and A. L. Yuille, "Efficient optimization of a deformable template using dynamic programming," in *IEEE Proc. Computer Vision and Pattern Recognition*, Hilton Head, SC, 2000.
- [55] H. Kollnig and H. H. Nagel, "3D pose estimation by fitting image gradients directly to polyhedral models," in *Proc. IEEE Int. Conf. Computer Vision*, 1995, pp. 569–574.
- [56] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau, "Robust real-time visual tracking using a 2D-3D model-based approach," in *Proc. Int. Conf. Computer Vision*, vol. 1, 1999, pp. 262–268.
- [57] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [58] H. Sawhney, S. Hsu, and R. Kumar, "Robust video mosaicing through topology inference and local to global alignment," in *Proc. Eur. Conf. Computer Vision*, Nov. 1998.
- [59] M. Irani, S. Hsu, and P. Anandan, "Video compression using mosaic representations," *Signal Process. Image Commun.*, vol. 7, pp. 529–552, Nov. 1995.
- [60] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, "Efficient representations of video sequences and their applications," *Signal Process. Image Commun.*, vol. 8, pp. 327–351, 1996.
- [61] A. Pope, R. Kumar, H. Sawhney, and C. Wan, "Video abstraction: Summarizing video content for retrieval and visualization," in *Proc. Asilomar Conf. Signals, Systems and Computers*, 1998.



Rakesh Kumar (Member, IEEE) received the Ph.D. degree in computer science from the University of Massachusetts at Amherst in 1992, the M.S. degree from the State University of New York, Buffalo, in 1985, and the B.Tech. degree from the Indian Institute of Technology, Kanpur, in 1983.

He is currently the head of the Media Vision Group at Sarnoff Corporation, Princeton, NJ. At Sarnoff, he has been directing commercial and government research and development projects in

computer vision with a focus in the areas of immersive telepresence and 3-D modeling from images, image registration, video manipulation, and exploitation. He is an author/co-author of more than 25 technical publications and is a co-inventor of five patents.

Dr. Kumar is an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.



Harpreet Sawhney (Member IEEE) received the Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 1992, focusing on Computer Vision.

He is a Senior Member, Technical Staff in the Vision Technologies Laboratory at Sarnoff Corporation, Princeton, NJ, where he has led R&D in 3-D modeling and manipulation from motion video, video enhancement and indexing, and video mosaicing, under a number of commercial and government programs since 1995. He led

R&D in video annotation and indexing at the IBM Almaden Research Center from 1992 to 1995. He has authored over 40 technical publications, holds five patents, and has a number of patent applications pending.



Supun Samarasekera received the Masters degree from the University of Pennsylvania in 1991.

Since 1997, he has been a Member of Technical Staff at Sarnoff Corporation, Princeton, NJ. He has worked on video enhancement, immersive telepresence, video geo-registration and video surveillance. Prior to Sarnoff, he worked at the Siemens Corporate Research Center from 1995 to 1997 and the University of Pennsylvania Medical Image Processing Group from 1991 to 1995. He is a co-inventor of six

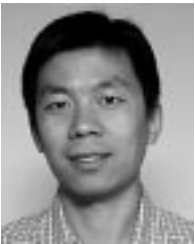
U.S. patents in the area of medical imaging.



Steve Hsu (Member, IEEE) received the B.S. degree from Caltech in 1982 and the Ph.D. degree from the Massachusetts Institute of Technology in 1988, all in electrical engineering.

Since 1988, he has been a Member of Technical Staff at Sarnoff Corporation, Princeton, NJ. His research interests have included image restoration, object recognition, compression, motion analysis, mosaicing, and 3-D scene reconstruction.

Dr. Hsu was awarded graduate fellowships from the National Science Foundation and from Bell Laboratories.



Hai Tao (Member IEEE) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1991 and 1993, respectively. He received the M.S. degree from Mississippi State University in 1994 and the Ph.D. degree from the University of Illinois at Urbana-Champaign in 1998, both in electrical engineering.

Since November 1998, he has been a Member of Technical Staff at the Vision Technology Laboratory of Sarnoff Corporation, Princeton, NJ. His research interests are in image and video processing, computer vision, computer graphics, and pattern recognition. He is the senior author of 30 technical papers and one book chapter.



Yanlin Guo (Member IEEE) received the B.S. degree in Electronics Engineering from Tsinghua University, Beijing, China, in 1993, the M.S. degree in electronics engineering from Tsinghua University, Beijing, China, in 1995, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, in 1998.

In 1998, she joined the Sarnoff Corporation, Princeton, NJ, where she is a Member of Technical Staff. Her main areas of interest are stereo and motion analysis, video and image enhancement, shape modeling, and analysis and medical imaging analysis.



Keith Hanna received the B.A. degree in engineering sciences and the D.Phil. degree in medical computer vision from Oxford University in 1986 and 1990, respectively.

He is a Senior Member, Technical Staff at the Vision Technologies Laboratory at Sarnoff Corporation, Princeton, NJ. He has led many commercial and government R&D programs in motion analysis and image alignment, video enhancement, pattern recognition, and embedded system implementation. He has led several programs that resulted in the formation of new companies and has worked for extended periods at several of these new companies. He has authored numerous technical publications and holds 11 patents.



Arthur Pope received the S.M. degree from Harvard University in 1989 and the Ph.D. degree in computer science with a concentration in computer vision from the University of British Columbia in 1995.

From 1983 until 1990, he was at BBN, Cambridge, MA, developing technologies for distributed interactive simulation. Since 1996, he has been with Sarnoff Corporation, Princeton, NJ, where he has led several projects to develop techniques and systems for processing aerial video.



Richard Wildes (Member, IEEE) received the Ph.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 1989.

In 1984–1988, he was a Research Assistant in the MIT Artificial Intelligence Laboratory. During that time, he was a National Science Foundation Graduate Fellow. In 1988, he joined the Department of Computer Science at the State University of New York at Buffalo as an Assistant Professor. During 1990, he joined the Sarnoff Corporation, Princeton, NJ, where

he is a Member of Technical Staff. His main areas of research interest are machine and biological perception (especially vision), robotics and artificial intelligence.

Dr. Wildes received the 1999 IEEE D. G. Fink Prize Paper award for his work on automated iris recognition.



David Hirvonen received the B.S. degree (with honors) in computer science from the University of Massachusetts (UMASS), Amherst, in 1998.

From 1996 to 1998, he was a Research Assistant with the UMASS Multi-media Indexing and Retrieval Group. During 1998, he joined the Vision Technologies Laboratory at Sarnoff Corporation, Princeton, NJ, where he has worked on improving image and video registration techniques. His current focus is medical imaging.



Michael Hansen received the B.S. and M.S. degrees in electrical engineering from the Pennsylvania State University in 1991 and 1993, respectively.

He is currently head of the Advanced Video Processing research group at Sarnoff Corporation, Princeton, NJ. His work and research interests involve real-time, embedded applications of computer vision including video enhancement, image registration, motion estimation, 3-D recovery from motion and stereo, and

other areas. He has also been active in the development of video processing hardware systems and integrated circuits, including Sarnoff's first vision system on a chip and several commercial products that use real-time vision algorithms and hardware.



Peter Burt (Member, IEEE) received the B.A. degree in physics from Harvard University in 1968 and the Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 1976.

He is currently the managing director of the Vision Technology Business Unit at Sarnoff Corporation, Princeton, NJ. From 1968 to 1972, he conducted research in sonar, particularly in acoustic imaging devices, at the U.S. Navy Underwater Systems Center, New London, CT. As a Postdoctoral Fellow, he has studied both natural and computer vision at New York

University (1976–78), Bell Laboratories (1978–79), and the University of Maryland (1979–1980). He was a member of the engineering faculty at Rensselaer Polytechnic Institute, Troy, NY, from 1980 to 1983. In 1983, he joined the David Sarnoff Research Center, where he has led the computer vision activities since 1984. He is active in the development of fast algorithms and computing architectures for real-time computer vision.