RSTA on the Move: Detection and Tracking of Moving Objects from an Autonomous Mobile Platform*

Larry S. Davis
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

Ruzena Bajcsy
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

Martin Herman
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

Randal Nelson
Department of Computer Science
University of Rochester
Rochester, NY 14627

Abstract

This report describes progress made during the past year on the UGV RSTA project being conducted by a consortium led by the University of Maryland and including the University of Pennsylvania, the University of Rochester, and the National Institute of Standards and Technology. We first review work done on the design, implementation and integration of real time vision algorithms for image stabilization, detection of moving objects from a moving platform and camera control. We then present brief descriptions of a number of supporting basic research projects being conducted by the members of the consortium.

1 Introduction

Our **RSTA on the Move** project is a program combining

- development and integration activities ultimately leading to an experimental, real time active vision system for locating and tracking moving targets from a mobile platform, and
- basic research on fundamental active vision problems including motion estimation, image sequence stabilization, detection and characterization of independent motion patterns, and real-time sensor control.

Our integration activities involve algorithm development and integration on the Datacube real time image processing platform, and experimentation on video sequences obtained, initially, offline from a sensor mounted on a HMMWV at NIST, and, ultimately, online using the same NIST platform with onboard real-time and parallel processing. Section 2 describes our progress on development and integration.

Real-time algorithms for image stabilization and moving object detection have been developed at Maryland, while Rochester has continued development on a more general real-time algorithm for detection of independently moving objects. Both the Maryland stabilization algorithm and the Rochester independent motion detector have been transferred to the NIST Datacube, and were demonstrated last summer at Martin Denver. Ongoing work involves the integration of these two algorithms on a common Datacube/SPARC platform, and design and implementation of spatio-temporal grouping algorithms for focusing attention of the active vision component of our system on an image window containing an independently moving object. Research at the University of Pennsylvania has emphasized camera control algorithms that will allow us to track the moving target and to maintain as large an image of the target as possible through control of a zoom lens. Some core camera control software has already been ported to the NIST platform, with NIST and Pennsylvania now collaborating on the design and implementation of the full camera control subsystem. Figure 1 shows the tasks of the individual contractors.

In addition to ongoing development and integration activities, the consortium supports a broad spectrum of fundamental research activities in time-varying image analysis and active vision. A set of research vignettes are presented in Section 3, including descriptions of research projects on motion estimation, image stabilization, comparison of image stabilization algorithms in the context of a real-time target acquisition and tracking system (joint research with the Army Research Laboratory), and camera control.

*The support of the Advanced Research Projects Agency (ARPA Order No. A422) and the Army Research Office under Grant DAAH04-93-G-0419 is gratefully acknowledged.

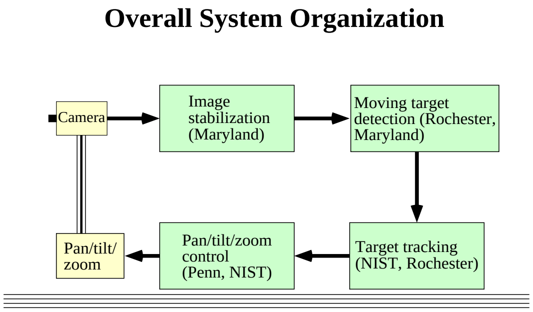

Overall System Organization
Camera
Image stabilization (Maryland)
Moving target detection (Rochester, Maryland)
Target tracking (NIST, Rochester)
Pan/tilt/zoom control (Penn, NIST)
Pan/tilt/ zoom

Figure 1: Block diagram of system architecture.

2 Integration Activities

2.1 University of Maryland

(Yiannis Aloimonos, Stephen Balakirsky, Rama Chellappa, Loong Fah Cheong, Cornelia Fermüller, Carlos Morimoto, Yi-Sheng Yao)

Research at Maryland emphasizes image stabilization, with some supporting research on detection of constrained (vehicle-like) independent motion. The goal of our image stabilization process is to maintain a stable scene background in a video image sequence. This is accomplished by estimating and compensating for the effects of the movement of the vehicle on the original input image sequence, so that in the resulting stabilized sequence the background of the scene appears, ideally, as if the vehicle were stationary.

After the sequence is stabilized, independently moving objects can be detected using either the flow-based approach being developed at the University of Rochester, or a frame-differencing approach developed by the University of Maryland. The Maryland approach is based on an efficient algorithm for computing a temporal median filter from the stabilized sequence. To optimize for detection of independent vehicle motion, we employ a filtering approach that integrates the results of velocity-tuned filters over several frames and produces the final output of the system. Details of the Maryland work are given below.

2.1.1 2D Image Stabilization

The 2D image stabilization algorithm is described in [Davis et al., 1994]; it uses the camera model presented in [Zheng and Chellappa, 1993]. The model decomposes the movement of the camera into four components: translation along x, translation along y, rotation around z (the optical axis), and scaling due to translation along z. (x, y, z) define a coordinate system centered at the camera and $(x, y, 1)$ define the image plane. The transformation between two frames under this motion model is

$$\begin{pmatrix} X_2 \\ Y_2 \end{pmatrix} = \mathrm{s} \begin{pmatrix} \cos\Theta & \sin\Theta \\ -\sin\Theta & \cos\Theta \end{pmatrix} \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} + \begin{pmatrix} \Delta X_2 \\ \Delta Y_2 \end{pmatrix} \quad (1)$$

where (X_i, Y_i) are the image frame coordinates at time t_i for $i = \{1, 2\}$; $(\Delta X_2, \Delta Y_2)$ is the translation measured in the image coordinate system of frame t_2; Θ is the rotation angle between the two frames; and s is the scaling factor.

The camera motion parameters are estimated by matching a small number of feature points between two frames. Given that N points are tracked, we first determine the scaling factor s based on the fact that the ratio of the distances between two arbitrary points measured in both frames is proportional to s. Then equation 1 is used to construct a $2N$-equation linear system that is solved using a least-square approach to compute the remaining translation and rotation parameters.

The stabilization process consists of computing the motion parameters between two consecutive frames f_{i-1} and f_i, composing all the transformations from a reference frame f_0 up to time t_i, and then warping frame f_i using the combined motion parameters. A hardware implementation of the system is described in [Morimoto et al., 1995]. This system was developed to use a commercially available parallel pipeline image processing board (Datacube MV200) connected to a SUN SPARCstation 20/612, and is able to process 7 frames per second, using images of size 128×128. A very similar scheme is described in [Burt and Anandan, 1994], where more specialized image processing hardware is used to stabilize images by registering frames using a hierarchical approach.

2.1.2 Detection of Independently Moving Objects

Image stabilization renders the background of the image approximately stationary. In order to overcome the effects of residual motions, we implemented a *temporal median filter*. This filter creates a *median image* that is composed of the median values of the last k frames. In a sequence of pixel gray levels from k frames, the gray levels arising from an independently moving object tend to be outliers with respect to the median of the sequence, so that they at least partially disappear from the median image. A simple image differencing scheme can then be applied to detect independent motion on a frame-by-frame basis.

A *filtered image* is a binary image obtained by thresholding the difference between the median image and the current frame. This process tends to erase the background and highlight the locations of independently moving objects. These filtered images still contain noise (due to imperfect stabilization) and spots corresponding to close scene objects that appear to move due to motion parallax. Velocity-tuned filters are used to reject the stabilization noise; motion parallax spots are also rejected if their apparent motion in the image is out of the velocity range for which the filter is tuned.

Assume that a filtered image f_i contains several spots, and we want to select only those that move linearly at a rate of p pixels per processed frame. If such spots also appeared in the previous frame, f_{i-1}, by the time frame f_i is captured these spots must have moved p pixels away from their f_{i-1} positions and are therefore located in frame f_i somewhere on *circles*, p pixels in radius, centered at their f_{i-1} positions. If we select the pixels that correspond to the intersections between the spots of f_i and the spots generated by replacing the spots of f_{i-1} by the appropriate circles (predicted positions), we obtain good candidates for regions moving at p pixels per frame. This scheme can be extended to include more than two frames, since spots in frame f_{i-2} should be $2p$ pixels away, and spots in frame f_{i-j} should be jp pixels away. Implementation issues regarding these filters are given in [Morimoto et al., 1995].

2.1.3 Experimental Results

Figure 2 shows a frame of a video sequence taken from a moving vehicle and Figure 3 shows the thresholded difference between the four-frame temporal median and the stabilized instance of that frame. Finally, Figure 4 shows the stabilized frame superimposed on the output of the velocity-tuned filters integrated over four frames.

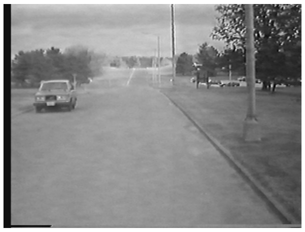


Figure 2: Frame from video sequence.

2.2 University of Rochester

(Randal Nelson, Rajesh P.N. Rao)

Current work at the University of Rochester addresses the RSTA goal of detecting and tracking independently moving objects from a moving platform. The detection of independently moving objects is a critical task for RSTA subsystems. Because objects that move independently represent possible threats, it is important to flag them as rapidly as possible and then track them so that identification systems can be brought to bear on them. We have been engaged in a project whose goal is to design, implement, and test a general framework for utilizing visual motion for the detection and recognition of events and objects. Overall, we have been developing a three-step process for motion recognition that includes detection, tracking, and recognition phases. Of these steps, the detection and tracking components are of most immediate interest to the RSTA demos, and we have been engaged in porting previously developed algorithms for these aspects of the problem to hardware on the NIST vehicle, and evaluating the algorithms under various field conditions. The hardware and mechanics on the NIST vehicle are consistent with those of the current RSTA vehicles, and thus provide a valid testbed relative to RSTA demo goals.

The fast detection of potentially significant motion events is based on identifying violations of qualitative rigid-world constraints. This provides a uniformly applicable strategy by which small regions of the scene can be selected for more thorough inspection. In previous work, we produced real-time algorithms for detecting independently moving objects from a moving platform [Nelson, 1991]. These techniques are more general than those based on affine stabilization of the visual field, and can function in situations containing substantial motion paral-

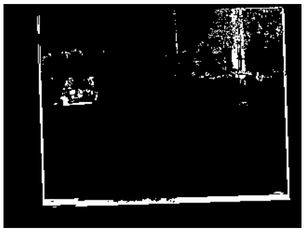


Figure 3: Thresholded difference between temporal median and stabilized frame.

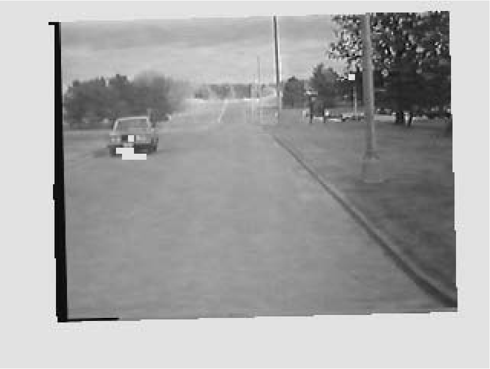


Figure 4: Stabilized frame with superimposed moving object regions.

lax at different depths, and skewed, non-planar radial flow (such as that produced in the near field during locomotion through hilly terrain), which cause problems for the affine methods. They can thus serve to augment affine stabilization algorithms, which have previously demonstrated their value in regimes where they are valid. We have recently ported these algorithms to a platform consistent with the hardware on the RSTA vehicles, and are engaged in evaluating their performance, both in isolation and in combination with low-level stabilization algorithms developed at Maryland.

The tracking step involves stabilization of the area of interest through active visual processes such as fixation and tracking to place the motion of interest in a canonical form that facilitates the final recognition procedure. The techniques of most immediate interest for RSTA involve the tracking of independently moving rigid objects. We are currently developing real-time algorithms for instantiating and maintaining hypotheses about the positions, extents, and motions of such objects on the basis of the output from the independent motion detection system. We have also recently developed techniques for accomplishing this for objects that move in a complex manner, such as people or animals [Polana and Nelson, 1994].

The identification step locates regions of interest via a more detailed analysis of motion. We are developing techniques based on *temporal texture analysis*, where we extract statistical spatial and temporal features from approximations to the motion field and use techniques analogous to those developed for gray-scale texture analysis to classify regional activities. Some results in this area are described in [Nelson and Polana, 1992]. In a second approach, which we term *activity recognition*, we use the spatial and temporal arrangement of motion features in conjunction with simple geometric image analysis to identify complexly moving objects such as machinery and locomoting people and animals [Polana and Nelson, 1993]. The remainder of this section concentrates on the motion detection processes.

Detection of moving objects is of critical importance to biological and robotic systems, both because such objects are frequently of primary interest to the system, and because dealing with them involves hard real-time constraints—the world won't wait while you think. A method of detecting independent motion, or motion having certain other qualitative characteristics such as periodicity, is thus valuable as a method for directing more sophisticated (and costly) processing to areas where it can be most effectively utilized.

In previous work, we developed methods for detecting three qualitative types of motion. The first technique, which we term **constraint ray filtering**, provides a robust method of detecting independently moving objects from a moving platform when information is available about the platform motion [Nelson, 1991]. The method is based on the observation that the projected motion at any point on the image sphere is constrained to lie on a half line (ray) in local velocity space whose parameters depend only on the observer's motion and the location of the image point. The second method, termed *animate motion detection*, allows rapid detection of animate objects with no information about the movement of the platform [Nelson, 1991]. It is based on the observation that animate moving objects typically *maneuver*, that is, they or their component parts follow trajectories for which the projected velocity changes rapidly compared to the velocity change due to self-motion. The third method allows detection and tracking of objects whose motion has a periodic component, such as walking or running animals, oscillating machinery, etc. [Polana and Nelson, 1994]. It is based on a Fourier transform technique.

These techniques can be used to isolate motion for identification by later recognition processes. The responses of the different qualitative detectors yield an indication of the sort of recognition process that should be assigned to the movement of interest. For example, detection of local, highly periodic movement would suggest the use of a phase-based structural classifier, while a distributed, non-periodic motion would suggest the use of temporal texture techniques.

The first two techniques were originally implemented as real-time systems on Datacube series 10 hardware, and demonstrated in a laboratory setting. Of these, the first, constraint ray filtering, is of the most immediate interest to the RSTA goal of detecting moving vehicles. We have ported this algorithm to hardware on the NIST vehicle, which is compatible with the hardware on the Martin Denver demo vehicles, and begun evaluation of the algorithm using outdoor driving sequences acquired from both the NIST vehicle and other vehicles.

Representative results of the moving object detection algorithm are illustrated in Figures 5 a–b. Figure 5a shows a frame in a video sequence acquired from a forward moving mobile platform. The independently moving objects are the cars moving left to right near the horizon. Figure 5b shows the superposition of the pixels detected by the independent motion detector onto the video frame. These pixels lie on the car moving across the image.

Theoretical analysis of the algorithm indicates

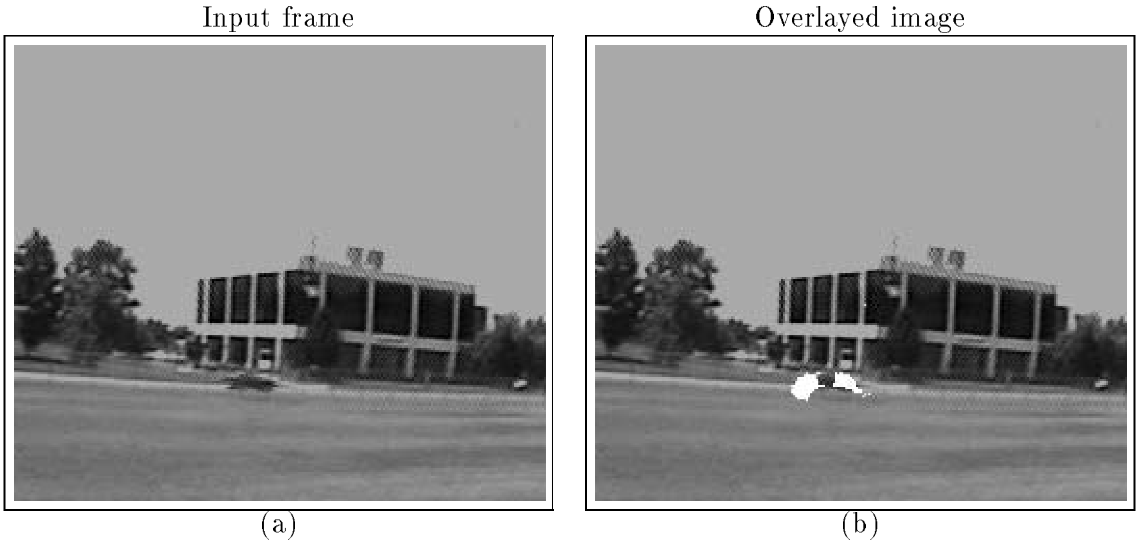


Figure 5: (a) Original image. (b) Independently moving pixels overlaid on (a).

that, although the underlying technique is capable of detecting independent motion in an arbitrary environment under arbitrary motion, the instantiation of the algorithm on the Maxvideo hardware has certain limitations. In particular, the choice of a first-order gradient-based flow estimation technique is dictated by the operations that the Maxvideo performs efficiently (namely, convolution), and this limits the accuracy of the motion field estimation. Further analysis indicates that when the magnitude of the motion due to vehicle movement exceeds the magnitude of the independent motion by more than a factor of two, detection is unreliable. We can recognize this situation and avoid false positives, but genuine independent motion may then go undetected.

The theoretical performance is borne out by field tests. When driving on roads, or slowly on relatively smooth terrain, the algorithm detected other moving vehicles in a variety of situations. However, at high velocity, and over rough off-road terrain, the detection limit is frequently exceeded, and independently moving objects are missed. Further analysis revealed that this effect is primarily due to rapidly changing vehicle pitch, with smaller effects from roll and yaw.

Two approaches can be used to resolve this situation. The first is to use a more accurate motion estimation algorithm. However, this is probably not practical in real time using the existing hardware, though we are exploring the use of a multi-resolution gradient-based algorithm. (Advances in hardware may change this picture for the next generation of vehicles, but for the moment we are limited to the current Maxvideo system). The second approach notes that the dominant source of large motions that swamp the detection algorithm is vehicle rotation, which is removable by stabilization techniques of a sort that have already been demonstrated.

We are currently engaged in instantiating the second approach, using several stabilization algorithms developed at the University of Maryland as pre-processors to the motion detection system. The different algorithms will be compared, and the best of them selected for our demonstration. We are also currently developing a predictive tracker that will use the (pixel map) output of the independent motion detection system to circumscribe and track potential target objects. These regions of interest will ultimately serve as inputs to the next phase of the system where recognition and higher-level planning are performed.

2.3 University of Pennsylvania

(Ruzena Bajcsy, Ulf Cahn von Seelen)

The University of Pennsylvania's research is concerned with camera control for tracking acquired targets. The controlled axes include mechanical degrees of freedom (pan, tilt) as well as an optical degree (zoom). The hardware platform consists of a TRC BiSight binocular camera platform controlled by a PMAC-VME motion controller that is connected to a Sun workstation via shared memory.

While object tracking by panning and tilting a camera is well known, the use of zoom in tracking is largely unexplored. For RSTA on the Move we want to maximize the spatial resolution of the tracked object while maintaining acquisition. This involves optimizing the tradeoff between spatial resolution and tracking performance. The closer the camera zooms in on the target, the faster the target moves in the image, and the harder it becomes to maintain acquisition of it.

To our knowledge, there exist only a few publications that deal with the control of zoom for tracking. In [Hwang et al., 1993] the zoom is used to achieve a desired object image size. A fuzzy controller combines estimates of the diagonal extent of the object, the variance of the object velocity, and the confidence of the shape estimate to compute a suitable focal length. The influence of the velocity variance ensures that the camera does not zoom in too closely if an object's motion varies greatly, in order to safely maintain acquisition.

In [Hosoda et al., 1995] the authors use a robot arm and camera zoom to achieve a desired image feature configuration. The focus of the work is on integrating the zoom into the control as a redundant mechanical degree of freedom, as the authors assume that the image Jacobian and thus the 3D positions of the image features in the world are known. This assumption abstracts from the main problem of using zoom in tracking, namely finding an image-based measure on which to servo the focal length.

In the *PennEyes* system [Madden and Cahn von Seelen, 1995] we have used various image-based measures to maintain the apparent size of a target in an image. In the current version we use cross-correlation to identify the target. This approach is more general, but it does not provide a ready estimate of the apparent target size. We work with the object distance instead, which we estimate by triangulation from the two camera views. Using our calibration of the zoom lens, we can compute a new focal length when the target distance changes so that the image size of the target remains constant. Figure 6 shows a typical run of the system in which the focal length is increased so that it compensates for the target motion away from the camera head.

With the expertise gained from zooming for size constancy, we can approach the problem of zooming for scale change. Changing scale poses increased demands on target identification and localization because commonly used approaches such as cross-correlation are not scale-invariant. Alternatives include feature-based tracking (e.g. [Reid and Murray, 1993]) or the use of adaptive correlation templates (e.g. [Parry et al., 1995]). We are currently investigating the latter approach.

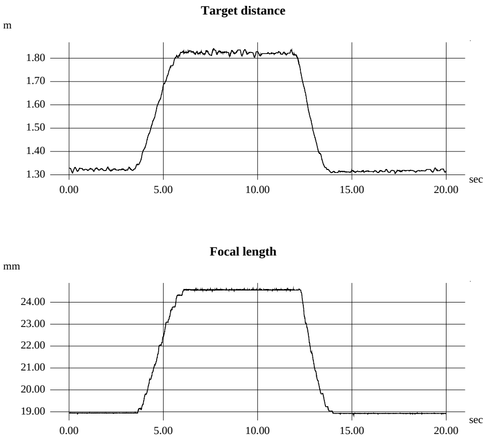


Figure 6: Focal length responding to changing target distance.

2.4 NIST

(Martin Herman, David Coombs, Sandor Szabo, Tsai-Hong Hong)

NIST is responsible for developing the vision processing platform, assisting in integrating University software onto the platform, and running the platform on vehicles at the NIST facility. In addition, NIST has collected video data using the NIST HMMWV, and is working on target tracking and gaze control software. NIST has completed development of the platform and has worked with the Universities to demonstrate two components of RSTA on the Move: image stabilization and independent motion detection.

The NIST vision processing platform is based on industry-standard components which allow us to integrate, test and distribute results with minimal amounts of effort. For example, approximately one hour was required to initially install and run software from each University. This allows us to spend a considerably greater portion of our time in analyzing and improving system performance. The platform, designed for mobile applications, was easily shipped, set up and demonstrated at ARPA's UGV Demo C in Denver in July 1995. Since then we have completed the power conversion of the system so that we can now run from vehicle DC power sources as well as conventional AC sources. We are close to upgrading our computing system to three processors and to Solaris 2.4 which will allow us to take full advantage of symmetric multiprocessing and real-time scheduling. Our design, integrating the RSTA software components and taking full advantage of multiple general purpose processors, and multiple specialized image processors is almost complete. By early Spring of 1996 we will be able to perform experiments on the NIST HMMWV on a regular basis. Because of our close involvement with the ARPA Demo II program and the Army Research Laboratory, we feel that the results of our effort can be readily integrated into future DoD mobile robot applications.

The NIST vision processing system, designed for RSTA on the Move applications, is capable of performing sophisticated experiments in mobile vehicle applications. The system consists of a TRC UniSight/BiSight camera head (pan/tilt/vergence), a Datacube MV200 image processor for low level image processing (acquisition, filtering, overlays, etc.), and a suite of fast SPARC processors (for motion analysis, tracking, etc.). Both the image stabilization algorithm from the University of Maryland and the independent motion detection algorithm from the University of Rochester rely on image processing taking place in both the specialized Datacube environment and the general purpose SPARC environment. The University of Pennsylvania tracking software relies on the specialized motion controller for the camera head (a Delta Tau PMAC motion controller board) and the SPARC environment.

The system components of the vision processing platform consist of a VME-based card cage housing all the processor boards, a two gigabyte ruggedized hard disk, and electronics for the cameras and the TRC camera head. The VME provides power and fast communications between a Themis SPARC 10MP processor board, the Datacube MV200 image processor board, and the Delta Tau PMAC motion controller board.

The Themis board is outfitted with an 80 MHz and two 90 MHz HyperSparc processors. Our plan is to dedicate one processor to image stabilization, one to independent motion detection, and one to tracking. The processors will be run in pipeline mode with the results from one stage being fed to the next stage. In the future we hope to add additional parallelization within stages to reduce the overall latency. To support this work we are in the process of changing from Solaris 1.1 to Solaris 2.4. After the change we will be able to take advantage of the multithreading libraries and the real-time scheduling facilities. Changing to a complete Solaris system will hopefully allow us to migrate away from having separate operating systems for development and real-time applications, thus further simplifying integration. All of the code is written in C/C++ and makes use of the GNU Free Software Foundation environment. We have installed and tested software from the University of Rochester and the University of Maryland under this environment without any problems.

The Datacube MV200 is practically the industry de-facto standard for real-time vision processing. We have installed a complete programming environment for the MV200: Imageflow, Advanced Imaging Tools, WitFlow, and Veil. We have also installed a miniwarper. Both the University of Rochester and the University of Maryland algorithms require the MV200. Currently we have only run the algorithms one at a time. Our plan is to pipeline the algorithms using two MV200 boards, one of which we will borrow from the University of Maryland during the experiments.

The Delta Tau PMAC motion controller board is used for control of the TRC head. NIST has experience in this area, having built a head (TRICLOPS) in the past, and we are planning to incorporate previously developed head control software into the RSTA application. We have received software from the University of Pennsylvania for controlling the head at a low level and have also developed our own software for computing quintic-based smooth trajectories.

The NIST system is completely self-contained. Each of the components is designed for modularity, having its own dedicated power conditioner to run off DC power sources. Sufficient power exists to run a fully configured four-processor SPARC 10MP, three MV200's, the motion controller board, and an additional I/O processor designed for a potential vestibular sensor system. All the components are housed in a sealed, ruggedized enclosure designed for outdoor vehicles. NIST, with the support of the Army Research Laboratory, also maintains a fully robotic HMMWV which enables us to perform experiments on a regular basis.

Data Collection. NIST has collected over six hours of videotape from color CCD cameras rigidly mounted on the HMMWV, driving at up to 40 kph on- and off-road at the NIST site in Gaithersburg, MD between December 1993 and June 1995. The terrain includes campus roads, fields and woods. Civilian vehicles can be seen driving on the NIST grounds and on the surrounding roads (including highway I-270) at ranges up to 2000 m. Pedestrians and deer are also visible on occasion. The cameras are rigidly mounted on the vehicle in forward-looking and oblique-looking (60 degrees off heading) orientations. The lens focal lengths used range from 5 mm to 75 mm. No stabilization was used (neither mechanical stabilization of the cameras nor digital stabilization of the video) and image jitter is particularly noticeable with longer focal-length lenses.

3 Supporting Basic Research

In addition to the integration activities outlined in the previous section, each of the consortium

Table 1: Detection results of three stabilization algorithms.

Algorithm	Threshold	% targets detected	# frames to acq. target	Avg. false alarms/frame	% targets segmented
Projection	17	0	NA	NA	NA
FTA 1	12	0	NA	NA	NA
FTA 2	12	100	7	1	67

members is also pursuing a program of basic research on enabling technologies for RSTA on the Move. In this section we provide brief descriptions of some of these research projects.

3.1 Performance Characterization of Image Stabilization Algorithms—University of Maryland

We have carried out a comparative study of image stabilization algorithms in the context of an automatic target tracking system. This study was conducted jointly with the Army Research Laboratory (ARL). The goal is to perform target acquisition through a process of background suppression and motion estimation. In order to accomplish this it is important that the input sequence be stabilized so that image motion due to camera motion as the camera is panned, or as the camera moves through the scene, is compensated for.

Three stabilization algorithms were compared with respect to target false alarm and false dismissal rates, time to acquisition of targets, and a gross measure of the accuracy of target segmentation.

- The first algorithm was developed at ARL to compensate for wind loading on an unmanned robotic platform. It is a simple algorithm that can only estimate integer image translations, and operates on normalized row and column projections of consecutive frames in the video sequence.
- The second algorithm was developed at the University of Maryland; it is a multiresolution version of the algorithm described in Section 2 of this report.
- The third algorithm was a generalization of the second one; it uses longer image sequences for motion estimation.

In spite of the fact that the stabilized image sequences obtained from the three algorithms were perceptually almost indistinguishable, there were dramatic differences in performance between the first two algorithms and the third. The first two algorithms had unacceptably high false dismissal and false alarm rates on the tested image sequences. On the other hand, when the target tracker was integrated with the third algorithm, it achieved a 0% false dismissal rate and a 1% false alarm rate on real IR sequences. Details of this study will be reported in [Balakirsky and Chellappa, 1996].

Figure 7 shows a typical image from one of the real FLIR sequences employed in the experiments. The target, near the top of the image, is outlined in a box. Table 1 compares the detection results of the three stabilization algorithms on one of the FLIR sequences. Again, even though there is little perceptual difference between the stabilized sequences produced by the three algorithms, the impact of the small differences on target acquisition and false alarm detection rates were quite significant.

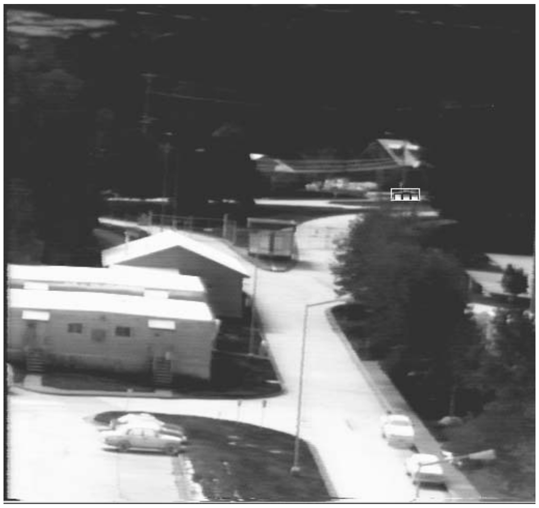


Figure 7: Typical image from a real FLIR sequence.

3.2 3D Model-Based Image Stabilization—University of Maryland

We have studied the use of combined visual cues and dynamic models for the stabilization of calibrated or uncalibrated image sequences [Yao et al., 1996].

Parameters relevant to image warping are estimated by combining information from different tracked tokens, namely points and horizon lines. These parameters are simply the camera rotational velocity if intrinsic camera parameters are available, or the projectivity coefficients, in the uncalibrated case. Image plane displacements of distant feature points may unambiguously characterize rotational motion. However, such points are sometimes difficult to detect and track, due to the absence of sufficient intensity gradient information. Horizon lines, when present, on the other hand, constitute very strong visual cues, requiring relatively simple operations for their tracking. These tokens are therefore both used in our stabilization scheme.

We have investigated how to use temporal information in a sequence to facilitate the estimation of parameters of interest. Image stabilization is a process closely related but not equivalent to image registration. Registration techniques can be extended for stabilization purposes. Image stabilization is inherently different in that it allows the use of dynamical information over long temporal windows. In unmanned ground vehicle application, cameras are mounted rigidly on the platform. The rotation of the vehicle arises from the rotational movement of the vehicle. It is therefore possible to employ a kinetic law which captures the rotation of the platform to model the temporal behavior of the parameters of interest. However, with the aid of visual cues, simple kinematic laws become feasible. We therefore use a kinematic law to model the temporal behavior of relevant parameters.

Specifically, for calibrated sequences, since the intrinsic parameters of the camera are known, the perspective projection model which describes the relationship between 3D scenes and their 2D projections can be used to characterize the projection of both distant points and horizon lines. After the points and horizon lines are tracked over the sequence, they can be used along with a kinematic law to estimate the rotational parameters. Based on the estimated parameters, a stabilized sequence is generated.

For uncalibrated sequences, to integrate distant points and horizon lines, a different description of the movement of horizon lines is employed. This leads to the estimation of eight projective coefficients, in order to stabilize the uncalibrated sequence. However, the estimates of these projective coefficients are very sensitive to the tracking of points and lines. On the other hand, the intrinsic parameters are often approximately known. Instead of estimating the eight projective coefficients, our uncalibrated stabilization scheme is then similar to the calibrated scheme and concentrates on estimating the three rotational parameters while assuming the approximate intrinsic parameters.

Both schemes have been tested on real sequences with good results. The results of this research are illustrated in Figures 8(a–d). Figure 8a shows a sample image from an outdoor sequence. Figure 8b is the same image with the horizon line superimposed. Figure 8c shows the point trajectories for features in that sequence and Figure 8d is a plot of the estimated 3D rotational parameters.

3.3 Perception of the UGV's Environment—University of Maryland

Our work on RSTA on the Move has concentrated on the interplay between the recovery of three-dimensional motion information and the recovery of descriptions of the immediate environment of the UGV. Regarding the perception of 3D motion we have implemented a set of techniques that recognize a collection of global patterns of robust spatiotemporal measurements. The localization of these patterns, which are independent of the structure of the scene in view, encodes the underlying 3D motion parameters, enabling stabilization through the interpolation of the UGV's intended motion in the temporal evolution of the measured motion. Figure 9 shows experiments using the approach developed in [Fermüller and Aloimonos, 1995] with real data collected from the vehicle.

A recent technical development related to the perception of the UGV's environment is the concept of iso-distortion surfaces, a framework for studying the relationship between the computation of 3D motion and depth from a sequence of images [Cheong and Aloimonos, 1995]. The underlying conceptual theme is that motion errors (e.g., errors between retinal motion and perceived 3D motion) affect depth estimates systematically. The understanding of the geometry of this distortion of depth is essential for understanding the interplay between 3D motion and shape processing and thus for interpreting visual motion. The introduced framework characterizes this relationship via a family of iso-distortion contours, which describes the loci over which depths are distorted by the same amounts. Figure 10 shows an example of the iso-distortion contours that result from intersecting the iso-distortion surfaces with the Zx-plane.

The tool of iso-distortion surfaces allows us to study the very practical problem of calculating the precision of an inertial system that is sufficient for obtaining unbiased estimates of the vehicle's heading direction, using algorithms that combine inertial and visual measurements. The process is explained in Figure 11.

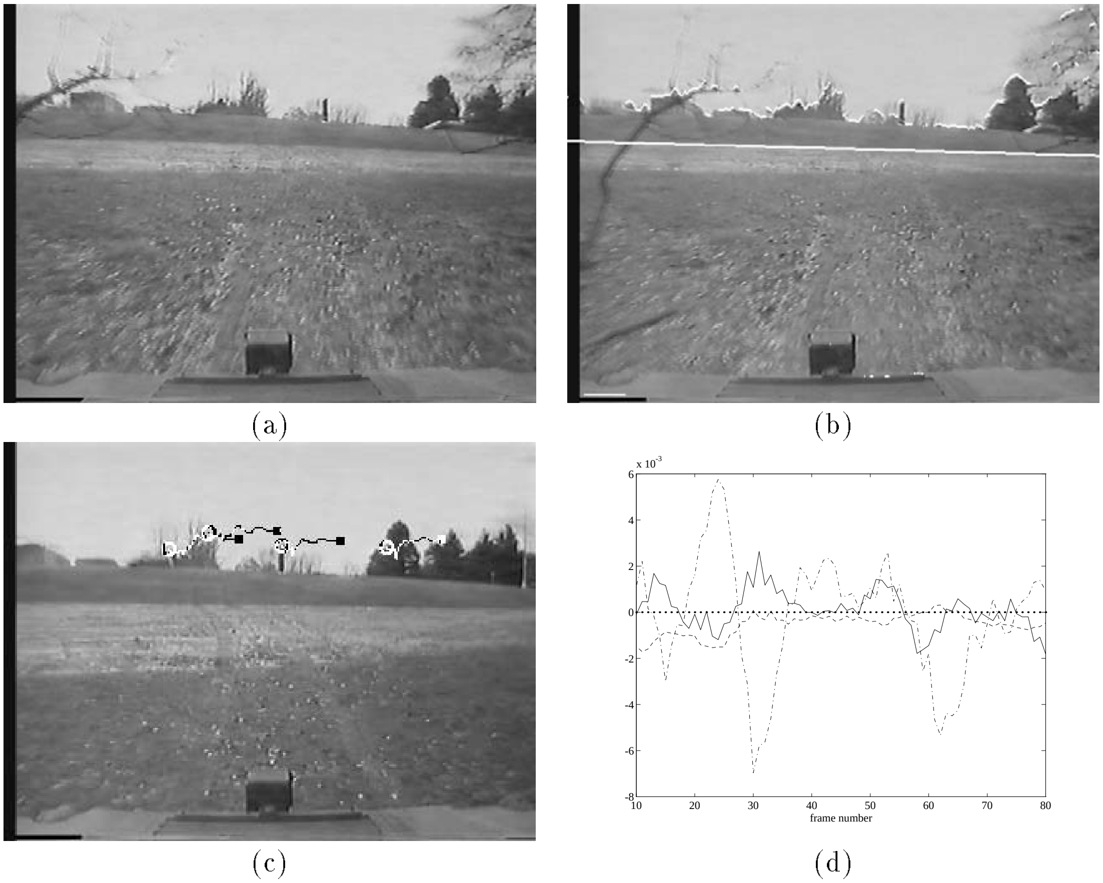


Figure 8: (a) A sample image from a sequence, (b) an image with horizon line detected, (c) an image with point trajectories, and (d) a plot showing the estimated 3D rotational parameters

3.4 Fast, Filter-Based, Object Location and Identification—University of Rochester

We have developed a visual location and identification system [Rao and Ballard, 1995b] based on efficiently computable iconic representations. The system uses two primary visual routines, one for identifying the visual image near the fovea (*object identification*), and another for locating a stored prototype on the retina (*object location*). The iconic representations are based on high-dimensional feature vectors obtained from the responses of an ensemble of *steerable Gaussian derivative spatial filters* at a number of orientations and scales. Such feature vectors serve as effective photometric descriptions of the local intensity variations present in the image region about a scene/object point; in addition, they can be made rotation and scale invariant [Rao and Ballard, 1995b]. The iconic feature vectors are stored in two separate memories. One memory is indexed by image coordinates while the other is indexed by object coordinates. Object location matches a localized set of model features with image features at all possible retinal locations. Object identification matches a foveal set of image features with all possible model features.

We describe here in more detail the routine for object location; details regarding the identification routine, which employs Kalman Filter theory and visual learning, can be found in [Rao and Ballard, 1995a]. The location routine crucially depends on the fact that only a single model object is being matched to other objects in an image at any instant. Let us denote this model that is to be located in the current image as

$$M = \{\mathbf{r}^m, m = 1, \ldots, m_{\max}\}. \qquad (2)$$

where $\mathbf{r}^m$ are the object's filter response vectors extracted from different spatial locations.

The location algorithm in its most general form proceeds as follows:

1. For each response vector $\mathbf{r}^m$ representing some model point m, create a *Saliency Image* S_m defined by

$$S_m(x, y) = \|\mathbf{r}(x, y) - \mathbf{r}^m\|^2 \qquad (3)$$

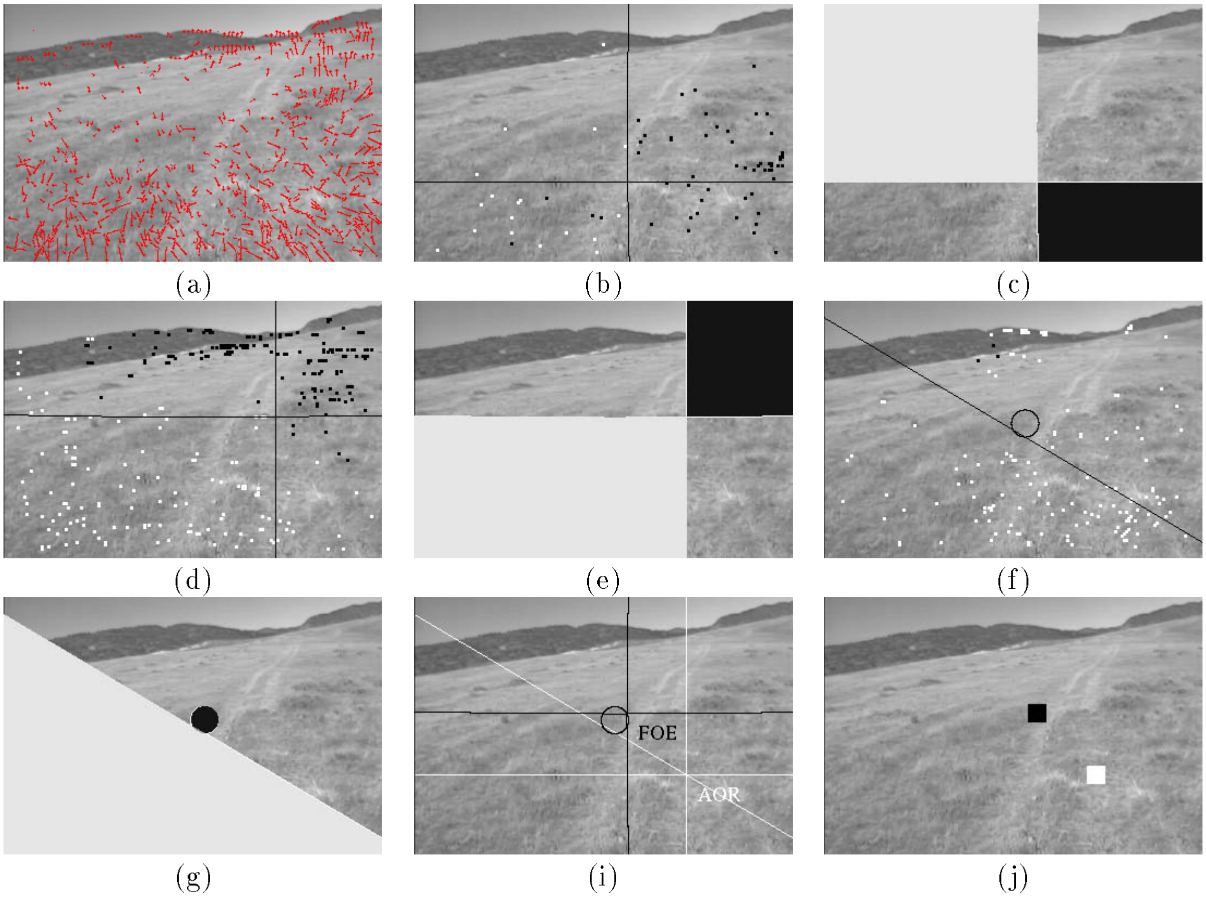


Figure 9: A camera mounted on the Martin Denver UGV captured a sequence of images as the vehicle moved along rough terrain in the countryside, thus undergoing continuously changing rigid motion. (a) shows one frame of the sequence with the normal flow field overlaid. (b), (d) and (f) show the positive (light color) and negative (dark color) vectors of the longitudinal patterns corresponding to the x-, y- and z-axes (see [Fermüller and Aloimonos, 1995]). (c), (e) and (g) show the corresponding fitted patterns. (i) shows, superimposed on the image, the boundaries of the patterns whose intersections provide the FOE and the AOR (the point where the rotation axis pierces the image plane). (j) Measurements are not everywhere available (strong intensity gradients are sparse), but a set of patterns can still be fitted, resulting in two bounded areas as locations for the FOE and the AOR.

2. Find the best match point in the image for each m using the following Winner-Take-All rule:

$$(x_{b_m}, y_{b_m}) = \text{argmin}_{(x,y)}\{S_m(x,y)\} \quad (4)$$

3. Construct a binary image B:

$$B(x,y) = \begin{cases} 1 & \text{if } (x,y) \in \{(x_{b_m}, y_{b_m})\} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

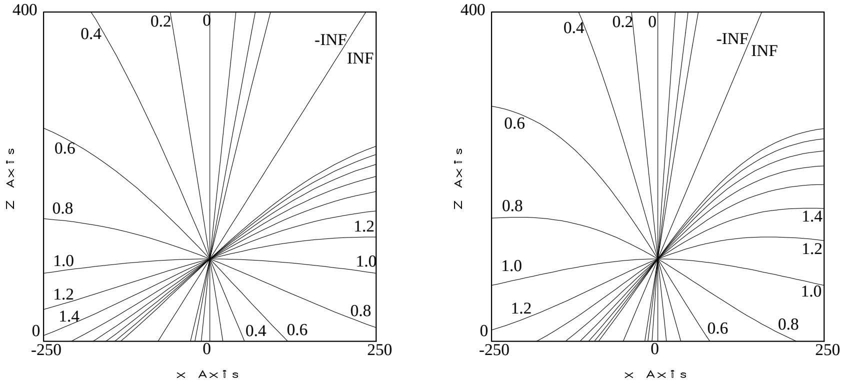
   where $m = 1, \ldots, m_{\max}$.

4. Output the location of the object in the current image as

$$(x_b, y_b) = \text{argmax}_{(x,y)}\ \{S(x,y) * B(x,y)\} \quad (6)$$

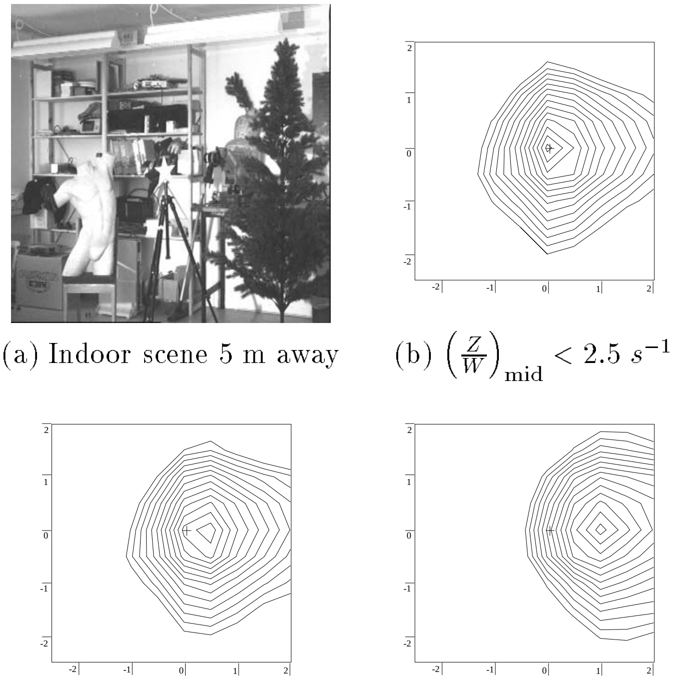
   where B is an appropriate blurring function whose size can usually be estimated in an active vision environment.

The location algorithm currently operates at close to real-time rates in an active vision system consisting of the University of Rochester binocular head with two movable color CCD cameras that provide input to a Datacube MaxVideo$^{\text{TM}}$ MV200 pipeline image-processing system. Given a live input image (of size 512×480) from the camera, the MV200 executes nine convolutions using nine different 8×8 discrete Gaussian derivative filter kernels on a low-pass filtered five-level pyramid of the image to obtain the response vectors for all points in the current image; these vectors are stored in a "memory surface" $\mathcal{S}$. During the memorization



(a) FOV=50° (b) FOV=70°

Figure 10: Iso-distortion contours resulting from intersecting the iso-distortion surfaces with the Zx-plane (the plane defined by the optical axis and the horizontal axis of the camera). The horizontal component x_0 of the actual Focus of Expansion is $x_0 = 50$. Assuming that the error in estimating x_0 is $x_{0e} = -50$ and the error in estimating the rotation around the y-axis is $\beta_e \simeq 0.001$, Figures 10a and 10b show iso-distortion contours for two different values of the FOV. The value next to each contour denotes the amount of multiplicative distortion (1.0 means no distortion).



(a) Indoor scene 5 m away (b) $\left(\frac{Z}{W}\right)_{\text{mid}} < 2.5\ s^{-1}$

(c) $\left(\frac{Z}{W}\right)_{\text{mid}} < 5\ s^{-1}$ (d) $\left(\frac{Z}{W}\right)_{\text{mid}} < 10\ s^{-1}$

Figure 11: The effectiveness of a relatively inexpensive, not highly accurate inertial sensor depends on the distribution of depths in the scene in view. The analysis using iso-distortion contours is based on whether "negative" depth values arise, and considers as a criterion for the estimation of the FOE the point that gives rise to a minimum number of non-positive depth measurements. The level contours in (b), (c), and (d) show the variation in the number of negative depths as the FOE estimates move away from the true FOE (indicated by the cross). The best FOE estimate is associated with the "bottom" of the contours (minimum number of negative depths). The axes of these contour plots represent the error of the FOE in degrees; they are not plotted at the same scale as the image in (a). FOE $= 30°$; $\beta_e = 0.04°/s$ (error in rotation around the y-axis); $(Z/W)_{\text{mid}}$ adjusted by changing W. The analysis suggests that an inertial sensor with an accuracy of $0.04°/s$ may be problematic in outdoor scenes but should be very successful in indoor scenes.

phase, filter responses are extracted for each of the sparse set of points located within the given object. During the location phase, a model response vector is loaded into the 8×8 convolution kernel and convolved with the memory surface $\mathcal{S}$ containing the response vectors for each point of the input image; the closest vectors can be selected by simply thresholding the results of the convolution at individual thresholds to obtain candidate match points.

Figure 12 shows an example of the performance of the location routine in a realistic scene. Here we demonstrate the algorithm's ability to find a model object (in this case, the stuffed doll) in the presence of object motion, clutter, and perspective distortion; '+' denotes the best matching location found by the algorithm.

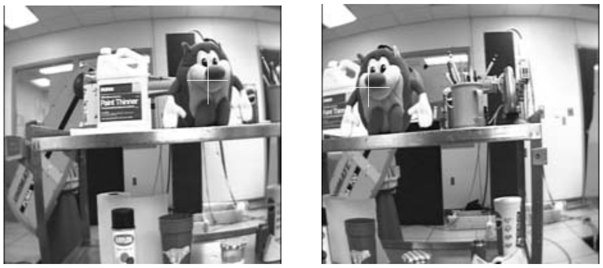


Figure 12: Example of locating a model object (stuffed doll) under conditions of motion, clutter, and perspective.

3.5 Active Intelligent Observers—University of Pennsylvania

Current active vision systems address two primary questions: how to select interesting parts of the scene to look at and how to maintain acquisition of the selected objects. We are interested in building an active intelligent observer on top of a reflexive gaze control system. Specifically, we propose to use high-level knowledge to direct the actions of an active vision system using feedback from low-level gaze control mechanisms.

Our approach comprises three phases. In the teaching phase, the active intelligent observer acquires a series of views of a reference object and integrates them into a structural model of the object. In the acquisition phase, the observer searches for the desired object in the scene and establishes the correct image size by moving and zooming. In the guidance phase, the observer dynamically constructs a gaze control path that leads to the optimal aspect for the current task. The structural model of the object allows the observer to determine the location of the optimal aspect in the view sphere and to generate intermediate views that guide the observer along the gaze control path. In robotic applications, the manipulated object must frequently be examined as to its identity and orientation. The active intelligent observer uses the structural model to determine the object's orientation and to move around it to view specific features.

Low-level image measures for gaze control are notoriously sensitive to changes in scale, orientation, and viewing aspect. However, if a simple template is augmented with high-level structural information, new views can be synthesized to guide the observer's gaze between known views. Conversely, the observer can infer the pose of an object from the current view by comparing it to the stored knowledge. The basic idea of this approach is to add a higher level of feedback to gaze control and close the perception-action loop around the visual servoing task.

References

[Balakirsky and Chellappa, 1996] S. Balakirsky and R. Chellappa. Performance characterization of image stabilization algorithms, Center for Automation Research Technical Report, University of Maryland, College Park, MD, to appear.

[Burt and Anandan, 1994] P. Burt and P. Anandan. Image stabilization by registration to a reference mosaic. In *Proc. of ARPA Image Understanding Workshop,* Monterey, CA, November 1994, pp. 425–434.

[Cheong and Aloimonos, 1995] L. Cheong and Y. Aloimonos. Isodistortion contours and egomotion estimation. In *Proc. of IEEE International Symposium on Computer Vision*, Coral Gables, FL, November 1995, pp. 70–76.

[Davis et al., 1994] L.S. Davis, R. Bajcsy, R. Nelson, and M. Herman. RSTA on the Move. In *Proc. of ARPA Image Understanding Workshop,* Monterey, CA, November 1994, pp. 435–456.

[Fermüller and Aloimonos, 1995] C. Fermüller and Y. Aloimonos. Qualitative egomotion. *International Journal of Computer Vision*, 15:7–29, 1995.

[Hosoda et al., 1995] K. Hosoda, H. Moriyama, and M. Asada. Visual servoing utilizing zoom mechanism. In *Proc. of IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 1995, pp. 178–183.

[Hwang et al., 1993] J. Hwang, Y. Ooi, and S. Ozawa. An adaptive sensing system with tracking and zooming a moving object. *IEICE Transactions on Information and Systems*, E76-D:926–934, 1993.

[Madden and Cahn von Seelen, 1995] B.C. Madden and U.M. Cahn von Seelen. PennEyes: A binocular active vision system. Technical Report, GRASP Laboratory, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, to appear.

[Morimoto et al., 1995] C.H. Morimoto, D. DeMenthon, L. Davis, R. Chellappa, and R. Nelson. Detection of independently moving objects in passive video. In *Proc. of IEEE Intelligent Vehicles Symposium*, Detroit, MI, 1995, pp. 270–275.

[Nelson, 1991] R.C. Nelson. Qualitative detection of motion by a moving observer. *International Journal of Computer Vision* 7:33–46, 1991.

[Nelson and Polana, 1992] R.C. Nelson and R. Polana. Qualitative recognition of motion from temporal texture. *CVGIP: Image Understanding*, 56:78–89, 1992.

[Parry et al., 1995] H.S. Parry, A.D. Marshall, and K.C. Markham. Integration of segmentation information and correlation technique for tracking objects in sequences of images. In *Proc. of Videometrics IV* (*SPIE Proceedings vol. 2598*), Philadelphia, PA, October 1995, pp. 208–219.

[Polana and Nelson, 1994] R. Polana and R.C. Nelson. Detecting activities. *Journal of Visual Communication and Image Representation*, 5:172–180, 1994.

[Polana and Nelson, 1993] R. Polana and R.C. Nelson. Detecting activities. In *Proc. of ARPA Image Understanding Workshop*, Washington, DC, April 1993, pp. 569–574.

[Rao and Ballard, 1995a] R.P.N. Rao and D.H. Ballard. Dynamic model of visual memory predicts neural response properties in the visual cortex. Technical Report 95-1, National Resource Laboratory for the Study of Brain and Behavior, 1995.

[Rao and Ballard, 1995b] R.P.N. Rao and D.H. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence* 78:461–505, 1995.

[Reid and Murray, 1993] I.D. Reid and D.W. Murray. Tracking foveated corner clusters using affine structure. In *Proc. of International Conference on Computer Vision*, Berlin, Germany, May 1993, pp. 76–83.

[Yao et al., 1996] Y.S. Yao, P. Burlina, and R. Chellappa. Stabilization of images acquired by unmanned ground vehicles. In these Proceedings.

[Zheng and Chellappa, 1993] Q. Zheng, and R. Chellappa. A computational vision approach to image registration. *IEEE Transactions on Image Processing* 2:311–326, 1993.