

Mobile Active-Vision Traffic Surveillance System for Urban Networks

Tamer Rabie,* Baher Abdulhai & Amer Shalaby

Intelligent Transportation Systems Centre, University of Toronto, Toronto, Ontario, M5S 1A4, Canada

&

Ahmed El-Rabbany

Department of Civil Engineering, Geomatics Engineering Group, Ryerson University, Toronto, Ontario, M5B2K3, Canada

Abstract: *This article discusses the development of a mobile bus-mounted machine vision system for transit and traffic monitoring in urban corridors, as required by intelligent transportation systems. In contrast to earlier machine vision technologies used for traffic management, which rely mainly on fixed-point detection and simpler algorithms to detect certain traffic characteristics, the new proposed approach makes use of a recent trend in computer vision research; namely, the active vision paradigm. Active vision systems have mechanisms that can actively control camera parameters such as orientation, focus, zoom, and vergence in response to the requirements of the task and external stimuli. Mounting active vision systems on buses will have the advantage of providing real-time feedback of the current traffic conditions, while possessing the intelligence and visual skills that allow them to interact with a rapidly changing dynamic environment, such as moving traffic and continuously changing image background.*

1 INTRODUCTION

Machine vision and video technologies are becoming increasingly popular in intelligent transportation systems

*To whom correspondence should be addressed. E-mail: tamer@cs.utoronto.ca.

(ITS) applications. Traffic surveillance and incident detection and management constitute the most widespread uses of video technology, whereas machine vision processing of license plate images for purposes of electronic toll enforcement makes up most of the other common applications. Recently, these technologies have been used in adaptive traffic signal control systems, for monitoring speeds, and for collecting travel time data.

ITS is an emerging global industry that capitalizes on advanced technologies to better manage the dynamic, overcongested transportation networks of today. The current ITS boom has given rise to the need for a comprehensive real-time surveillance of traffic conditions over the transportation network to allow for dynamic control and management of traffic. Existing traffic detection technologies cover a wide spectrum of technologies as well as performance, ranging from modest pavement-buried inductive loop detectors to more advanced pole-mounted off-road detectors such as microwave, radar, and camera-based detectors. All existing detector types, however, share a common limitation of being point detectors reflecting only traffic conditions at the locations of the detectors.

Many ITS applications require the collection of traffic data over an extended time period at one or more locations. Wide area video traffic surveillance is one such example. Such applications require the installation of

numerous video cameras along with communications infrastructure to transmit video images to a Traffic or Transit Management Center (TMC) housing computer and video equipment for data processing and dissemination of information or control. In contrast, other applications require video images to be collected by means of tripod-mounted camcorders, with the images stored on videotape for subsequent viewing or machine vision processing (Shuldiner, 1999).

In the past, vehicle-mounted machine vision technologies have been employed extensively in a variety of applications that deal mainly with intelligent vehicle (IV) systems and automatic vehicle driving (AVD). AVD is a generic term used to address a technique aimed at automating, entirely or in part, one or more driving tasks. The main challenges that AVD techniques have to contend with include: the possibility to follow the road and keep within the right lane, maintaining a safe distance between vehicles, adjusting the vehicle's speed according to traffic conditions and road characteristics, changing lanes to overtake vehicles and avoid obstacles, helping to find the correct and shortest route to a destination, and the movement and parking within urban environments (Broggi et al., 1998). Unlike existing technologies, our research focuses on integrated development of a wide-area mobile-surveillance system, where the tracker (the vision system on the bus), the target (traffic in the camera view) and the processing tools (computer, systems, and algorithms) are all in a mobile environment, posing new challenges to be addressed.

2 THE BUS-MOUNTED ACTIVE VISION SYSTEM

In this project, we develop a binocular machine vision system, mounted on buses in an urban corridor, to dynamically monitor traffic conditions along the corridor. Although the concept of using buses as probes of traffic conditions has recently gained considerable momentum, the use of mobile vision technology as proposed in this work has not been investigated to the best of our knowledge. Stationary real-time monocular machine vision systems have recently been developed for tracking vehicles under congested conditions for the purpose of traffic management (Beymer et al., 1997).

From a technical perspective, our machine-vision-based approach is active and dynamic, as both the tracking agent and the targets are mobile in a dynamic environment. The level of complexity is significantly higher than the case of pole-mounted video technology, where the camera and the background are both practically static, which mainly relies on passive-vision techniques to detect moving traffic objects and characteristics. The proposed approach employs recent trends in computer

vision research, namely, the active vision paradigm. Active vision systems have mechanisms that can actively control camera parameters such as orientation, focus, zoom, and vergence in response to the requirements of the task and external stimuli.

Mounting active vision systems on buses has the advantage of providing real-time network-wide feedback of the current traffic conditions while possessing the intelligence and visual skills that allow them to interact with a rapidly changing dynamic environment, such as moving traffic. The main approach is to stabilize the visual field of view of the camera system by compensating for the vibrations due to the motion of the bus. This can be achieved by computing the displacement between successive video image frames and updating the gaze angles of the camera to cancel out this displacement, which can be computed as a translational offset in the image coordinate system. Once the video image stream is stable, the next task is to detect and track vehicles that appear in the camera field of view. The relative speed of vehicles with respect to the speed of the bus can be estimated by tracking the motion of vehicles between frames. An accurate estimate of traffic speed can be obtained from the knowledge of the speed of the camera, which has the same speed as the bus, and the relative speeds estimated from the video images. Further traffic characteristics such as density and the presence of incidents can also be deduced from the image sequence.

2.1 Technical rationale and approach

Components similar to the bus-mounted machine vision system are currently available in other fields. In particular, recent research carried out by Rabie and Terzopoulos (1996, 1997, 1998) on active vision in a simulated 3D virtual environment has enabled a new paradigm for computer vision research that is called "animat vision." The essence of the concept is to implement active vision systems allowing virtual animal robots (or animats) to understand perceptually the realistic virtual worlds in which they are situated, so that they may interact effectively with other animats situated within these worlds. A set of active vision algorithms have successfully been implemented, within the animat vision framework, that integrate motion, stereo, and color analysis. These algorithms support robust color-object tracking, vision-guided navigation, visual perception, and obstacle recognition and avoidance abilities, enabling the animat to better understand and interact with its dynamic virtual environment. It should be pointed out that the images acquired by the animats are 2D projections of the simulated 3D virtual world using computer graphics camera models. In comparison, the images acquired by the bus-mounted vision cameras are real-world images.

We are combining the stereo, motion, and color algorithms together with a robust tracking algorithm, namely, the KLT (Kanade–Lucas–Tomasi) feature tracker, to increase the robustness and functionality of our overall vision system. The KLT feature tracker is based on the early work of Lucas and Kanade (1981), and later developed fully by Tomasi and Kanade (1991). The only published and readily accessible description of this tracker is contained in an article by Shi and Tomasi (1994). The final bus-mounted vision system is designed to be able to detect moving targets of interest (moving vehicles) and segment their region of support using motion detection and optical-flow estimation. It then changes its stereo camera gaze angles to fixate the detected target of interest. With the target inside the left and right camera's field of view, the feature tracker algorithm takes control to keep the moving target fixated and in view. Tracking also facilitates the estimation of the relative speed of the tracked vehicle. The stereo algorithm estimates the relative distance between the bus-mounted cameras and vehicles visible in front of the bus. These distances are used by the tracker to filter out undesirable features (disconnected features and features on objects that are too close to the bus), and focus on tracking features that belong to the same vehicle.

This research builds on existing and tested vision techniques as a starting point from which we continue further development, augmenting them with enhancements suitable for the new application at hand.

2.2 The prototype bus-mounted vision system

The bus-mounted vision system, which was briefly discussed in Rabie et al. (2002), consists of two main modules: the motion stabilization module and the stereo-tracking module, as shown in Figure 1. Together they implement a gaze control capability that enables the camera system to stabilize the visual world due to vibrations caused by the motion of the bus, as well as to detect visual targets in the camera field of view, change the gaze to lock them in, and visually track them between image frames. Disparities between the stabilized left and right camera images are estimated by the stereo-tracking module, thus giving an estimate of the relative distance to objects in the image.

2.3 Visual field stabilization using optical flow

It is necessary to stabilize the visual field of the stereo camera system because of the continuous motion of the bus. The optokinetic reflex in animals stabilizes vision by

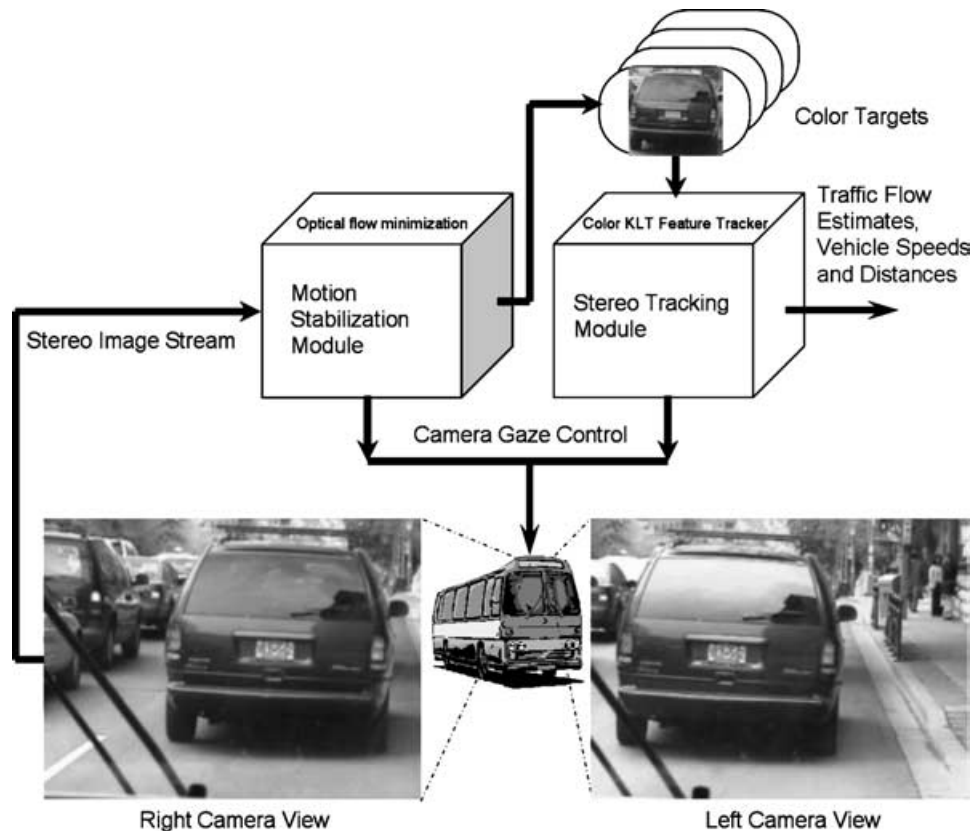


Fig. 1. The prototype bus-mounted active vision system diagram.

measuring image motion and producing compensatory eye movements. Stabilization is achieved by computing the displacement between successive image frames and updating the camera gaze angles by this displacement offset. The displacement is computed as a translational offset in the image frame coordinate system by a least-squares minimization of the optical flow constraint equation between image frames at times t and $t - 1$ (Burt et al., 1989; Irani et al., 1994). Given a sequence of time-varying images, points imaged on the retina appear to move because of the relative motion between the eye and objects in the scene (Gibson, 1979). The instantaneous velocity vector field of this apparent motion is usually called optical flow (Ballard and Brown, 1982). Optical flow can provide important information about the spatial arrangement of objects viewed and the rate of change of this arrangement (Horn, 1986). Various techniques for determining optical flow from a sequence of two or more frames have been proposed in the literature (Horn and Schunck, 1981; Anandan, 1989; Lucas and Kanade, 1981). The optical flow constraint equation is given by (Horn, 1986)

$$uI_x + vI_y + I_t = 0 \quad (1)$$

where $(I_x, I_y, I_t)^T$ is the spatiotemporal image intensity gradient given as $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t})^T$. Values of (u, v) satisfying this constraint equation lie on a straight line in velocity space. The image intensity is computed as

$$I(x, y, t) = \frac{1}{3}[R(x, y, t) + G(x, y, t) + B(x, y, t)] \quad (2)$$

where $R, G,$ and B denote the color component channels. The error function

$$E(u, v) = \sum_{x, y \in \text{fovea}} (uI_x + vI_y + I_t)^2 \quad (3)$$

is minimized by simultaneously solving the two equations $\partial E/\partial u = 0$ and $\partial E/\partial v = 0$ for the image displacement (u, v) .

Optical flow, once reliably estimated, can be very useful in various computer vision applications. Discontinuities in the optical flow can be used in segmenting images into moving objects (Burt et al., 1989; Irani and Peleg, 1992). Navigation using optical flow and estimation of time-to-collision maps have been discussed in Campani et al. (1995) and Meyer and Bouthemy (1992).

Once the camera's visual view is stabilized against any vibrations, the camera gaze is redirected at a moving target of interest. Redirecting gaze when a target of interest appears in the image frame can be a complex task. One solution would be to section the peripheral image into smaller patches or focal probes (Burt et al., 1989) and search all the probes for large motion fields. The strategy would work well for sufficiently small images, but for

dynamic vision systems that must process natural images, this approach is not effective. We are exploring a method based on motion cues to help narrow down the search for a suitable gaze direction. We first create a saliency image consisting of the optical flow field between two stabilized image frames. The saliency image is then convolved with a circular disk of area equal to the expected area of the target object of interest as it appears in the image frame. Reasonably, small areas suffice because objects in the image frame are typically small in front of the bus-mounted camera. Methods for estimating appropriate areas for the object, such as Jagersand's information theoretic approach (Jagersand, 1995), may be applied. The blurring of the saliency image emphasizes moving objects in the image. The maximum in the blurred saliency image is taken as the location of the fastest moving object and serves as the new gaze direction for the stereo camera system.

3 TARGET TRACKING

Once the stereo camera system has been redirected to gaze at an appropriate target, the stereo-tracking module assumes the task of selecting good features from the current image frame, consistently tracking these features over time. Tracking moving objects in video streams has been a popular topic in the field of computer vision in the last few years. The different tracking techniques for video data can be classified into four main approaches:

1. *3D Model-Based Tracking*: Three-dimensional model-based vehicle-tracking systems have previously been investigated in the literature (Koller et al., 1993; Baker and Sullivan, 1992). The emphasis is on recovering trajectories and models with high accuracy for a small number of vehicles. The most serious weakness of this approach is the reliance on detailed geometric object models. It is unrealistic to expect to be able to have detailed models for all vehicles that could be found on the roadway.
2. *Region-Based Tracking*: The idea here is to identify a connected region in the image associated with each vehicle and then track this region over time using a cross-correlation measure. Initialization of the process is most easily done by subtracting the background scene from the acquired image. A Kalman filter-based adaptive background model allows the background estimate to evolve while the weather and time of day affect lighting conditions. Foreground objects (vehicles) are detected by subtracting the incoming image from the current background estimate, looking for pixels where this difference image is above some threshold and then

finding connected components (Gloyer et al., 1994). This approach works fairly well in free-flowing traffic. However, under congested traffic conditions, vehicles partially occlude one another instead of being spatially isolated, which makes the task of segmenting individual vehicles difficult. Such vehicles will become grouped together as one large blob in the foreground image.

3. *Active Contour-Based Tracking*: A dual to the region-based approach is tracking based on active contour models, or snakes (Kass et al., 1987). The idea is to have a representation of the bounding contour of the object and keep dynamically updating it. The advantage of having a contour-based representation instead of a region-based representation is reduced computational complexity. However, the inability to segment vehicles that are partially occluded remains. If one could initialize a separate contour for each vehicle, then one could track even in the presence of partial occlusion (Beymer et al., 1997).
4. *Feature-Based Tracking*: Finally, yet another approach to tracking abandons the idea of tracking objects as a whole, but instead tracks subfeatures such as distinguishable points or lines on the object. The advantage of this approach is that even in the presence of partial occlusion, some of the subfeatures of the moving object remain visible. The technology of tracking points and line features is developed fully as the KLT feature tracker by Tomasi and Kanade (1991).

For our specific application, we require efficiency, robustness to occlusion, and real-time tracking at all times. The feature-based tracking approach, described above,

satisfies our requirements. We, thus, incorporate the KLT feature tracker into the stereo-tracking module. This tracker locates good features by examining the minimum eigenvalue of each 2 by 2 image-gradient matrix, and features are tracked using a Newton–Raphson method of minimizing the difference between the two matrices in two consecutive frames. Multiresolution tracking allows for large displacements between images. A readily accessible mathematical description of this tracker is contained in the article by Shi and Tomasi (1994).

3.1 Vehicle speed estimation

To help our tracking algorithm focus on the tracked vehicle and to reduce distraction due to background clutter in the image sequence, we feed the KLT feature, tracking algorithm difference images instead of the actual images (as depicted clearly in Figure 2). A difference image is created by subtracting the previously acquired image of the road from the image acquired at the current time instant. This can be given by

$$I_{\Delta}(t) = |I(t) - I(t - 1)| \quad (4)$$

This has the advantage of blocking out the details in the background of the tracked vehicle, while only emphasizing the vehicle to be tracked. This is possible due to the fact that the closer the vehicle is to the camera, the larger its motion will be, and the farther away the vehicles are, the more insignificant their motions will be. Thus, when subtracting the previous image from the current image, only the large motions of the vehicle in front of the bus will be emphasized in the difference image, whereas the vehicles that are far away from the bus get canceled out of the difference image. To facilitate this, we make use of small camera fields of view, thus

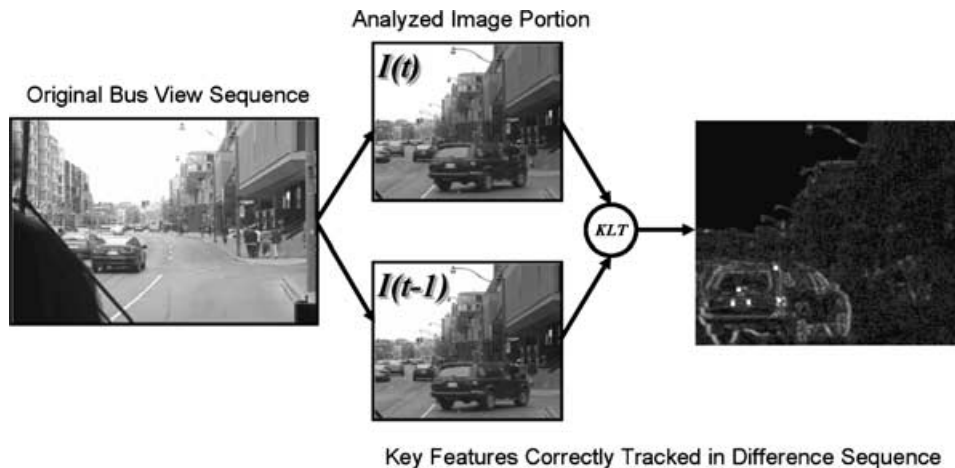


Fig. 2. An image from a single bus-mounted camera. Difference images $I_{\Delta}(t) = |I(t) - I(t - 1)|$ are fed to the KLT feature-tracking algorithm instead of the actual images.

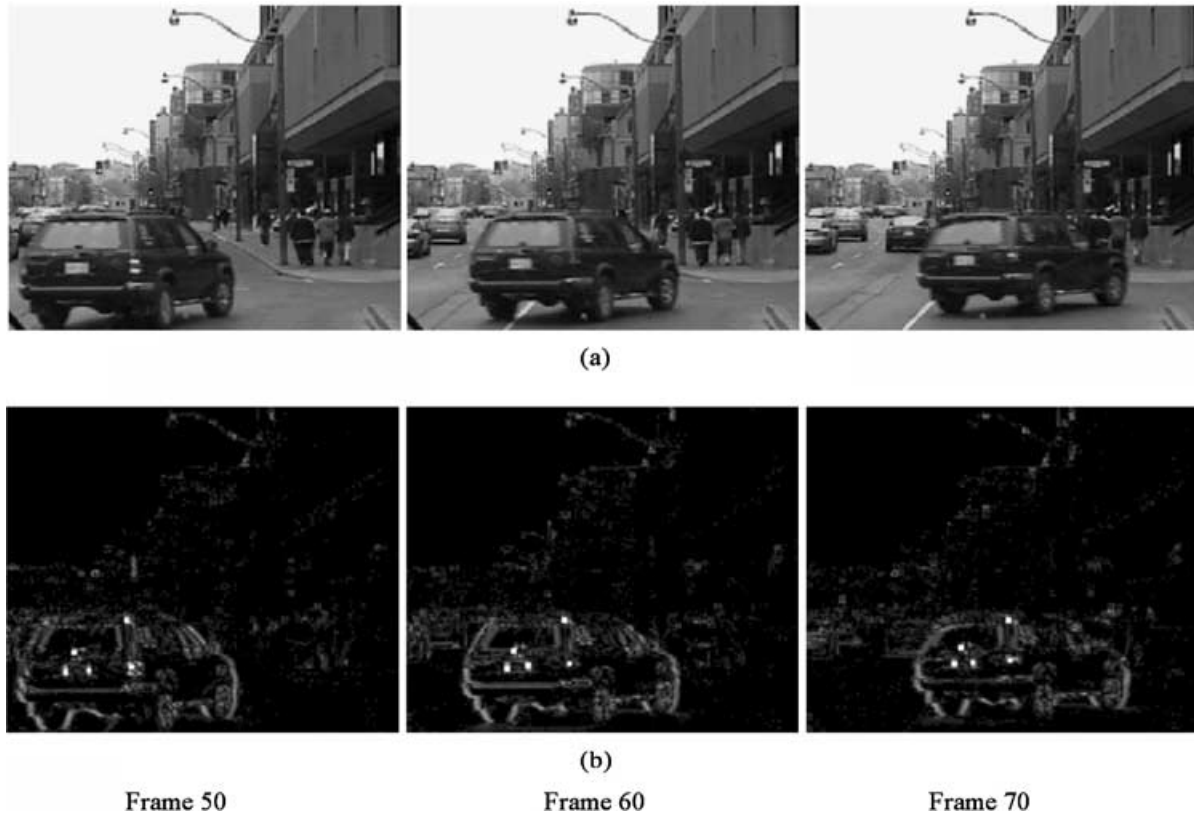


Fig. 3. Good features represented as bright white feature points are tracked over the sequence of 20 difference image frames. (a) The original bus view sequence, (b) the tracked features in the difference image sequence.

only capturing a small area of the image in front of the bus, where the possibility of capturing only a single vehicle is higher. Even if the camera captures more than one vehicle in front of the bus, the difference image will still emphasize the motion of the closer vehicle to the bus-mounted camera. Figure 3a shows three selected image frames of a longer sequence acquired by a camera mounted on a Toronto Transit Commission (TTC) bus in Toronto. Figure 3b shows the corresponding difference images with the three bright white points on the license plate of the 4×4 vehicle corresponding to the good features that the tracker algorithm was able to select and successfully track between image frames. The images clearly show that key features are correctly detected and tracked over time. The displacement of these tracked features between image frames can give the relative speed of the tracked vehicle, knowing that the video camera is recording at 30 frames per second. The speed of the tracked vehicle is estimated based on the speed of the bus, which is optimally obtained through a Kalman filter-based integrated low-cost GPS/Dead Reckoning (DR)/Signpost (SP) system as depicted in Figure 4, with

Δd indicating the traveled distance between two time epochs, which is measured by the bus' odometer.

4 STEREO VISION

An image acquired from a pair of horizontally mounted left and right cameras viewing the same scene from slightly displaced view points is commonly known as a stereo image pair. Stereo vision is concerned with the extraction of scene depth information by matching corresponding points in the left and right images of a stereo image pair. The horizontal difference between the two matched points is known as the disparity. The depth from the camera to the matched point is proportional to the reciprocal of the disparity (Chen and Bovik, 1995). The task of determining the correspondence between points in the two views is known as the correspondence problem and is considered difficult. In general, it is a 2D search through the entire image space (Jenkin and Tsotsos, 1994). Knowledge of the camera geometry can be used to limit the search to be 1D along the epipolar



Fig. 4. Flow chart of the integrated GPS/dead reckoning/signpost for bus location.

line, which is the intersection of the left and right image planes with the epipolar plane (the plane through a point in the scene and the nodal points of the two cameras) (Horn, 1986).

4.1 Our approach to disparity estimation

Classical approaches to determine stereo disparity try to deal with the correspondence problem with two basic algorithms: area-based (Lucas and Kanade, 1981; Horn, 1986) and feature-based approaches (Marr and Poggio, 1979; Horn, 1986). Other methods extract local Fourier phases of left and right images and the phase difference at each location is used to estimate disparity (Sanger, 1988; Langley et al., 1990; Fleet et al., 1991). Several approaches take into consideration available biological and neurophysiologic data about the human visual system (Marr and Poggio, 1979; Sanger, 1988; Jones and Malik, 1992). There is biological evidence that the pattern of light projected on the human retina is sampled and spatially filtered. Very early in the cortical visual processing, receptive fields become oriented and are well approximated by linear spatial filters, with impulse response functions that are similar to partial derivatives of a Gaussian function (Young, 1986).

The technique that we develop for estimating stereo disparity draws ideas from this early visual processing in the human cortex. We implement the receptive fields as steerable spatial filters that process the input stereo image pair. The steerable filter responses at an image location form a “feature vector” that is used for solving the correspondence problem. The outputs of a steerable filter convolved with an image at multiple orientations provide very rich information about a local neighborhood around each pixel. Thus, matching image patches from the left and right images of a stereo pair becomes simpler and the probability of a correct match increases as the length of the feature vector increases.

Oriented filters are important for many computer vision and image-processing tasks, such as texture analysis, image enhancement, and motion analysis. One approach to finding the response of a filter at many orientations is to apply many versions of the same filter, each differing from one another by a small rotation in angle. A more efficient approach is to apply a few filters corresponding to a few angles and interpolate between the responses. With the correct filter set and the correct interpolation rule, it is possible to determine the response of a filter of arbitrary orientation without explicitly applying that filter.

“Steerable filter” is a term used to describe a class of spatial filters in which a filter of arbitrary orientation is synthesized as a linear combination of a set of basis filters. Steerable filters, first developed by Freeman and Adelson (1991), have recently been used for estimation of scene motion (Huang and Chen, 1995) and for object recognition (Ballard and Wixson, 1993) and stereopsis (Jones and Malik, 1992). As an example, consider the 2D circularly symmetric Gaussian function

$$G(x, y) = e^{-(x^2+y^2)} \quad (5)$$

The first x derivative of this Gaussian is

$$G_1^{0^\circ} = \frac{\partial}{\partial x} G(x, y) = -2xe^{-(x^2+y^2)} \quad (6)$$

and the same function rotated 90° is

$$G_1^{90^\circ} = \frac{\partial}{\partial y} G(x, y) = -2ye^{-(x^2+y^2)} \quad (7)$$

Thus, the derivative in an arbitrary direction θ can be synthesized by taking a linear combination of the basis filters $G_1^{0^\circ}$ and $G_1^{90^\circ}$ as follows:

$$G_1^\theta = \cos(\theta)G_1^{0^\circ} + \sin(\theta)G_1^{90^\circ} \quad (8)$$

The $\cos(\theta)$ and $\sin(\theta)$ terms are the corresponding interpolation functions for those basis filters. Because

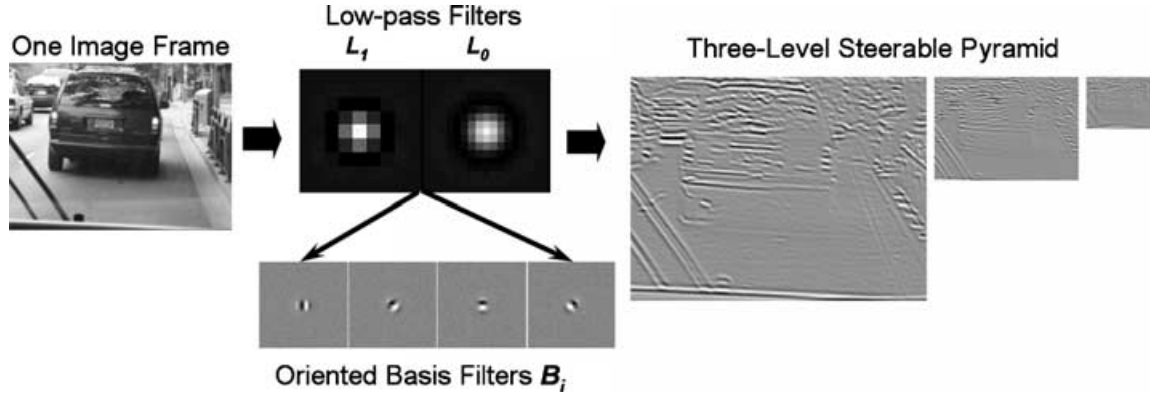


Fig. 5. System diagram for the first level of the steerable pyramid. L_0 and L_1 are low-pass filters, and B_i are oriented basis filters. Successive levels of the pyramid are computed by applying the B_i and L_1 filtering and subsampling operations recursively. The four basis filters (B_i) are oriented at 0° , 45° , 90° , and 135° from left to right.

convolution is a linear operation, it is possible to synthesize an image filtered at an arbitrary orientation by taking linear combinations of the images filtered with $G_1^{0^\circ}$ and $G_1^{90^\circ}$:

$$G_1^\theta * I(x, y) = \cos(\theta)G_1^{0^\circ} * I(x, y) + \sin(\theta)G_1^{90^\circ} * I(x, y) \quad (9)$$

This gives an illustration of steerability, which is a very useful property, because the response of a steerable filter at an arbitrary orientation can be obtained from a small number of precomputed basis responses using the corresponding interpolation functions. Simoncelli and Freeman have recently introduced a multiscale, multiorientation steerable filter image decomposition framework called the Steerable Pyramid (Simoncelli and Freeman, 1995), which we use as a front-end for our stereo algorithm. It has the advantage of producing feature descriptions that are both translation invariant and rotation invariant.

4.2 Disparity-estimation algorithm

Our disparity-estimation algorithm starts by decomposing the left and right images into steerable pyramid representations using the framework developed by Simoncelli and Freeman (1995). The input images are initially low-pass filtered using a low-pass filter (L_0) with a radially symmetric frequency response. Each successive level of the pyramid is constructed from the previous level's low-pass band by subsampling it, then convolving it with a bank of oriented basis filters (B_i) and a low-pass filter (L_1). Other orientations at each level are synthesized by taking linear combinations of the basis-filtered images. The number of basis filters that are needed for steering the filter is $n + 1$ for the n th order filter. We use third-order filters, thus requiring four basis filters ori-

ented at 0° , 45° , 90° , and 135° (Freeman and Adelson, 1991). Figure 5 shows these four spatial basis filters (B_i) which form a steerable basis set; any orientation of this filter can be written as a linear combination of the basis filters. Figure 5 also shows the two low-pass filters (L_0 and L_1) used to construct the pyramid (Simoncelli and Freeman, 1995). The extreme right of Figure 5 shows an example of a three-level steerable pyramid for a single orientation for the right image of a stereo image pair.

Feature vectors $f_R(x, y, l)$ and $f_L(x, y, l)$ are then constructed from the right and left pyramid responses for each pixel at each level of the pyramid by combining the responses of the multiorientation steerable filters at each pixel into a vector that provides a very rich description of the intensities at that pixel in the image. To further enrich the description of each pixel, we make use of the (R, G, B) color signals from our color images by including them in the feature vector. This simple addition improves our matching process considerably, by restricting the matching process to areas of similar color composition, which can be considered as a sort of color-feature constraint.

An initial disparity map is estimated at each individual level by matching left and right feature vectors by minimizing the mean square error (MSE) between left and right feature vectors. The MSE measure is computed over all the elements in the vector as follows:

$$E_m = \frac{1}{S} \sum_{i \in S} [f_R^i(x, y, l) - f_L^i(x + d_x, y + d_y, l)]^2 \quad (10)$$

where S is the size of the feature vector. The MSE measure E_m is computed for a limited range of horizontal and vertical disparities $d_x(l) \in D_x(l)$ and $d_y(l) \in D_y(l)$ within a window of size $D_x(l) \times D_y(l)$ (typically, $D_x(0) = 20$ and $D_y(0) = 10$). The $[d_x(l), d_y(l)]$ value that minimizes the MSE within this window is taken as the best initial

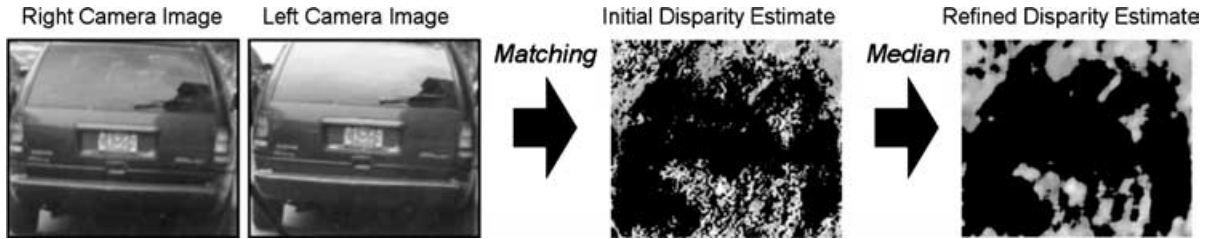


Fig. 6. Minivan sequence taken onboard a TTC bus in Toronto, showing a portion of a single right and left stereo image pair, and its initial disparity map estimate. The darker the gray shades in the disparity map the closer the object is to the bus-mounted camera. The full-frame disparity map is median filtered with a window size of 13×13 pixels to fill in the misclassified disparity estimates, thus, clearly showing in black the dominant minivan in the foreground.

disparity estimate at pixel (x, y, l) at pyramid level l . A boundary condition of zero disparity at image borders is applied. Also, a zero disparity condition is applied to locations where no match is possible such as across constant intensity areas. The disparity range used lies within $[-D(l)/2, D(l)/2]$. The disparity range differs from level to level and is given as

$$D_x(l) = \frac{D_x(0)}{2^l} \quad \text{and} \quad D_y(l) = \frac{D_y(0)}{2^l} \quad (11)$$

Note that there is no need to equally weigh the dimension of the (R, G, B) color signals to correspond to the rest of the feature vector element dimension (i.e., the dimensions of the steerable filter responses) because in Equation (10), the MSE compares corresponding elements of the same dimension together. Thus, for example, the element that corresponds to red in the right feature vector f_R^{red} is compared to the red element in the left feature vector f_L^{red} . Similarly, the steerable filter elements from the right vector are compared to corresponding elements from the left. Therefore, the dimensions of the corresponding elements of the feature vector must be the same for proper matching.

A coarse-to-fine-flow-through strategy is then taken based on the assumption that for level l disparity estimates $|d(l)| > |D(l)/4|$ are more accurately estimated at the coarser level $l + 1$. Thus, at coarse levels large disparities are estimated presumably more accurately, and these flow through to the finer levels, whereas small disparities that are estimated from the finer levels are assumed accurate because they cannot be estimated at coarser levels due to the loss of high-frequency structure from the original coarse-level images.

Each disparity estimate $[d_x(l), d_y(l)]$ at each level is median filtered at an appropriate scale (window size used increases from coarse levels to fine levels—mainly 3×3 and 5×5 for 128×128 images) before flow through is performed. The full frame level is then median filtered with a window size of up to 13×13 to give the final disparity estimate. The median filtering step is required to correct for outlier disparity estimates that deviate from

the correct expected estimate (a form of smoothness constraint on the estimates). Figure 6 clearly shows the disparity map estimated for a single pair of stereo images taken on board a Toronto Transit Commission (TTC) bus.

In future work, we will make use of the stereo bus-mounted camera by employing this stereo algorithm to estimate the relative distance between the bus-mounted cameras (the observer) and the tracked vehicles visible in the image. This will allow the tracking module to reject objects that are too close or too distant, thus giving a more accurate estimate of the relative speed of the tracked vehicle. Tracking features on objects that are too close to the observer can be prone to errors due to the difficulty of accurately estimating large displacements typical of close objects. The estimated disparities will also be used to allow the tracker to focus on tracking feature points that have similar disparity estimates (low-variance disparities) and where there are no disparity discontinuities (high-variance disparities) in between them indicating that they belong to the same vehicle.

5 CONCLUSIONS

This article presents the results of phase I of an ongoing research program aimed at the development of a mobile, bus-mounted machine vision technology for transit and traffic monitoring in urban corridors for ITS applications. The development of the core vision modules is detailed in this article including visual field stabilization using optical flow, target tracking, vehicle speed estimation, stereo analysis and disparity estimation. Clear distinction has been made between the new approach and existing video-based static surveillance techniques. Unlike existing technologies, the present research focuses on integrated development of a wide-area mobile-surveillance system where the tracker (the vision system on the bus), the target (traffic in the camera view), and the processing tools (computer, systems, and algorithms)

are all in a mobile environment. This approach broadens the horizons and potential benefits of video-based traffic network surveillance, while posing new challenges and interesting problems to researchers and practitioners. The core of the approach is based on a new paradigm for computer vision research that is called "animat vision" developed by the lead author. The essence of the concept is to implement active vision systems allowing virtual animal robots (or animats) to understand perceptually the realistic virtual worlds in which they are situated so that they may effectively interact with other animats situated within these worlds. Analogously, a stereo-eyed bus can interact with surrounding traffic in a congested urban network and infer the nature and status of its environment.

In the next phase of this research, the developed vision modules will be integrated into a fairly sophisticated dynamic machine vision system that will be mounted on GPS-equipped transit buses, calibrated, and extensively validated in the City of Toronto. It will also be used to further develop a number of applications, including direct measurement of traffic variables such as speeds, volumes and densities as well as higher level applications such as the detection of traffic-impeding incidents, transit fleet progress monitoring, dynamic bus arrival time estimation and priority assignment at intersections, to name a few examples.

ACKNOWLEDGMENTS

The authors would like to acknowledge the valuable support of Dr. Gasser Auda during the early stages of the project. This research is funded by Communications and Information Technology Ontario (CITO).

REFERENCES

- Anandan, P. (1989), A computational framework and algorithm for the measurement of visual motion, *International Journal of Computer Vision*, **2**, 283–310.
- Baker, K. D. & Sullivan, G. D. (1992), Performance assessment of model-based tracking, in *Proceedings of the IEEE Workshop on Applications of Computer Vision*, Palm Springs, CA, 28–35.
- Ballard, D. H. & Brown, C. M. (1982), *Computer Vision*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Ballard, D. H. & Wixson, L. E. (1993), Object recognition using steerable filters at multiple scales, in *Proceedings of the IEEE Workshop on Qualitative Vision*, Los Alamitos, CA, 2–10.
- Beymer, D., McLauchlan, P., Coifman, B. & Malik, J. (1997), A real-time computer vision system for measuring traffic parameters, in *IEEE Conference on Computer Vision and Pattern Analysis (CVPR'97)*.
- Broggi, A., Bertozzi, M., Fascioli, A. & Conte, G. (1998), *Automatic Vehicle Guidance: The Experience of the ARGO Autonomous Vehicle*, World Scientific Publishing Co., River Edge, NJ.
- Burt, P. J., Bergen, J. R., Hingorani, R., Kolczynski, R., Lee, W. A., Leung, A., Lubin, J. & Shvaytser, H. (1989), Object tracking with a moving camera: An application of dynamic motion analysis, in *Proceedings of the IEEE Workshop on Visual Motion*, 2–12.
- Campani, M., Giachetti, A. & Torre, V. (1995), Optic flow and autonomous navigation, *Perception*, **24**, 253–67.
- Chen, T. Y. & Bovik, A. V. (1995), Stereo disparity from multiscale processing of local image phase, in *Proceedings of the Fifth International Conference on Computer Vision (ICCV'95)*, MIT, Cambridge, MA, 188–93.
- Fleet, D., Jepson, A. & Jenkin, M. (1991), Phase-based disparity measurement, *CVGIP: Image Understanding*, **53**, 198–210.
- Freeman, W. T. & Adelson, E. H. (1991), The design and use of steerable filters, *IEEE Transactions PAMI*, **13**, 891–906.
- Gibson, J. J. (1979), *The Ecological Approach to Visual Perception*, Houghton Mifflin, Boston, MA.
- Gloyer, B., Aghajan, H. K., Siu, K. Y. & Kailath, T. (1994), Vehicle detection and tracking for freeway traffic monitoring, in *Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers*, **2**, 970–4.
- Horn, B. K. P. (1986), *Robot Vision*, MIT Press, Cambridge, MA.
- Horn, B. K. P. & Schunck, B. G. (1981), Determining optical flow, *Artificial Intelligence*, **17**, 185–203.
- Huang, C. L. & Chen, Y. T. (1995), Motion estimation method using a 3D steerable filter, *Image and Vision Computing*, **13**, 21–32.
- Irani, M. & Peleg, P. (1992), Image sequence enhancement using multiple motion analysis, in *Proceedings of the 11th IAPR, International Conference on Pattern Recognition*, 216–21.
- Irani, M., Rousso, B. & Peleg, S. (1994), Recovery of egomotion using image stabilization, in *Proceedings of the IEEE Workshop on Visual Motion*, 454–60.
- Jagersand, M. (1995), Saliency maps and attention selection in scale and spatial coordinates: An information theoretic approach, in *Proceedings of the International Conference on Computer Vision (ICCV'95)*, MIT, Cambridge, Massachusetts, 195–202.
- Jenkin, M. R. M. & Tsotsos, J. K. (1994), Active stereo vision and cyclotorsion, in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR'94)*, 806–11.
- Jones, D. G. & Malik, J. (1992), A computational framework for determining stereo correspondence from a set of linear spatial filters, in *Proceedings of the European Conference on Computer Vision (ECCV'92)*, Portofino, Italy, 395–410.
- Kass, M., Witkin, A. & Terzopoulos, D. (1987), Snakes: Active contour models, *International Journal of Computer Vision*, **4**, 321–31.
- Koller, D., Daniilidis, K. & Nagel, H. (1993), Model-based object tracking in monocular image sequences of road traffic scenes, *International Journal of Computer Vision*, **10**, 257–81.
- Langley, K., Atherton, T. J., Wilson, R. G. & Larcombe, M. H. E. (1990), Vertical and horizontal disparities from phase, in *Proceedings of the European Conference on Computer Vision (ECCV'90)*, 315–25.
- Lucas, B. D. & Kanade, T. (1981), An iterative image registration technique with an application to stereo vision, in *Proceedings Image Understanding Workshop*, 121–30.

- Marr, D. & Poggio, T. (1979), A computational theory of human stereo vision, in *Proceedings of the Royal Society of London, Series B*, **204**, 301–28.
- Meyer, F. & Bouthemy, P. (1992), Estimation of time-to-collision maps from first order motion models and normal flows, in *Proceedings of the 11th IAPR, International Conference on Pattern Recognition*, 78–82.
- Rabie, T., Shalaby, A., Abdulhai, B. & El-Rabbany, A. (2002), Mobile vision-based vehicle tracking and traffic control, in *Proceedings of the 5th IEEE International Conference on Intelligent Transportation Systems*, Singapore, September 3–6, pp. 13–18.
- Rabie, T. F. & Terzopoulos, D. (1996), Motion and color analysis for animat perception, in *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, Portland, Oregon, 1090–7.
- Rabie, T. F. & Terzopoulos, D. (1998), Stereo and color analysis for dynamic obstacle avoidance, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, Santa Barbara, California, 245–52.
- Sanger, T. (1988), Stereo disparity computation using gabor filters, *Biological Cybernetics*, **59**, 405–18.
- Shi, J. & Tomasi, C. (1994), Good features to track, in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600.
- Shuldiner, P. W. (1999), Video technology in traffic engineering and transportation planning, *Journal of Transportation Engineering*, **125**(5), 377–83.
- Simoncelli, E. P. & Freeman, W. T. (1995), The steerable pyramid: A flexible architecture for multi-scale derivative computation, in *IEEE International Conference on Image Processing*, Washington DC, 444–7.
- Terzopoulos, D. & Rabie, T. F. (1997), Animat vision: Active vision in artificial animals, *VIDERE: Journal of Computer Vision Research*, **1**, 2–19.
- Tomasi, C. & Kanade, T. (1991), *Detection and Tracking of Point Features*, Technical Report, cmu-cs-91-132, Carnegie Mellon University.
- Young, R. A. (1986), Simulation of human retinal function with the Gaussian derivative model, in *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR'86)*, 564–9.