Moving Object Segmentation using Scene Understanding

A. G. Amitha Perera Glen Brooksby Anthony Hoogs Gianfranco Doretto
GE Global Research
One Research Circle, Niskayuna, New York 12309
{perera,brooksby,hoogs,doretto}@research.ge.com

Abstract

We present a novel approach to moving object detection in video taken from a translating, rotating and zooming sensor, with a focus on detecting very small objects in as few frames as possible. The primary innovation is to incorporate automatically computed scene understanding of the video directly into the motion segmentation process. Scene understanding provides spatial and semantic context that is used to improve frame-to-frame homography computation, as well as direct reduction of false alarms. The method can be applied to virtually any motion segmentation algorithm, and we explore its utility for three: frame differencing, tensor voting, and generalized PCA. The approach is especially effective on sequences with large scene depth and much parallax, as often occurs when the sensor is close to the scene. In one difficult sequence, our results show an 8-fold reduction of false positives on average, with essentially no impact on the true positive rate. We also show how scene understanding can be used to increase the accuracy of frame-to-frame homography estimates.

1. Introduction

We address the problem of detecting and segmenting moving objects in video taken from a translating, rotating and zooming sensor. We are particularly interested in detecting very small objects (less than 100 pixels in area) using as few frames as possible. These constraints are not met by most existing work. Many methods have been developed for moving object detection under the assumption that the camera is not translating and many frames are available, so that statistical background models can be learned [13]. This method has been extended to handle small sensor motions [16], but all of these methods violate our constraint of using as few frames as possible, since a sufficient number of frames must be observed to model the background.

Motion segmentation algorithms handle sensor translation without statistical background modeling by attempting to segment the scene into objects with consistent apparent motion. Many are also designed to operate on as few as two frames. However, most assume that objects are considerably larger than those we want to detect [8], and some do not exploit the spatial coherence of objects [15].

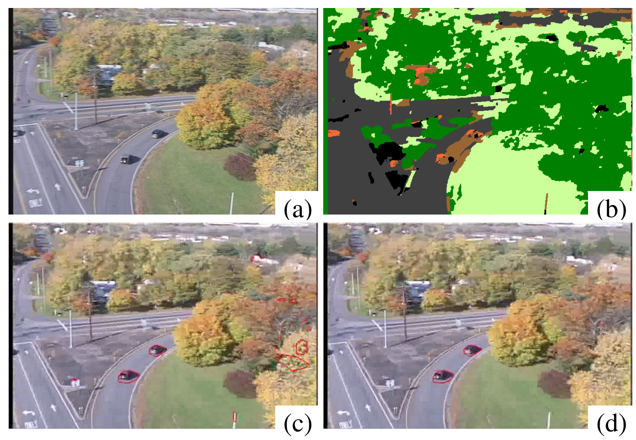

Figure 1. Comparative results of moving object detection with and without scene understanding. The video frame (a) was automatically classified into the label map (b). There are 6 classes: dark gray=road, orange=vehicle, brown=building, dark green=tree, light green=grass, and black=shadow. The result of a standard moving object detector is shown with a red outline in (c); note the false alarms in the trees. The result of using scene content to assist the same algorithm is shown in (d). False alarms are considerably reduced.

In this paper, we develop an integrated approach for segmenting independently moving objects by combining moving object detection with scene understanding. When scene knowledge is available, even with limited accuracy or fidelity, moving object detection and segmentation can be significantly improved, particularly given our constraints of small objects and rapid detection. Domain-specific scene knowledge is used to improve camera egomotion correction, to filter intermediate data such as corner points, and to filter detection false alarms. Thus detection sensitivity is greatly increased without increasing false alarms, leading to the detection of smaller objects in fewer frames.

In our current method, scene understanding is computed automatically on each video frame by categorizing each

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

pixel into one of K classes using a combination of region segmentation, texture and color [3]. The domain determines the class library; we have experimented primarily with aerial video, where the classes are {road, tree, grass, building, vehicle, shadow}. Our experiments show that pixel-wise classification accuracies of 65% are sufficient to reduce false alarms from 7.6 to 0.9 per frame on average.

An example result is shown in Figure 1. The scene content recognition image in Figure 1b is used to condition moving object detection in three ways, resulting in the detections in Figure 1d. First, homography computation is tied to the ground plane by discarding interest points near or embedded in "tree" pixels. Second, detected moving objects are discarded if they are not near or embedded in "road" pixels. Third, moving objects are discarded if they do not contain any "vehicle" pixels. This result illustrates how even a simple, fast (sequential) moving object detector can be effective on difficult conditions when enhanced with scene content.

To validate the framework, we have developed a straightforward motion segmentation algorithm based on previous work [4, 2]. The method is based on computing an intensity difference image between two frames after aligning them through a homography. While it meets our constraints of sensitivity and rapid detection, it also assumes that the scene is roughly planar, or that the camera is distant from the scene compared to scene depth. However, we demonstrate that scene understanding can compensate for these limitations when they are violated.

Virtually all motion segmentation algorithms can be enhanced using our framework. The primary requirement is that some form of local measurement or feature is used, such as corner points or correlation maxima. Even when this requirement is not met, simple spatial filtering based on scene content class can be used to discard highly unlikely movers, such as those embedded in dense trees. In preliminary experiments, we demonstrate the utility of scene understanding for two general motion segmentation algorithms, boundary tensor voting [9] and generalized principal component analysis (GPCA) [15].

A fundamental component of generating the frame-difference-based moving object detections is computing frame to frame homographies. This is done by matching pixels, corners, or other features between subsequent frames F_t and F_{t+1}. Typically, robust methods such as MLESAC [14] are used to find the set of features that provides the best mapping between frames. However, when the planar scene constraint is grossly violated, even robust estimation cannot recover the correct ground-plane homography. This paper also describes how the homography estimation can be improved by incorporating scene understanding.

The rest of this paper is organized as follows. Section 2 details the primary motion segmentation method with which we have experimented, including the homography estimation process used for frame stabilization. Section 3 describes the scene understanding algorithm that automatically classifies the pixels in each frame. Section 4 describes how scene understanding can be used to improve the performance of both motion segmentation and homography estimation algorithms. Section 5 demonstrates the value added by presenting experimental results on some difficult aerial video. Section 6 closes the paper with a summary and conclusions.

2. Moving Object Segmentation

To meet our constraints of detecting very small movers in as few frames as possible, we extended the image differencing method of [4]. We have also conducted preliminary studies with tensor voting [8] and GPCA as described in Section 5.

The first step in the frame differencing approach to motion segmentation is frame stabilization. For aerial video, this is posed as estimating the ground plane to ground plane homography between two frames. This homography allows the frames to be warped into a common coordinate system prior to image differencing. The homographies are also useful for defining a common coordinate system for the whole sequence, which is essential for long duration tracking [5].

To estimate the frame-to-frame homographies, we use the KLT tracker [12] to generate feature correspondences between frames. We then robustly estimate a homography from these correspondences, using MLESAC [14] as the robust loss function. (In the sequel, when we refer to homography estimation, we mean this robust estimation.) We follow [14] for the formulation. Given a set of correspondences $\mathcal{C}_t = \{(x'_i, x)\}$ between frame F_{t-1} and frame F_t, we define the probability distribution of a correspondence using a mixture of a Gaussian for the inliers and a uniform distribution for the outliers:

$$\mathrm{P}\big((x'_i, x_i)\big) = \gamma\, \mathrm{P_g}\big(d(x'_i, x_i)\big) + (1-\gamma)\frac{1}{v}, \qquad (1)$$

where $\mathrm{P_g}$ is a Gaussian density function, γ is the mixing parameter (estimated from the data via EM), d computes the symmetric transfer error, and v is a parameter specifying the width of the outlier window. The robust homography estimate is

$$\hat{H}_t = \arg\max \sum_i \log \mathrm{P}\big((x'_i, x_i)\big), \qquad (2)$$

and is computed using random sampling.

When the scene is mostly planar, the loss function discards the few correspondences on moving objects as outliers and the algorithm correctly estimates the ground plane transformation. When the scene has more depth, many of the features tracked by KLT may not be on the ground

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

plane, and the estimated homography plane may not be the ground plane. This is especially true when the scene contains highly textured, non-planar objects like trees, since they generate many more "trackable" features than the fairly uniform features such as roads and grass. Section 4.2 describes two novel approaches to alleviating this problem.

The motion at frame F_t is detected by warping frame $F_{t-\tau}$ to the coordinates of frame F_t using the homography $H_{t,\tau} = \prod_{j=t}^{t-\tau} \hat{H}_j$, subtracting the images, thresholding the difference, median filtering, and applying morphology:

$$I_{cc} = \mathit{Close}(\mathit{Median}(\mathit{Thresh}(T_D, F_t - H_{t,\tau}(F_{t-\tau})))).$$

I_{cc} is a binary image where "on" pixels indicate motion, and T_D is a fixed threshold determining motion detection sensitivity. We then find connected components, and filter by minimum component size S when possible (minimum and maximum object sizes are known a priori in many domains, including aerial video).

This algorithm still produces many false alarms when S is small because of parallax motion and noise. Hence we compute multi-frame persistence $Per(C)$ for connected component C based on the average persistence of its pixels. For each pixel i, we maintain an accumulator A_i over all frames. When i is included in any connected component on a given frame, A_i is incremented. Otherwise, A_i is decremented unless it is already 0. Then we compute

$$Per(C) = \frac{1}{\|C\|} \sum_{i \in C} A_i$$

and threshold the result to obtain moving objects. This method provides robustness with respect to individual pixels appearing in the difference image on a given frame, which tend to be noisy.

A key advantage of this approach is that reasonable results can be obtained by comparing only two frames incrementally as data is received (after a one-time lag of τ frames). The value of τ is set based upon the minimum object velocity to be detected; larger values are required to detect slower objects (in our experiments we use $\tau = 4$).

The method is fast, and works well for scenes where the scene depth is small relative to the distance from the camera. For scenes with more depth, the algorithm generates many false positives due to parallax caused by the camera motion. We reduce almost all of these false alarms by incorporating scene understanding, as described in Section 4.

3. Scene Understanding

The goal of the scene understanding algorithm is to classify each pixel on each video frame into one of K classes. We assume that the class library is complete, such that each pixel has one true class (or that "other" is a valid class). The class library is inherently domain-specific, but can be arbitrarily broad.

The scene understanding method is described elsewhere [6], and summarized here. In principle any scene content recognition algorithm could be used, particularly since the overall formulation is not sensitive to scene content accuracy. However, since accurate boundary localization should be helpful for segmenting small objects, we use an approach that first estimates image structure through dense region segmentation, then performs content recognition by classifying each region using perceptual features derived from the region graph. Frames are segmented into regions by minimizing a variant of the Mumford-Shah functional [11].

The region segmentation is derived from the mostly closed contours in the edge map which minimize the functional. Feature vectors are then computed to measure the characteristics of each region, such as area, intensity/color, perimeter, parallelism, etc., and its relation to its neighbors, such as contrast, relative area, relative parallelism, and so on [3].

Training consists of hand-labeling one or a few frames of a video and using those to create a set of labeled region feature vectors. These form the training set for a multi-class boosting classifier, based on [10] but with enhancements for exploiting the decision space of pairwise classifiers.

During classification, each video frame is first segmented into regions. Then, each region is classified after computing its feature vector, producing a single class label per region. All the pixels in a region are assigned the region's class label.

In our aerial video experiments, the complete class library consists of {road, tree, grass, building, vehicle, shadow}. Figure 2 shows the training images used. The training images are in the same geographical region and taken at roughly the same time as the tested video, but otherwise the training data and testing data are very different. Of course, none of the tested video frames were used in training. Example classification results are shown in Figure 1b and Figure 3f. The overall classification accuracy is 65% on this data, although this varies widely between classes, from 15% on vehicles to 83% on roads. This accuracy and variability is taken into account when scene content is used for moving object detection, and still provides significant overall improvement.

4. Using Scene Understanding

Scene understanding can be used in a number of ways to reduce or eliminate errors in motion segmentation and in frame stabilization. We describe some methods below, and postpone results illustrating the improvements to the next section.

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

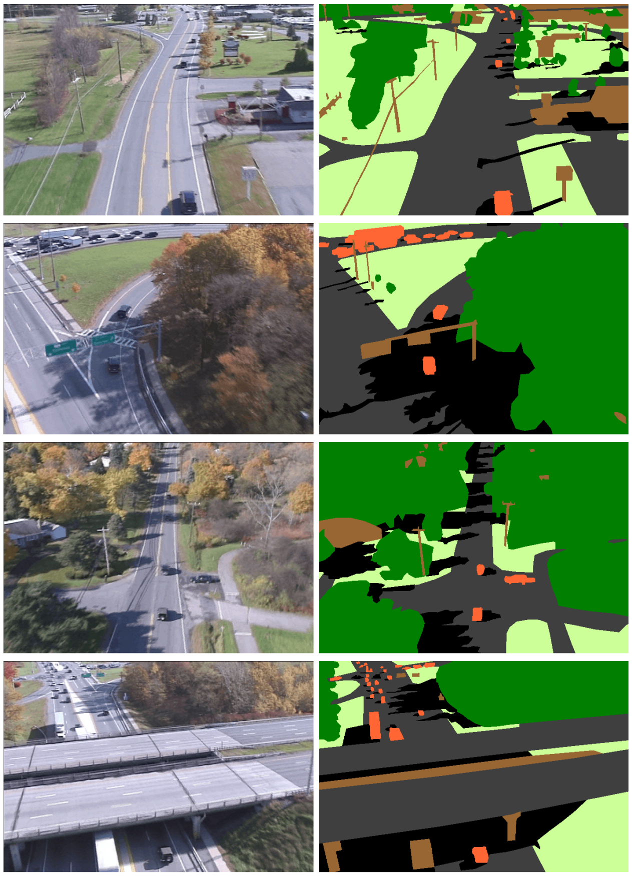

Figure 2. The images used to train the classifier. The left column shows the images, and the right column shows the corresponding hand-labeling. These training images were taken in the same geographical area and roughly the same time, but are otherwise very different from the test images.

4.1. Moving object detection

When using motion segmentation for moving object detection, scene understanding can be used to eliminate parallax-induced false positives by exploiting domain knowledge. For example, when tracking vehicles it is reasonable to presume that valid movers will appear on roadways and not in trees. It is also likely that apparent motion induced by parallax will be present largely in trees and will induce false positives. Thus, we define a subset K_M of the defined classes K for which moving objects are allowed.

Because of inaccuracy in both the scene understanding and moving object estimates, there must be allowance for a large number of incorrectly classified pixels. On a given frame, we determine a connected component, C, to be valid if:

$$\frac{|\{p_i \,|\, p_i \in C \wedge k_i \in K_M\}|}{|C|} > T_K$$

where k_i is the class of pixel p_i and T_K is a fixed threshold (typically 0.1 in our experiments). Our experience is that an accuracy of 65% in scene content classification is sufficient to remove the vast majority of false positives.

Similarly, we specifically exploit the vehicle class such that movers are only validated if they contain a minimum number of pixels classified as vehicles. This is feasible when vehicles are reliably detected.

This simple filtering by scene content can be applied to any motion segmentation algorithm. In addition, instead of a post-processing filtering step, motion segmentation algorithms can also use scene understanding as another feature to use during the segmentation process. In the tensor voting approach, for example, the vote by a token could be weighted on whether the receiving site is or is not in the same class as the token.

4.2. Homography estimation

A correct estimate of ground-plane homographies from aerial video requires that the data used in the estimation mostly arise from the ground plane. When the scene has a significant depth, however, much of the data comes from points above the ground plane. Scene understanding can be used here to determine which sections of the image are unlikely to be on the ground plane. For example, points in areas classified as tree are unlikely to be on the ground plane. We incorporate this domain knowledge into the correspondence probability (1) to obtain

$$\mathrm{P}_{\mathrm{su}}\big((x_i', x_i)\big) = \begin{cases} \frac{1}{v}, & x_i, \notin \{\text{grass, road, shadow}\} \\ \mathrm{P}\big((x_i', x_i)\big), & \text{otherwise.} \end{cases} \tag{3}$$

Using this in (2) gives much more accurate homography estimates, since the points in the trees are treated as outliers and have no effect on the estimate.

5. Experimental Results and Discussion

Homography estimation Figure 3 illustrates the effect of scene understanding on homography estimation. Figure 3a shows the feature points generated by the KLT tracker. Robustly estimating a homography from these points and using it to register frames for image differencing yields the moving objects shown in Figure 3b. Observe the false alarms on the road; they indicates the estimated homography did not correctly map the ground plane (which contains the road). To get a better estimate, we filter the points using the automatically generated scene understanding map shown in Figure 3f, by keeping only points on regions classified as grass, road or shadow. We are then left with the points in Figure 3c. Not all the above-ground points are removed because of errors in the scene understanding map. However, enough are removed to perform a robust homography estimation, and a greatly improved moving object detection, as shown in Figure 3d. Note that there are no longer

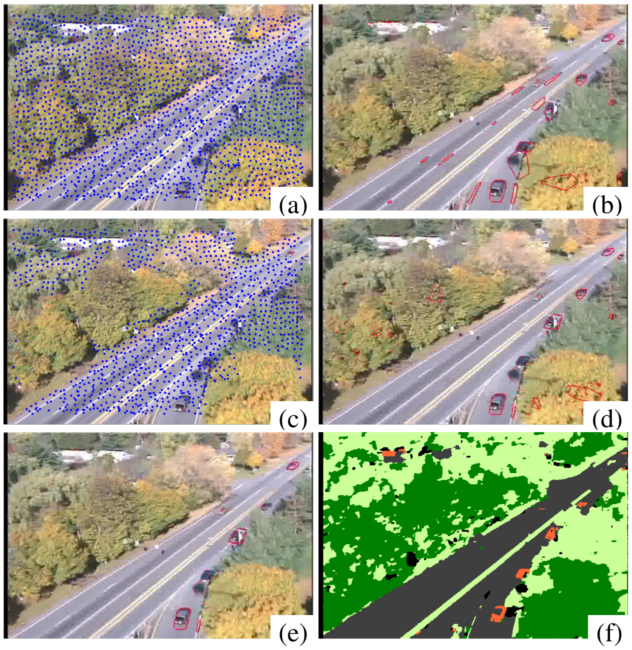

Figure 3. (a) The points used for homography estimation without scene understanding results in (b) a lot of false moving object detections in the trees and roads. (c) When scene understanding is used to improve the homography estimation, (d) the false alarms on the road vanish, but more parallax-induced false alarms appear in the trees. (e) When scene understanding is used also for moving object detection, most of the false alarms are reduced. (f) The scene understanding map for this frame.

any false alarms on the road, but there are additional false alarms on the trees on the left. This as expected, because the correct ground plane homography induces parallax in the trees. The estimate is improved even though the scene understanding map misclassifies some trees as grass. If we further filter the moving object detections by keeping only those objects on the road or in shadow, we obtain the much better moving object detection results shown in Figure 3e. The attached video "mod-comparison" shows these results on each frame. The top image on the video shows the results without scene understanding, and the bottom image shows the results with scene understanding. The points used for the homography estimation are displayed as blue points, and the detected moving objects are displayed with a red outline.

Moving object segmentation To analyze the contribution of scene understanding to moving object detection, we quantify the detection results using a hand-labeled database of moving vehicles as ground truth. Figure 4a shows the false positive count at each frame relative to this database; for clarity, Figure 4b shows a 10-frame moving average of the same data. The red curve is moving object detection without scene understanding used at any stage. Note how the false positive rate sharply increases about frame 200 as the vehicles near the corner and parallax induces motion in the trees. The green curve results when scene understanding is used for the homography estimation but not for moving object detection. This indicates a net increase of the parallax-induced movers in the trees compared to movers due previously to mediocre homography estimation. The blue and pink curves use scene understanding for both homography computation and moving object detection. In the former, moving objects are kept on road or shadow; in the latter, only road is allowed. Clearly, there is an enormous reduction in the false positive rate when scene understanding is used. The average false positive count per frame over the whole sequence drops from 7.6 to 0.9.

What the false positive count does not reflect, however, is the non-random nature of each false positive. The ground truth database only contains vehicles moving on the road. Many of the false positives correspond to consistently detected regions exhibiting parallax-induced motion: for example the tops of two telephone-poles are consistently detected. In one case, the false positive is the shadow of the helicopter on which the camera was mounted. This certainly appears as a moving object on the ground plane, but is classified as a false positive. Some of the false positives are caused by *split blobs*: sometimes the difference region from the front of the vehicle is not merged with the difference region at the back of the vehicle, resulting in two regions on one vehicle. We count this as one true positive and one false positive, thus increasing the false positive count. The attached video "mod-comparison" much better illustrates the non-random nature of many of the false positives. Figures 1c vs 1d and 3b vs 3e show frames illustrating the difference that scene understanding makes to moving object detection.

Figure 4c shows the true positive count at each frame, and Figure 4e the detection rate—the ratio of the true positive count to the true count. It shows that the false positive reduction discussed above does not sacrifice detection rate: the detection rate is essentially unchanged after introducing scene understanding. In the beginning of the sequence the detection rate is low because most of the moving objects are extremely small vehicles in the distance, which are difficult to detect in general. In our frame-differencing implementation, in particular, split blobs often cause each component to fail a size threshold, and the vehicle is not detected at all. When vehicles in the image are a little larger, from about frame 200 on, our detection rate is very high. The average detection rate over the whole sequence, including the tiny vehicles at the beginning, is 0.93 without scene understanding, and only drops to 0.91 with scene understanding, while the false positive count drops 8-fold. From frame 200 on, the averages are 0.95 without scene understanding to 0.92 with, still with an 8-fold drop in the false positive count.

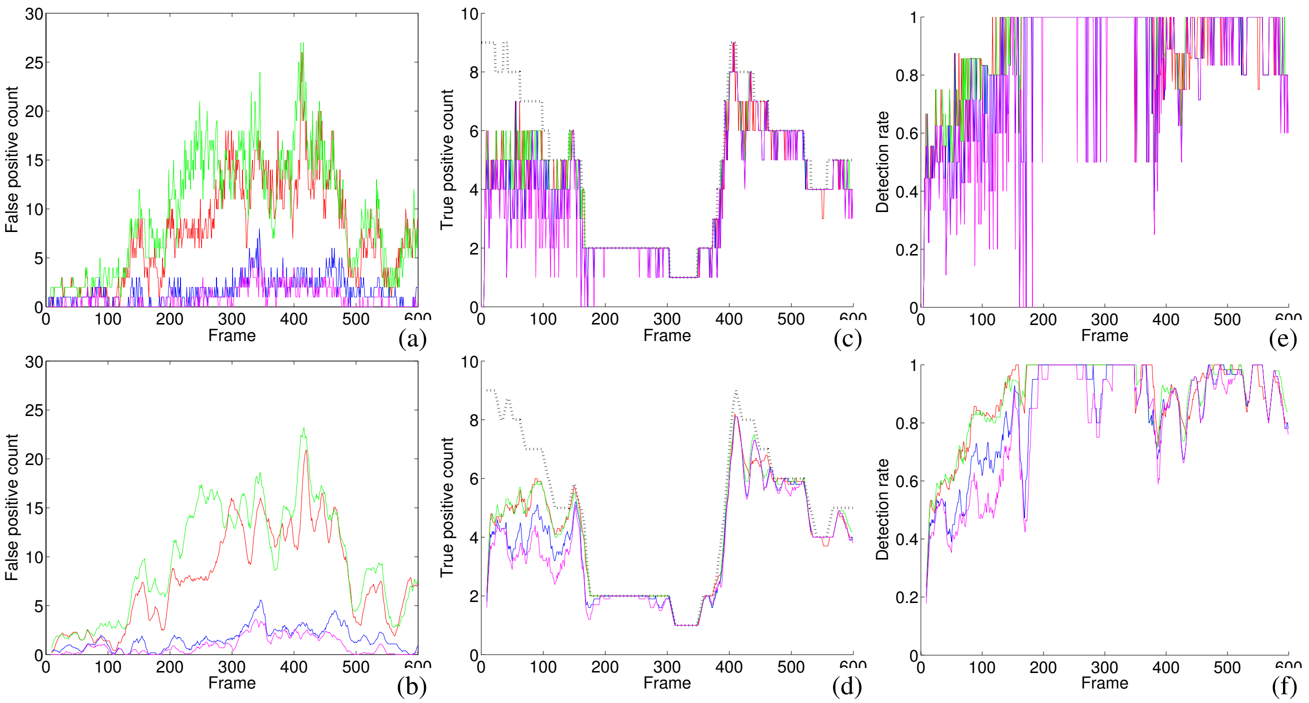

Figure 4. From left to right, the columns plot false positive count, true positive count, and detection rate at each frame. For clarity, the bottom row shows the plots of the top row after applying a moving average of 10 frames. The red line is the result of the basic frame-differencing MOD algorithm. The green line results when scene understanding is used for the homography estimation. The blue line results when it is used for both homography estimation and moving object detection. The pink line is similar to the blue line, except with a tighter filter on the moving object detection. The dotted black line in the middle column is the ground truth count of moving objects. See the text for a more detailed discussion.

Generalized PCA We performed some preliminary experiments with the GPCA approach described in [15], where the authors address the problem of motion segmentation from multiple views in the case of affine cameras. As input to the algorithm, we generated trajectories over 10 frames using the KLT tracker. We experimented with the trajectories as-are (in image coordinates) as well as with stabilized trajectories. In both cases, the results were disappointing, with the motion groups appearing to be random. We hypothesize that one reason for the failure is that the formulation in [15] does not have any spatial constraints, allowing the points widely distributed in the image to be grouped together. Moreover, in our data, the motion of the vehicles are described by very few points, with as little as four feature trajectories on a vehicle. This may not be enough to define a clear subspace, and so the motion of the vehicle may be lost in the noise. Using scene understanding to filter out most of the points in the trees and grass did not improve the result much. We conclude that the generalized PCA approach as currently formulated is not suited to our problem.

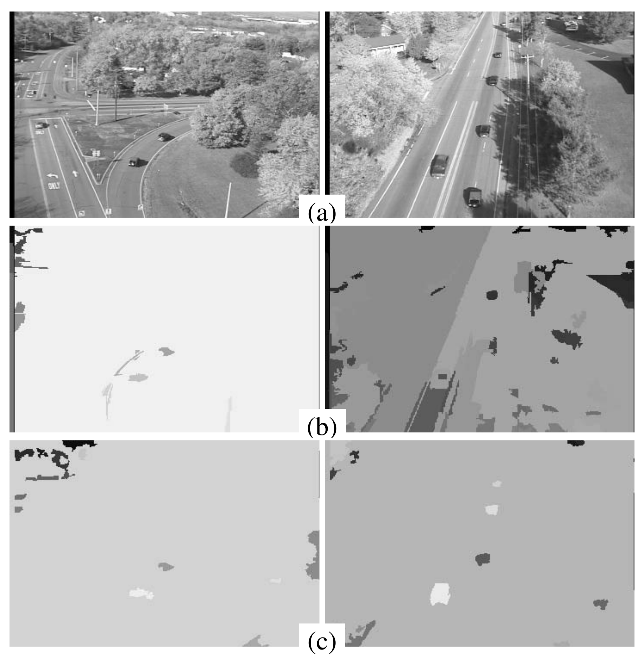

Figure 5. (a) shows the video frames. (b) and (c) show the motion segmentation result using EDISON segmentation and our scene understanding, respectively, as the initial segmentation.

Tensor voting The tensor voting approach to motion segmentation extends [8, 9] to a multi-frame estimation by introducing an additional time dimension, so that each token is represented by (x, y, t, v_x, v_y). Each token is represented as a 5D ball tensor, and each token then votes at other token's locations. The votes from a token are constrained to the tokens in the same region as itself. These regions are obtained from an oversegmentation of the reference frame, and serves to incorporate monocular cues into the voting process. The votes from tokens undergoing the same motion re-enforce each other, resulting in the most salient tensors clustered together by motion region in 5D space. The most salient tensors are kept, defining a sparse motion field. These tensors then vote for every pixel to generate final dense motion field. This algorithm can work with as little as two frames, but can also utilize multiple frames if they are available. The current approach uses the EDISON algorithm (based on [1, 7]) to perform the oversegmentation. In this experiment, the scene understanding image was used as an alternative segmentation, by creating regions from each connected component in the scene classification map having the same label. Figure 5b shows the resulting motion segmentation when using the EDISON segmentation to constrain the voting, and Figure 5c shows the much-improved result when scene understanding is used. The key advantage of the scene understanding segmentation is that it eliminates intra-object boundaries while preserving inter-object boundaries. This provides a greater chance for the tensors within an object to re-enforce each other, increasing the ability to correctly segment the motion regions.

6. Summary and conclusions

Detecting small, independently moving objects from a moving sensor is very difficult, particularly in the presence of heavy parallax-induced apparent motion. In this paper, we have demonstrated that scene understanding can be used to address these problems, even with a relatively simple and fast moving object detector. In particular, we have demonstrated good results with our approach where other, more complex, approaches fail.

More generally, scene understanding can be used to significantly improve many video analysis algorithms. Scene understanding provides spatial and semantic context at the pixel level, to be used by any higher level process; we have previously demonstrated its utility for tracking [5]. As we have shown in this paper, fusion with scene understanding can elevate a simple, ineffective algorithm to become a feasible solution to hard problems.

7. Acknowledgments

We wish to thank: René Vidal for providing his generalized PCA code; Changki Min and Gérard Medioni for running their algorithm on our data and providing results; and John Kaufhold and Roderic Collins for generating the scene understanding results.

This report was prepared by GE GRC as an account of work sponsored by Lockheed Martin Corporation. Information contained in this report constitutes technical information which is the property of Lockheed Martin Corporation. Neither GE nor Lockheed Martin Corporation, nor any person acting on behalf of either; a. Makes any warranty or representation, expressed or implied, with respect to the use of any information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or b. Assume any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method or process disclosed in this report.

References

[1] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002. 7

[2] R. Cutler, C. Shekhar, B. Burns, R. Chellappa, R. Bolles, and L. Davis. Monitoring human and vehicle activity using airborne video. In *28th Applied Imagery Pattern Recognition Workshop*, Washington, DC, 1999. 2

[3] A. Hoogs, R. Collins, R. Kaucic, and J. Mundy. A common set of perceptual observables for grouping, figure-ground discrimination and texture classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(4):458–475, Apr. 2003. 2, 3

[4] M. Irani and P. Anadan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(6):577–589, June 1998. 2

[5] R. Kaucic, A. G. A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *Proc. CVPR*, pages 990–997, 2005. 2, 7

[6] J. Kaufhold, R. Collins, A. Hoogs, and P. Rondot. Recognition and segmentation of scene content using region-based classification. In *IEEE International Conference on Pattern Recognition*, Hong Kong, 20–24 Aug. 2006. 3

[7] P. Meer and B. Georgescu. Edge detection with embedded confidence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(12):1351–1365, Dec. 2001. 7

[8] M. Nicolescu and G. Medioni. Layered 4D representation and voting for grouping from motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(4):492–501, Apr. 2003. 1, 2, 6

[9] M. Nicolescu and G. Medioni. Motion segmentation with accurate boundaries - a tensor voting approach. In *Proc. CVPR*, 2003. 2, 6

[10] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999. 3

[11] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Proc. CVPR*, 1996. 3

[12] J. Shi and C. Tomasi. Good features to track. In *Proc. CVPR*, pages 593–600, 1994. 2

[13] C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. CVPR*, pages 246–252, 1999. 1

[14] P. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *CVIU*, 78:138–156, 2000. 2

[15] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization and GPCA. In *Proc. CVPR*, volume II, pages 310–316, 2004. 1, 2, 6

[16] C.-S. C. Ying Ren and Y. K. Ho. Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, 24, 2003. 1

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 **IEEE**