Detecting Multiple Moving Objects in Crowded Environments with Coherent Motion Regions

Anil M. Cheriyadat[1,2], Budhendra L. Bhaduri[1], and Richard J. Radke[2]
[1]Computational Sciences and Engineering, Oak Ridge National Laboratory
Oak Ridge, TN, 37831 USA (cheriyadatam@ornl.gov,bhaduribl@ornl.gov)
[2]Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute,
Troy, NY 12180 (rjradke@ecse.rpi.edu) *

Abstract

We propose an object detection system that uses the locations of tracked low-level feature points as input, and produces a set of independent coherent motion regions as output. As an object moves, tracked feature points on it span a coherent 3D region in the space-time volume defined by the video. In the case of multi-object motion, many possible coherent motion regions can be constructed around the set of all feature point tracks. Our approach is to identify all possible coherent motion regions, and extract the subset that maximizes an overall likelihood function while assigning each point track to at most one motion region. We solve the problem of finding the best set of coherent motion regions with a simple greedy algorithm, and show that our approach produces semantically correct detections and counts of similar objects moving through crowded scenes.

1 Introduction

The inherent ability of our visual system to perceive coherent motion patterns in crowded environments is remarkable. Johansson [7] described experiments supporting this splendid visual perception capability, demonstrating the innate ability of humans to distinguish activities and count independent motions simply from 2D projections of a sparse set of feature points manually identified on human joints. When we conducted similar experiments with the video segments used in this paper, reducing each to a swarm of moving bright dots (automatically extracted features) against a dark background, human observers were easily able to detect and classify the moving objects. This motivated us to develop an automated system that can detect and count independently moving objects based on feature point trajectories alone.

We propose an object detection system that uses the locations of tracked low-level feature points as input, and produces a set of independent *coherent motion regions* as output. We define a coherent motion region as a spatiotemporal subvolume fully containing a group of point tracks; ideally, a single moving object corresponds to a single coherent motion region. However, in the case of many similar moving objects that may overlap from the camera's perspective, many possible coherent motion regions exist. Therefore, we pose the multi-object detection problem as one of choosing a good set of disjoint coherent motion regions that represents the individual moving objects. This decision is based on a track similarity measure that evaluates the likelihood that all the trajectories within a coherent motion region arise from a single object, and is made using a greedy algorithm. Figure 1 shows sample video frames with our algorithm's results overlaid.

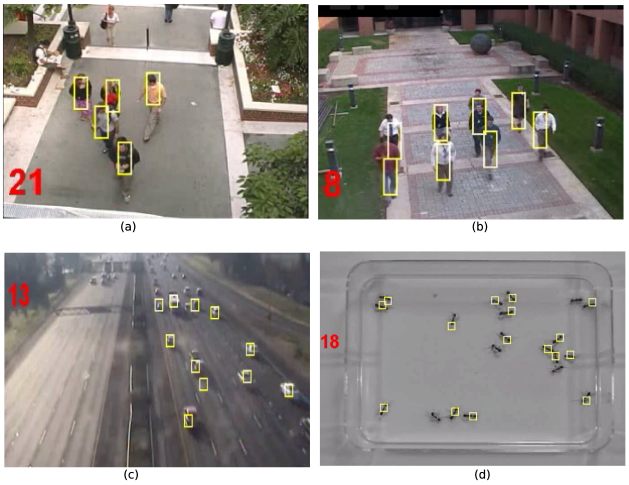


Figure 1: Sample multi-object detection results from our algorithm. The red number indicates the aggregate count of independently moving objects detected in the video up to the given frame.

With the exception of defining a single spatial bound-

*This work was supported in part by the US National Science Foundation, under the award IIS-0237516.

ing box that defines the expected object size, the proposed approach can be considered as a general purpose algorithm for detecting and counting large numbers of similar moving objects. Our algorithm neither uses object-specific shape models nor relies on detecting and assembling object-specific components. Camera calibration is not required. The proposed algorithm is computationally efficient, and well-suited for camera network applications that require sensors to distill video into compact, salient descriptions. We demonstrate that our system can localize and track multiple moving objects with a high detection rate.

The remainder of the paper is organized as follows. In Section 2, we review recent work on multi-object detection in video. In Section 3, we describe our framework for estimating a good set of disjoint coherent motion regions. Results obtained from several real video sequences are presented in Section 4. Section 5 concludes the paper with ideas for future work.

2 Related Work

Previous approaches to detecting multiple moving objects, often humans in particular, include methods based on complex shape models [6, 22], generative shape models based on low-level spatial features [18, 14], bag-of-features [20], low-level motion pattern models [17], body-part assembly models [21, 17, 13, 11] and low-level feature track clustering [4, 12, 5].

Gravila [6] proposed a shape exemplar-based object detection system. A shape exemplar database is constructed from a set of hundreds of training shapes and organized into a tree structure. The problem of matching a new image region against the exemplars on the tree nodes is posed in a Bayesian framework. The proposed object detection system addressed automated pedestrian detection from a moving vehicle. Similarly, Leibe et al. [8] proposed an approach for detecting pedestrians in crowded scenes that used a set of hundreds of ground-truth segmentations of pedestrians. These are used to build a bag-of-features type codebook for objects (based on squares centered at detected interest points). These local detectors are combined with global shape cues learned from training silhouettes into an MDL-type method to assign foreground pixels to human support regions using mean shift estimation. Viola et al. [20] proposed another bag-of-features approach to detect pedestrians in a video sequence. A detector is scanned over two frames of image sequence to extract features that relate to variations in intensity gradients within this spatiotemporal volume. The features thus extracted are used to train an AdaBoost classifier to detect walking people.

Song et al. [17] addressed the problem of detecting humans based on motion models. A perceptual model of human motion is constructed from training sequences. Labeled features on the human body are tracked, and the joint probability density function of these features' positions and velocities are used to model human motion. During detection, features are tracked over two consecutive frames and a person is detected if the features' positions and velocities maximize the posterior probability of the motion model.

Zhao and Nevatia [22] presented a key advance in finding people in crowds. People are modeled with shape parameters represented by a multi-ellipsoidal model, appearance parameters are represented by color histograms, and a modified Gaussian distribution is used to model the background. Candidates for head locations identified from the foreground blobs direct a Markov chain Monte Carlo (MCMC) algorithm to create new human hypotheses.

Another interesting approach [13, 11, 21] is the simultaneous detection and tracking of humans based on body part assembly. Body part detectors rely on spatial features and simple structural models to generate body part candidates. Body part candidates are assembled in a probabilistic framework, allowing the system to make hypotheses about probable human locations even in the presence of inter- and intra-object occlusions. Leordeanu and Collins [9] took an unsupervised learning approach based on the pairwise co-occurrences of parts to identify parts belonging the same object. Each object part is a collection of scale invariant feature points extracted from frames and grouped based on certain similarity conditions. Moving objects are identified by matching parts and estimating their pairwise co-occurrences.

Brostow and Cipolla [4] discussed scenarios in which crowds were so dense that background subtraction or model-based detection approaches would fail, since the crowd takes up most of the frame and there are few meaningful boundaries between entities. They proposed an unsupervised Bayesian algorithm for clustering tracked low-level interest points based entirely on motion, not on appearance. Using a spatial prior and a likelihood model for coherent motion, they obtained qualitatively encouraging results on crowded videos of people, bees, ants, and so on. Similarly, Rabaud and Belongie [12] addressed similar scenarios for counting moving objects. Under crowded situations, low-level feature extraction and tracking can result in fragmented and noisy feature-point trajectories. They proposed a trajectory conditioning strategy by propagating a spatial window along the temporal direction of each trajectory. New spatial coordinates for fragmented trajectories are obtained by averaging other trajectory coordinates inside this spatial window. Cheriyadat and Radke [5] proposed an algorithm for clustering feature point tracks in crowded scenes into dominant motions using a distance measure based on longest common subsequences. Vidal and Hartley [19] posed the point trajectory clustering as a problem of finding linear subspaces representing independent motion.

Point trajectories are projected to a five-dimensional space using the PowerFactorization method and moving objects are segmented by fitting linear subspaces to projected points under a generalized principal component analysis framework.

Tu and Rittscher [18, 14] took a different approach to crowd segmentation by arranging spatial features to form cliques. They posed the multi-object detection problem as one of finding a set of maximal cliques in a graph. The spatial features form the graph vertices, and each edge weight corresponds to the probability that features arise from the same individual. Their earlier work [18] dealt with overhead views and the spatial feature similarity measure was based on the assumption that the vertices lie on the circular contour of a human seen from above. In the later work [14], they used a a variant of the expectation-maximization algorithm for the estimation of shape parameters from image observations via hidden assignment vectors of features to cliques. The features are extracted from the bounding contours of foreground blob silhouettes, and each clique (representing an individual) is parameterized as a simple bounding box. Our work is most similar to these approaches; the coherent motion regions we propose are similar to maximal cliques in a graph. However, our work differs in the important sense that the coherent motion regions extend in time as well as space, enforcing consistency in detected objects over long time periods and making the algorithm robust to noisy or short point tracks. As a result of enforcing the constraint that selected coherent motion regions contain disjoint sets of tracks, our algorithm cannot be viewed as taking place on a static graph. If we were to pose our algorithm in a graph framework, the edges and their weights would change at every iteration of the greedy algorithm, as described further below.

We see our approach as a trade-off between algorithms that require object models and/or a moderate amount of prior information [6, 22, 21, 14, 13] and algorithms that possess no model for objects at all [4, 12, 5, 19]. Our algorithm operates directly on raw, unconditioned low-level feature point tracks, and tries to minimize a global measure of the coherent motion regions rather than approaching the problem with bottom-up clustering.

3 Algorithm Overview

3.1 Feature Point Tracks

Our algorithm begins with a set of low-level spatial feature points tracked over time through a video sequence. We define the ith feature point track by X^i:

$$X^i = \{(x_t^i, y_t^i), t = T_{init}^i, \ldots, T_{final}^i\}, i = 1, \ldots, Z. \quad (1)$$

Here, Z represents the total number of point tracks. The lengths of the tracks vary depending on the durations for which corresponding feature points are successfully tracked.

In our implementation, we first identify low-level features in the initial frame using the standard Shi-Tomasi-Kanade detector [16] as well as the Rosten-Drummond detector [15], a fast algorithm for finding corners. The low-level features are tracked over time using a hierarchical implementation [3] of the Kanade-Lucas-Tomasi optical flow algorithm [10]. The new features are tracked along with the existing point tracks to form a larger trajectory set. For trajectories that have initially stationary segments, we retain only the remaining part of the trajectory that shows significant temporal variations.

The user is also required to sketch a single rectangle that matches the rough dimensions of the objects to be detected in the sequence. Let the dimensions of this rectangle be $w \times h$.

3.2 Trajectory Similarity

We require a measure of similarity between two feature point tracks. If both X^i and X^j exist at time t, we define

$$\begin{aligned} d_t^x(i,j) &= (x_t^i - x_t^j)\left(1 + \max\left(0, \frac{|x_t^i - x_t^j| - w}{w}\right)\right) \\ d_t^y(i,j) &= (y_t^i - y_t^j)\left(1 + \max\left(0, \frac{|y_t^i - y_t^j| - h}{h}\right)\right) \\ D_t(i,j) &= \sqrt{d_t^x(i,j)^2 + d_t^y(i,j)^2} \end{aligned}$$

That is, if the features do not fit within a $w \times h$ rectangle, the distance between them is nonlinearly increased. Our expectation is that feature point tracks from the same underlying object are likely to have a low maximum D_t as well as a low variance in D_t over the region of overlap. Hence, we compute an overall trajectory similarity as

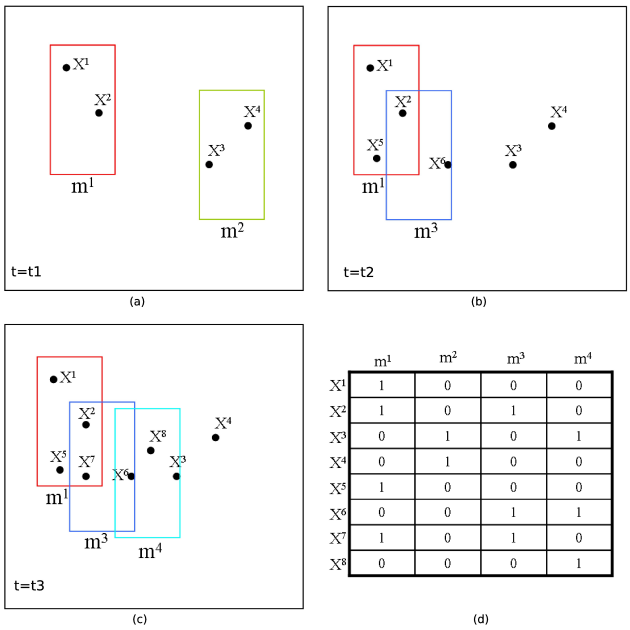
$$S(i,j) = \exp\{-\alpha * (\max(D_t(i,j)) + \mathrm{var}(D_t(i,j))\}, \quad (2)$$

where the maximum and variance are taken over the temporal region where both trajectories exist. For those trajectories where there is no overlap the similarity value is set to zero. The pairwise similarities are collected into a $Z \times Z$ matrix S. In our experiments below, we set α as 0.025.

3.3 Coherent Motion Regions

A key component of our algorithm is what we term a coherent motion region. Mathematically, a coherent motion region is a spatiotemporal subvolume that fully contains a set

of associated feature point tracks. In our case, each coherent motion region is a contiguous chunk of (x, y, t) space that completely spans the point tracks associated with it. Note that all the feature point tracks inside this region might not have complete temporal overlap. The set of all coherent motion regions can be represented by a binary $Z \times M$ matrix A indicating which point tracks are associated with each coherent motion region.



	m^1	m^2	m^3	m^4
X^1	1	0	0	0
X^2	1	0	1	0
X^3	0	1	0	1
X^4	0	1	0	0
X^5	1	0	0	0
X^6	0	0	1	1
X^7	1	0	1	0
X^8	0	0	0	1

Figure 2: (a) Illustrating candidate coherent motion regions for a given set of point tracks. (b,c) The coherent motion regions are updated by sliding a spatial window over each frame and adding/deleting columns of the matrix A. (d) The matrix A after time instant $t = t_3$.

In practice, we generate the coherent motion region matrix A by looping over the video frames, sliding a $w \times h$ spatial window over each frame. A column is added to A if a spatial window is found that contains a new subset of feature points not already represented as a column of A. If a spatial window is found that is a proper superset of an existing coherent motion region, the column of A corresponding to that region is deleted. In this way, only "maximal" coherent motion regions are recorded in the A matrix.

Figure 2 illustrates several example coherent motion regions for a set of point tracks. The dots in the figure denote the spatial locations of the feature points at a particular time instant. At time instant $t = t_1$, the point tracks form two coherent motion regions, shown as m^1 and m^2 in Figure 2a. At a later time instant $t = t_2$, the spatial location of point track X^5 triggers an update of coherent motion region m^1, because the newly formed group is a superset of the previous one. Track X^6 triggers the generation of a new coherent motion region m^3. Figure 2d shows the status of the matrix A after the time instant $t = t_3$ shown in Figure 2c. For the video sequences used in this paper, we found M to be in the range 3000-5000. Figure 3 shows an example video frame in which coherent motion regions are identified as red rectangles.

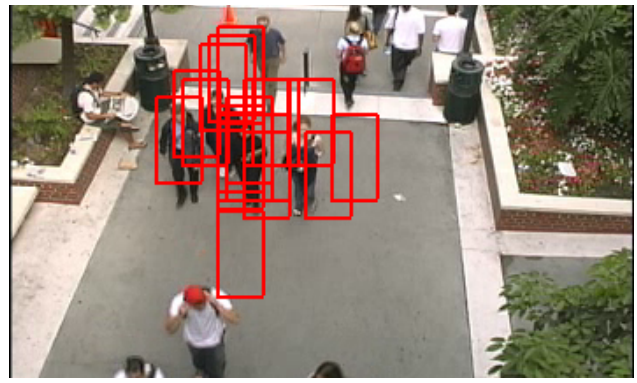


Figure 3: Coherent motion regions identified by the algorithm for an example frame.

We also associate an M-vector L with the set of coherent motion regions, indicating a "strength" related to the overall likelihood that the coherent motion region corresponds to a single object, created by

$$L(j) = A(j)^T S A(j) \tag{3}$$

where $A(j)$ is the jth column of A.

3.4 Selecting the Coherent Motion Region Subset

Conceptually, we would like to select a subset V of coherent motion regions that maximizes the sum of the strengths in V, subject to the constraint that a point track can belong to at most one selected coherent motion region. We use a greedy algorithm to estimate a good subset V as follows:

1. Set $V = \emptyset$.
2. Determine $j^* \notin V$ with maximal $L(j)$.
3. Add j^* to V and set $A(i, j) = 0$ for $j \neq j^*$ and all i such that $A(i, j^*) = 1$.
4. Recompute $L(j)$ for the remaining $j \notin V$ and repeat steps 2-4 until $L(j^*) < \beta \cdot \max_{j \in V} L(j)$. In our implementation, we used β's in the range 0.01-0.1.

This approach differs from a soft-assign approach of assigning tracks to cliques (e.g., extending [18]), since the greedy algorithm explicitly enforces the constraint that selected coherent motion regions be disjoint at each iteration.

Our approach could be viewed as starting with a highly connected graph in which edges are progressively removed and the sum of edge weights (likelihood) updated.

We observed that the best coherent motion regions are typically selected around objects that have been visible for some time. Since new feature point tracks are detected and tracked in each frame, a longer-duration object has a greater opportunity to gather feature tracks. Although some of an object's tracks may be lost or "picked up" by other objects, the remaining consistent tracks will yield a coherent motion region with high likelihood value. Selecting a high-likelihood coherent motion region in the greedy algorithm results in lowering the likelihood values of other coherent motion regions containing tracks that are part of the selected coherent motion region.

4 Results

We illustrate the performance of our algorithm on seven different video sequences typically used in multi-object tracking, featuring various types of objects including moving people, ants and vehicles. Although features related to object appearance and shape could be extracted from these sequences, we show that good performance can be obtained from feature point tracks alone.

The first video sequence, termed the *campus sequence*, shows a busy campus walkway (Figure 4a-d). The second video sequence, termed the *GE sequence*, shows a group of nine people walking together with significant intra-object occlusion during most of the sequence (Figure 4e-f). The third sequence, termed the *metro sequence*, was obtained from the PETS database [2] and shows people walking across a rail platform (Figure 4g-h). The fourth and fifth sequences, termed *highway-1 and highway-2*, show fast-moving vehicles on a highway (Figure 4i-j). During most of the *highway-2* sequence, the camera is swaying heavily in the wind, which is typical of such outdoor camera installations. The sixth video sequence, termed the *overhead sequence* and part of the CAVIAR database [1], shows four people walking through a corridor (Figure 4k). The seventh video sequence, termed the *ant sequence*, shows many ants moving randomly on a glass plate (Figure 4l-m).

The feature point trajectory generation is fast and comparable with the frame rate of the video sequence (i.e., ≈ 15 frames/sec). The multi-object detection algorithm is implemented in non-optimized Matlab code that takes less than 3 minutes of running time for each test video sequence reported here. Figure 5 shows comparisons between object counts made by a human observer and determined by the algorithm for each sequence. For the *campus*, *metro*, *highway-1* and *highway-1* sequences, both the counts were made for the spatial region identified by the red rectangle indicated in Figure 4a, g, i and j. The average false negative and average false positive detections across all the sequences are 12% and 4% respectively. We observed that the undercountings in Sequences 4 and 5 were due to the failure of the KLT tracker to generate feature points on several of the fast-moving vehicles.

As illustrated in Figure 6, the proposed general purpose object detection system yields comparable results with the object-specific trackers reported by Rittscher et al. [14] and Zhao et al. [22] for person detection and tracking, as well as with the general-purpose tracker of Brostow and Cipolla [4]. Our algorithm is able to overcome some of the difficulties in detecting individual objects that move in unison. To get a better appreciation of the detection results, we refer the reader to the following web link for the result videos: `http://www.ecse.rpi.edu/~rjradke/pocv/`.

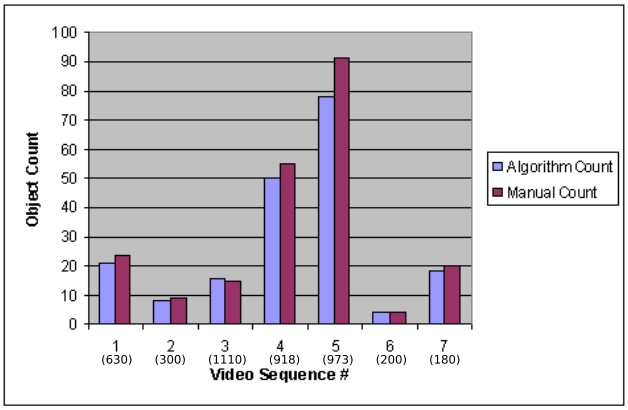


Figure 5: Object count determined by the algorithm compared with manual count for seven video sequences. The number in parentheses indicates the number of frames in each sequence.

5 Conclusions and Future Work

In this paper we introduce a simple approach based on coherent motion region detection for counting and locating objects in the presence of high object density and inter-object occlusions. We exploit the information generated by tracking low-level features to construct all possible coherent-motion-regions, and chose a good disjoint set of coherent motion regions representing individual objects using a greedy algorithm. The proposed algorithm requires no complex shape or appearance models for objects.

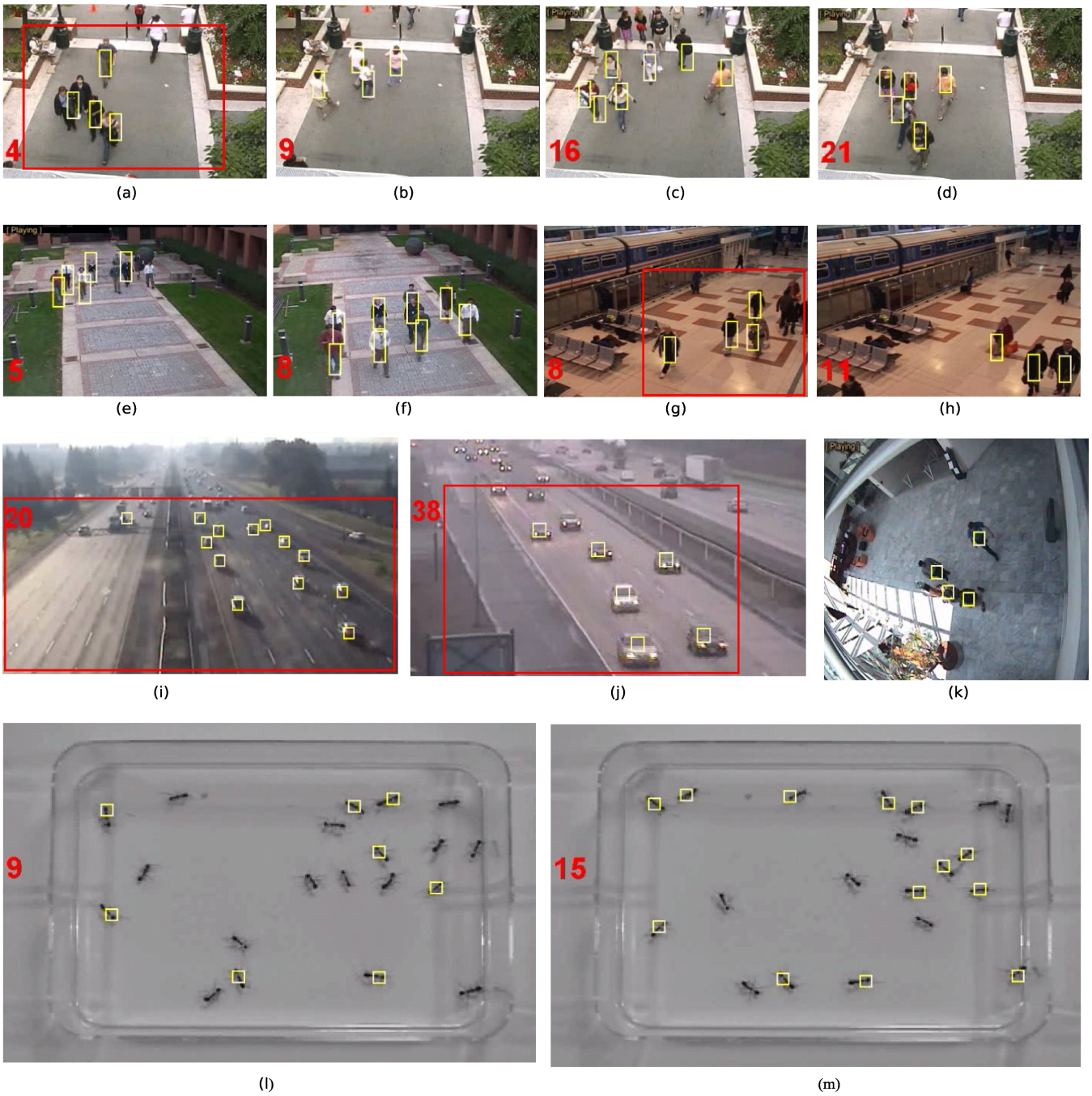


Figure 4: Sample results for seven different video sequences. The red number indicates the aggregate count of independently moving objects detected in the video up to the given frame. (a)-(d) Campus sequence. (e)-(f) GE sequence. Both sequences include substantial inter-object overlap and occlusion. (g)-(h) Metro sequence. (i) Highway 1 sequence. (j) Highway 2 sequence. (k) Overhead sequence. (l) Ant sequence. Note that several stationary ants are not detected by our algorithm, which is designed to find moving objects. The large red rectangle overlaid on subfigures (a),(g),(i) and (j) indicates the region within the frame where moving objects are detected and counted.

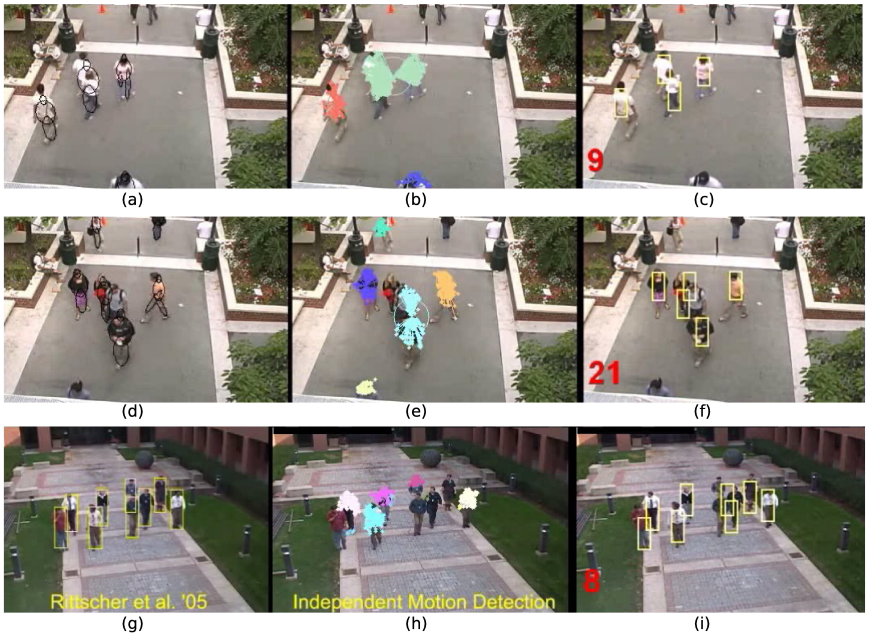


Figure 6: (a),(d) illustrate example results from Zhao and Nevatia's model based tracker [22], (b),(e) illustrate example results from Brostow and Cipolla's independent motion detector [4], and (c),(f) illustrate our algorithm's results for corresponding frames from the *campus sequence*. (i) illustrates an example result from Rittscher et al.'s object-specific tracker [14], (j) illustrates an example result from Brostow and Cipolla's independent motion detector, and (k) illustrates our algorithm's result for one frame of the *GE sequence*.

False positives in our algorithm would occur in the presence of heterogeneous object classes. For example, a person pulling a large cart might be counted as two moving persons, or a large truck may be counted as two cars. False negatives occur when very few feature points are identified on an object, resulting in very low likelihood values and hence undetected objects.

Possible extensions of this work include the incorporation of other cues such as appearance which could allow the system to maintain class-specific counts (e.g., cars vs. pedestrians or adults vs. children).

6 Acknowledgements

Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC for the U. S. Department of Energy under contract no. DEAC05-00OR22725.

This manuscript has been authored by employees of UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. Accordingly, the United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

References

[1] CAVIAR database. In *http://homepages.inf.ed.ac.uk/rbf/CAVIAR/*, 2004.

[2] PETS benchmark data. In *Proc. of IEEE Workshop on Performance Evaluation and Tracking and Surveillance*, 2007.

[3] G. Bradski. OpenCV: Examples of use and new applications in stereo, recognition and tracking. In *Proc. of International Conference on Vision Interface*, 2002.

[4] G. J. Brostow and R. Cipolla. Unsupervised Bayesian detection of independent motion in crowds. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 594–601, 2006.

[5] A. Cheriyadat and R. Radke. Automatically determining dominant motions in crowded scenes by clustering partial feature trajectories. In *Proceedings of the First ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC-07)*, September 2007.

[6] D. Gavrila. Pedestrian detection from a moving vehicle. In *Proceedings of the 6th European Conference on Computer Vision*, pages 37–49, 2000.

[7] G. Johansson. Visual motion perception. *Scientific American*, 14:76–78, 1975.

[8] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[9] M. Leordeanu and R. Collins. Unsupervised learning of object features from video sequences. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2005.

[10] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[11] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Proceedings of the European Conference on Computer Vision*, pages 69–82, 2004.

[12] V. Rabaud and S. Belongie. Counting crowded moving objects. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 705–711, 2006.

[13] D. Ramanan and D. A. Forsyth. Finding and tracking people from the bottom up. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 18–20, 2003.

[14] J. Rittscher, P. Tu, and N. Krahnst. Simultaneous estimation of segmentation and shape. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 486–493, 2005.

[15] E. Rosten and T. Drummond. Machine learning for high speed corner detection. In *Proc. of European Conference on Computer Vision*, 2006.

[16] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[17] Y. Song, X. Feng, and P. Perona. Towards detection of human motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13–15, 2000.

[18] P. Tu and J. Rittscher. Crowd segmentation through emergent labeling. In *Proc. ECCV Workshop on Statistical Methods in Video Processing*, 2004.

[19] R. Vidal and R. Hartley. Motion segmentation with missing data using power factorization and GPCA. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[20] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc of IEEE International Conference on Computer Vision*, pages 734–741, 2003.

[21] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 2007.

[22] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413, 2004.