2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance

Affinity Propagation Feature Clustering with Application to Vehicle Detection and Tracking in Road Traffic Surveillance

Jun Yang[†‡], Yang Wang[†‡], Arcot Sowmya[†], Bang Zhang[†‡], Jie Xu[†‡], Zhidong Li[†‡]
[†]School of Computer Science and Engineering, the University of New South Wales
[‡]National ICT Australia
{jun.yang, yang.wang, bang.zhang, jie.xu, zhidong.li}@nicta.com.au
sowmya@cse.unsw.edu.au

Abstract

In this paper, we investigate the applicability of the newly proposed data clustering method, affinity propagation, in feature points clustering and the task of vehicle detection and tracking in road traffic surveillance. We propose a model-based temporal association scheme and novel preprocessing and postprocessing operations which together with affinity propagation make a quite successful method for the given task. Our experiments demonstrate the effectiveness and efficiency of our method and its superiority over the state-of-the-art algorithm.

1. Introduction

Vehicle detection and tracking is a fundamental task in modern intelligent traffic systems. In the computer vision community, many methods have been proposed in the past two decades. Based on their technical approaches, they can be roughly divided into five categories: background subtraction [19], foreground segmentation [13, 25], supervised-learning-based detector [20, 9], edge-model matching [18, 26, 8, 33, 17, 11], feature points tracking and clustering [23, 15, 3, 2, 1, 24, 21, 30, 16, 14].

Feature points clustering has long been used in vehicle detection and tracking, and some promising results have been achieved. There are roughly two classes of approaches: one is from a top-down graph-partitioning perspective [23, 15, 24], the other features a bottom-up multi-level hierarchical clustering scheme [2, 1, 21, 16]. McLauchlan et al [23] represent the feature points as an fully connected graph and use a greedy procedure to gradually remove the edges according to some predefined heuristic criteria. The remaining subgraphs after edge removal are considered as vehicle candidates. Using the same graphical representation, Kanhere et al [15] run normalized-cut algorithm on the graph to automatically segment it into partitions. Du and Piater [3] instantiate and initialize clusters using the EM algorithm, followed by merging overlapping clusters and splitting spatially disjoint ones. Kanhere and Birchfield [14] find stable features heuristically and then group unstable features by a region growing algorithm. Yang et al [30] pose the feature clustering problem as a general MAP problem and find the optimal solution using MCMC.

Affinity propagation [5] is a recently proposed data clustering method which can find clusters with much lower error in less than one-hundredth the mount of time than other clustering methods. It has since been used in many diverse fields such as computer vision [29, 4, 28, 6, 7, 22, 34, 10], image coding [12], speech recognition [32], data mining [31], etc.

In this paper, we propose a vehicle detection and tracking method based on affinity propagation feature clustering. The rest of this paper is organized as follows. First we revisit the affinity propagation algorithm in section 2. The similarity function customized for our task is given in section 3. We then discuss the proposed temporal association scheme in section 4, and preprocessing and postprocessing steps in section 5. We present and analyze the experimental results in section 6. The paper concludes in section 7.

2. The Affinity Propagation algorithm

Affinity propagation [5] is essentially a special case of loopy belief propagation algorithm specifically designed for data clustering. Taking as input a matrix S of real-valued similarities between data points, affinity propagation simultaneously considers all data points as potential exemplars (cluster centers), and works by recursively exchanging real-valued messages between data points until a good set of exemplars and corresponding clusters gradually emerges.

The similarity $S(i, k), i \neq k$ indicates how well data point k is suited to be the exemplar for data point i, while $S(k, k)$ is referred to as "preference" and data points with

978-0-7695-4264-5/10 $26.00 © 2010 IEEE
DOI 10.1109/AVSS.2010.40

higher preference values are more likely to be chosen as exemplars. The preference thus serves as a parameter for controlling the number of identified clusters. If all data points are equally likely to be exemplars a priori, the preferences should be set to a common value, usually being the median of input similarities. For N data points, the affinity propagation algorithm can be viewed as a searching process which tries to maximize the net similarity $\mathcal{S} = \sum_{i=1}^{N} S(i, L(i))$, or to minimize the energy function $\mathcal{E} = -\mathcal{S}$, where $L(i)$ is the index of the exemplar point i belongs to.

Two types of messages are exchanged between data points, each representing a different kind of competition. Point i sends point k the "responsibility" $R(i,k)$ which reflects the accumulated evidence for the suitability of point i choosing point k as its exemplar, considering other potential exemplars other than k. Point k sends point i the "availability" $A(i,k)$ which reflects the accumulated evidence for the suitability of point k serving as the exemplar for point i, considering the support from other points who choose point k as their exemplar. From a probabilistic point of view, $S(i,k)$ can be interpreted as the log-likelihood of data point i given that its exemplar is point k, and $R(i,k)$ and $A(i,k)$ can be viewed as log-probability ratios. At any iteration of the message-passing procedure, the sum $R(k,k) + A(k,k)$ can be used to identify exemplars, and the corresponding clusters can be calculated. The whole procedure terminates after a predefined number of iterations, or if the decided clusters have stayed constant for some iterations. A summary of the affinity propagation algorithm is given in Algorithm 1.

Compared to other clustering methods, affinity propagation has several advantages: (1) Simplicity and speed. The message update rules involve only simple and local computations which are easy to implement, and it does not need multiple restarts to yield a good solution. (2) Accuracy and robustness. Rather than requiring the number of clusters be prespecified and an initial set of exemplars chosen, it simultaneously considers all the data points by passing soft messages around and automatically generates the appropriate clustering configuration, avoiding unlucky initializations and hard decisions. (3) Generality and flexibility. It can handle arbitrary pairwise similarity functions, being metric or nonmetric.

Algorithm 1: Affinity Propagation

```
Input: similarity matrix $S$, $maxits$, $convits$, $\lambda$
Output: exemplar index vector $L$
$iter \leftarrow 0, conv \leftarrow 0, A \leftarrow 0, R \leftarrow 0, L \leftarrow -\infty$
repeat
    $iter \leftarrow iter + 1$
    $OL \leftarrow L$
    $OR \leftarrow R$
    forall $i, k$ do
        $R(i,k) \leftarrow S(i,k) - \max_{k':k' \neq k} \{A(i,k') + S(i,k')\}$
    $R \leftarrow (1-\lambda) * R + \lambda * OR$
    $OA \leftarrow A$
    for $i \neq k$ do
        $A(i,k) \leftarrow \min \left\{0, R(k,k) + \sum_{i':i' \notin \{i,k\}} \max\{0, R(i',k)\}\right\}$
    forall $k$ do
        $A(k,k) \leftarrow \sum_{i':i' \neq k} \max\{0, R(i',k)\}$
    $A \leftarrow (1-\lambda) * A + \lambda * OA$
    foreach $k$ do
        if $R(k,k) + A(k,k) > 0$ then $L(k) \leftarrow k$
    foreach $k$ do
        if $L(k) \neq k$ then $L(k) \leftarrow \arg\max_{i:L(i)=i} S(k,i)$
    if $L = OL$ then
        $conv \leftarrow conv + 1$
    else
        $conv \leftarrow 1$
until $iter > maxits$ or $conv > convits$
```

3. Similarity function

Our goal is to cluster a set of individually tracked feature points into separate vehicle hypotheses. Let f_i^t represent the image position of feature point i at time t and $f_i^{\tau_i:t}$ its trajectory in image space. With camera calibration and assuming that all the feature points are at a certain height (e.g. half the height of a typical vehicle) in the real world, for feature point i the corresponding real world position $\mathcal{F}_i^t$ and trajectory $\mathcal{F}_i^{\tau_i:t}$ can be calculated. In terms of separating moving objects, there are many spatial and temporal cues which have been used in the past and current research. With adaptation to feature points grouping, we utilize the following similarity function:

$$S(i,k) = -\alpha_s d_s(i,k) - \alpha_m d_m(i,k) - \alpha_b d_b(i,k), \quad (1)$$

which consists of three terms:

$d_s(i,k)$ measures the spatial closeness between point i and point k and reflects the rationale that close points are likely to belong to the same vehicle:

$$d_s(i,k) = ||\mathcal{F}_i^t - \mathcal{F}_k^t||. \quad (2)$$

$d_m(i,k)$ models the motion similarity between the two trajectories with the intuition that feature points from the same vehicle should have similar motion patterns:

$$d_m(i,k) = \max_{j=\max\{\tau_i,\tau_k\}}^{t} ||(\mathcal{F}_i^t - \mathcal{F}_i^j) - (\mathcal{F}_k^t - \mathcal{F}_k^j)||. \quad (3)$$

$d_b(i,k)$ calculates the number of background pixels between the two points and encodes the belief that points which are more separated by background are less likely to come from the same vehicle:

$$d_b(i,k) = \sum_{f=f_i^t}^{f_k^t} \delta\big(B^t(f) = 1\big), \quad (4)$$

where B^t is the background mask at time t.

4. Temporal association

Temporal association is needed for two purposes. The first is to track vehicles for further applications. The second is to solve inconsistent clustering results from the two consecutive frames caused by unstable image features resulting from sensor noise, camera shake, blur, lighting change, etc. We use a temporal association scheme based on vehicle model fitting.

First we have a 3D cuboid vehicle model built from typical vehicle dimension data. With known lane direction and camera parameters, the mapping of the 3D vehicle model to the 2D image plane is determined by the position of the model on the ground plane. Suppose $C_i^t = \{f_{1_i}^t, f_{2_i}^t, \ldots, f_{n_i}^t\}$ is a feature cluster from time t. By projecting the 3D model from the real world position of the coordinate center of C_i^t, we get a 2D vehicle shape mask $M_{C_i^t}$ at the corresponding image position. A model fitting criteria is defined for C_i^t based on the number of feature points from C_i^t falling into $M_{C_i^t}$:

$$fit(C_i^t) = \frac{\sum_{f \in C_i^t} \delta\big(M_{C_i^t}(f) = 1\big)}{|C_i^t|}. \qquad (5)$$

Based on the smooth motion assumption, for a vehicle, its spanned areas at time t and $t-1$ should observe a tiny displacement and a big overlap, and the union of the two areas should still closely resemble a vehicle shape. Thus our association rule is, C_i^t is associated with C_k^{t-1} that is the largest cluster at time $t-1$ whose union with C_i^t fits the vehicle model reasonably well:

$$k = \arg\max_j \big\{|C_j^{t-1}| \big| fit(C_i^t \cup C_j^{t-1}) > \beta_a\big\}. \qquad (6)$$

C_i^t is then merged into C_k^{t-1} and the new cluster inherits the vehicle label of C_k^{t-1}. For each unassociated C_i^t, we remove its features which were already clustered at time $t-1$. The resulting cluster is denoted as $\tilde{C_i^t}$, and is assigned a new vehicle label if it satisfies $fit(\tilde{C_i^t}) > \beta_n$ and $|\tilde{C_i^t}| > \gamma_n$.

By incorporating an empirical vehicle model in finding the temporal correspondences instead of using schemes that simply count number of shared feature points, we introduce some semantics into the process, and avoid undesirable associations that could lead to cluster drift, dilation and other meaningless results.

5. Preprocessing and postprocessing

Three preprocessing steps are required before applying our method: (1) Offline camera calibration by assuming a planar road surface. (2) For the whole sequence, build a reference background image bg using simple image averaging; For each frame I^t, generate the background mask B^t using simple arithmetic and logical operations ($B^t = \neg(|I^t - bg| > \epsilon)$). (3) For each frame, detect and track KLT [27] feature points, and prune the ones falling in the background region or out of ROI.

After feature clustering and temporal association, we run two postprocessing steps to reinforce a plausible and desirable output: (1) Cluster shape maintenance. For each cluster C_i^t, we remove its member feature points that are outside the vehicle shape mask $M_{C_i^t}$ or inside the convex hull of another cluster. This accounts for the situation of outlier feature points that are either generated by noisy signal or wrongly clustered particularly along the occlusion boundaries, as well as the unfortunate case of having clustered the feature points on two separating vehicles. (2) Hypotheses verification by dropping small noise-generated clusters whose sizes are below a predefined threshold γ_d.

6. Experiments

We use three diversified traffic video sequences for the purpose of evaluation. The three sequences are from different locations, different camera installations and different weather conditions. *city1* and *city2* are two challenging sequences of a busy urban intersection with *city1* recorded with a low-height off-axis camera on a cloudy day and *city2* with a low-height on-axis camera on a rainy day. They both contain lots of complicated traffic events during traffic light cycles such as the typical process of vehicles decelerate, merge, queue together, then accelerate and separate. The low height nature of the cameras brings extra difficulties to the detection task. *city3* is recorded near a busy urban bridge with a medium-height off-axis camera under a sunny weather. For each sequence, the groundtruth is manually labeled at approximately ten frames apart. Refer to Table 1 for detailed sequence information.

We compare the performance of our method to the normalized cut approach [15]. We implement both methods in unoptimized C code and run them on a Pentium IV 3.0 GHz machine. In our normalized cut implementation: (1) We use the same spatial and temporal cues, and the pairwise affinity between point i and point k is defined as $A(i,k) = \exp S(i,k)$. (2) For the recursive top-down bipartition process, we adopt a model-based termination criteria rather than the original one which is based on the normalized cut value. The partitioning of graph G terminates if $fit(G) = 1$. (3) All the feature points are assumed a fixed height rather than relying on the original novel height estimation procedure which is found to be unstable in our experiments. (4) The same temporal association scheme, preprocessing and postprocessing steps are applied. For affinity propagation, we set the maximum number of allowed iterations $maxits$ to 1000, the number of iterations for deciding convergence $convits$ to 10, the damping factor λ to 0.5, the preferences $S(k,k)$ to the median of the input sim-

sequence name	camera position	camera height	weather condition	sequence length	labeled frames	labeled vehicles	TPR at 10% FPR				running time (ms/frame)			
							NCut		AP		NCut		AP	
							v1	v2	v1	v2	v1	v2	v1	v2
city1	off-axis	low	cloudy	2016	200	2035	0.65	0.61	0.66	0.68	85	136	82	112
city2	on-axis	low	rainy	2016	200	2592	0.67	0.66	0.73	0.73	108	244	101	182
city3	off-axis	medium	sunny	1366	136	1509	0.42	0.40	0.45	0.44	64	80	63	76

Table 1: Sequence and performance information. TPR stands for true positive rate, and FPR stands for false positive rate.

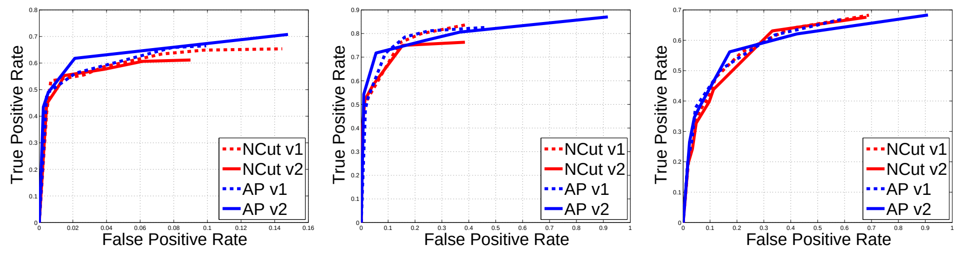

Figure 1: ROCs on the three sequences: from left to right, *city1*, *city2* and *city3*.

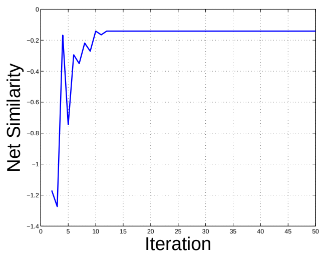

Figure 2: Convergence.

ilarities S. All the other parameters in the two algorithms are set by trial-and-error to achieve the best performance.

To facilitate the analysis of the effectiveness and efficiency of our method, we designed two versions for each of the two methods. We label the individual vehicle lanes for each sequence beforehand. The first version works on each lane separately, referred as v1. While the second version works on all the lanes combined, referred as v2. The ROC curves of the two methods each with two versions on the three sequences are shown in Fig. 1 (please note the different limits of the vertical axis for the plots). The performance calculation is based on the comparison with hand-labeled groundtruth on a strict frame-by-frame basis. From the curves, we can see that our method performs either comparably or better than normalized cut (see also Table 1). For both methods, the performance on *city1* is worse than that on *city2* because of the more frequent inter-lane occlusions for off-axis cameras, while the performance on *city3* is worse than that on both *city1* and *city2* because of the lower video quality of *city3* and specific difficulties in sunny weather such as shadow cast by vehicles, trees, etc (see Fig. 5). Generally speaking, for each method, the performance of v1 is better than or equal to that of v2 since by treating each lane individually we actually introduce some prior knowledge about separating vehicles from different lanes. However, *city1* gives an exception because in off-axis cameras feature points from one vehicle usually spill over two or even more lanes, thus making lane separation a disadvantage rather than an advantage. And it seems that affinity propagation can benefit much more from considering all the lanes together than normalized cut can. Fig. 3 and Fig. 4 give an example where normalized cut does not work well while affinity propagation does. In this situation, the number of feature points on a faraway vehicle is much less than that on a vehicle close to the camera. In trying to regularize the sizes of the partitions, normalized cut segments the feature set prematurely and ends up with many fragmented clusters. Affinity propagation avoids premature decisions by considering all the feature points simultaneously at all times. We can see from Fig. 4 that affinity propagation finally reaches the correct result even though the identified exemplar points do not necessarily have a physical meaning. Fig 2 plots the corresponding value of net similarity at each iteration for this example, and it takes about only ten iterations to converge. In terms of running speed, our method is faster than normalized cut as shown in Table 1. For each method, the two versions also give us an idea of how the algorithm scales with problem size. We see from Table 1 that the running time of normalized cut increases much faster with problem size than that of our method does. Fig. 5 gives some example detection results of our method on the three sequences and Fig. 6 shows some example vehicle trajectories generated.

7. Conclusions

In this paper, we investigate the feasibility of using affinity propagation based feature clustering for vehicle detection and tracking in road traffic surveillance, and show its distinct advantages over the state-of-the-art method.

In our future work, we will exploit three directions of improvement: the first is to incorporate prior knowledge (such as the distance to the center of the lane) into the setting of preference values; the second is to fuse information from features other than interest points to improve the accuracy and robustness of the clustering results; the third is to develop a spatial-temporal affinity propagation framework to

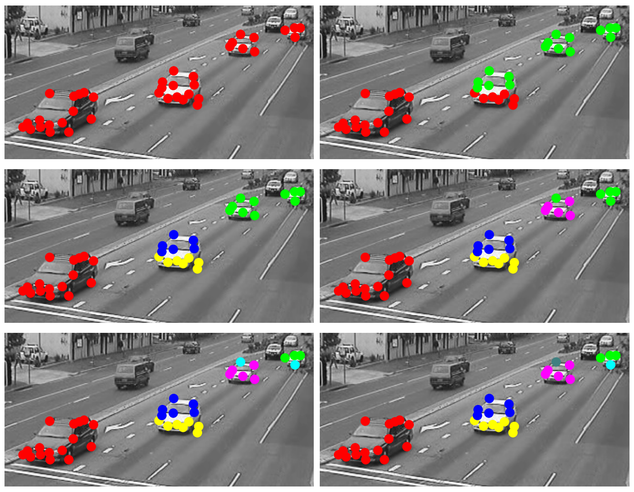

Figure 3: A working example of normalized cut feature clustering. From left to right, top to down, the images show the recursive partitioning process.

unify detection and tracking.

Acknowledgement

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

The authors thank Dr. William Uther for useful discussions and RTA (Roads and Traffic Authority), NSW for providing the video sequences.

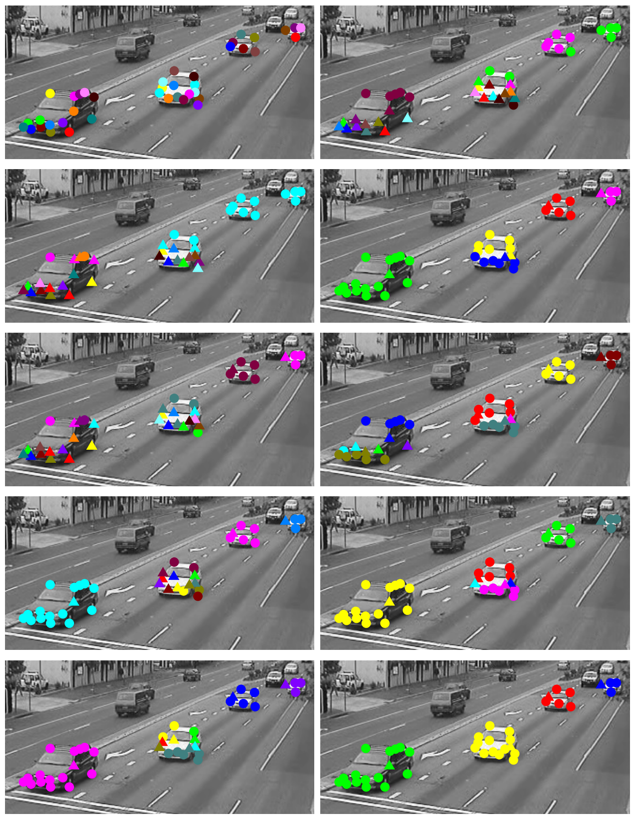

Figure 4: A working example of affinity propagation feature clustering. From left to right, top to bottom, the images show the clustering results at each iteration up to convergence. Identified exemplars are represented by triangles.

References

[1] G. Antonini and J.-P. Thiran. Counting pedestrians in video sequences using trajectory clustering. *TCSVT*, 16:1008–1020, 2006. 1

[2] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *CVPR*, pages 594–601, 2006. 1

[3] W. Du and J. Piater. Tracking by cluster analysis of feature points using a mixture particle filter. In *AVSS*, pages 165–170, 2005. 1

[4] D. Dueck and B. Frey. Non-metric affinity propagation for unsupervised image categorization. In *ICCV*, pages 1 –8, oct. 2007. 1

[5] B. J. Frey and D. Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315(5814):972–976, 2007. 1

[6] Y. Fu, Z. Li, X. Zhou, and T. Huang. Laplacian affinity propagation for semi-supervised object classification. In *ICIP*, volume 1, pages I –189 –I –192, 16 2007-oct. 19 2007. 1

[7] Y. Gao and Q.-H. Dai. Shot-based similarity measure for content-based video summarization. In *ICIP*, pages 2512 –2515, oct. 2008. 1

[8] S. Hinz. Detection and counting of cars in aerial images. In *ICIP*, pages 997–1000, 2003. 1

[9] O. Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *CVPR*, pages 696–701, 2005. 1

[10] Y. Jia, J. Wang, C. Zhang, and X.-S. Hua. Finding image exemplars using fast sparse affinity propagation. In *MM*, pages 639–642, New York, NY, USA, 2008. ACM. 1

[11] Y. Jia and C. Zhang. Front-view vehicle detection by markov chain monte carlo method. *PR*, 42:313–321, 2009. 1

[12] W. Jiang, F. Ding, and Q.-L. Xiang. An affinity propagation based method for vector quantization codebook design. In *ICPR*, pages 1 –4, dec. 2008. 1

[13] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Occlusion robust tracking utilizing spatio-temporal markov random field model. In *ICPR*, pages 1140–1144, 2000. 1

[14] N. Kanhere and S. Birchfield. Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features. *TITS*, 9:148–160, 2008. 1

[15] N. K. Kanhere, S. J. Pundlik, and S. Birchfield. Vehicle segmentation and tracking from a low-angle off-axis camera. In *CVPR*, pages 1152–1157, 2005. 1, 3

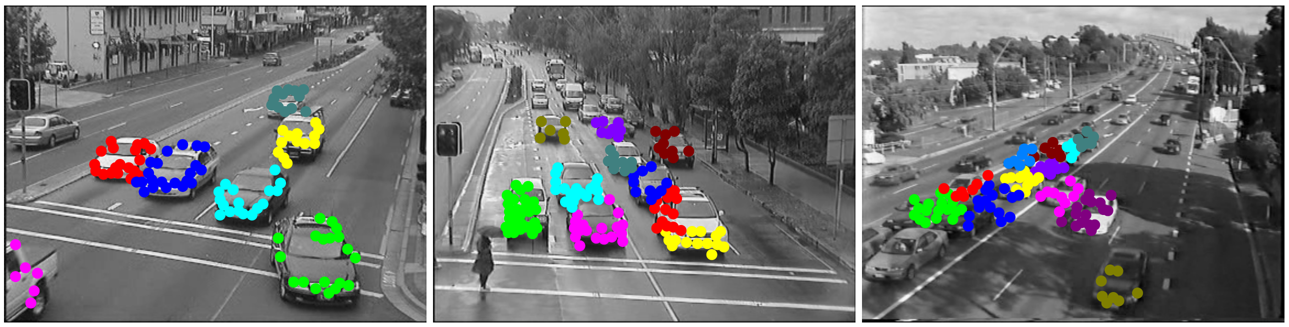

Figure 5: Example detections on the three sequences: from left to right, *city1*, *city2* and *city3*.

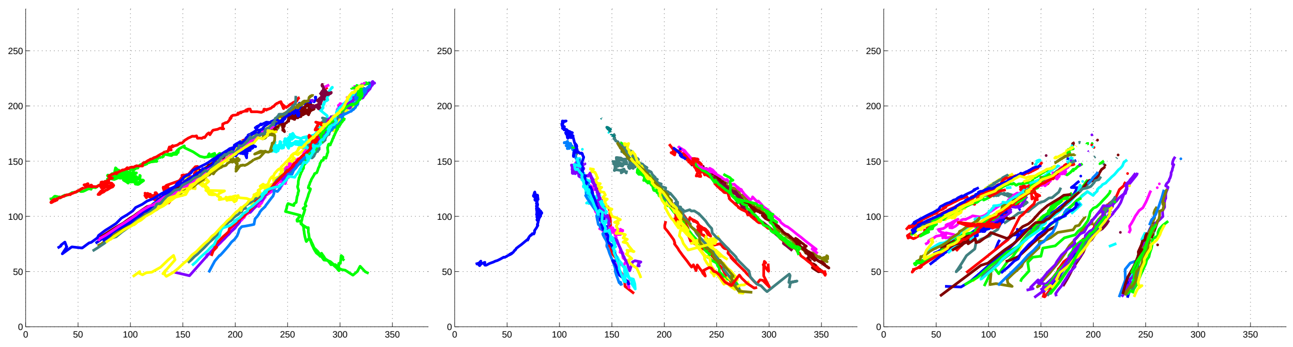

Figure 6: Example vehicle trajectories on the three sequences: from left to right, *city1*, *city2* and *city3*.

[16] Z. Kim. Real time object tracking based on dynamic feature grouping with background subtraction. In *CVPR*, 2008. 1

[17] Z. W. Kim and J. Malik. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In *ICCV*, pages 524–531, 2003. 1

[18] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *IJCV*, 10:257–281, 1993. 1

[19] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *ECCV*, pages 189–196, 1994. 1

[20] A. Levin, P. A. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *ICCV*, pages 626–633, 2003. 1

[21] Y. Li and H. Ai. Fast detection of independent motion in crowds guided by supervised learning. In *ICIP*, pages 341–344, 2007. 1

[22] S. Ma, W. Wang, Q. Huang, S. Jiang, and W. Gao. Effective scene matching with local feature representatives. In *ICPR*, pages 1 –4, dec. 2008. 1

[23] P. F. McLauchlan, D. Beymer, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *CVPR*, pages 495–501, 1997. 1

[24] V. Rabaud and S. Belongie. Counting crowded moving objects. In *CVPR*, pages 705–711, 2006. 1

[25] X. Song and R. Nevatia. A model-based vehicle segmentation method for tracking. In *ICCV*, pages 1124–1131, 2005. 1

[26] T. N. Tan, G. D. Sullivan, and K. D. Baker. Model-based localisation and recognition of road vehicles. *IJCV*, 27:5–25, 1998. 1

[27] C. Tomasi and J. Shi. Good features to track. In *CVPR*, pages 593–600, 1994. 3

[28] R. Verma and P. Wang. On detecting subtle pathology via tissue clustering of multi-parametric data using affinity propagation. In *ICCV*, pages 1 –8, oct. 2007. 1

[29] J. Xiao, J. Wang, P. Tan, and L. Quan. Joint affinity propagation for multiple view segmentation. In *ICCV*, pages 1 –7, oct. 2007. 1

[30] J. Yang, Y. Wang, G. Ye, A. Sowmya, B. Zhang, and J. Xu. Feature clustering for vehicle detection and tracking in road traffic surveillance. In *ICIP*, pages 1152–1157, 2009. 1

[31] X. Zhang and C. Furtlehner. Data streaming with affinity propagation. In *ECML*, pages 628–643, 2008. 1

[32] X. Zhang, J. Gao, P. Lu, and Y. Yan. A novel speaker clustering algorithm via supervised affinity propagation. In *ICASSP*, pages 4369 –4372, 31 2008-april 4 2008. 1

[33] T. Zhao and R. Nevatia. Car detection in low resolution aerial image. In *ICCV*, pages 710–717, 2001. 1

[34] Z.-Q. Zhao and H. Glotin. Diversifying image retrieval with affinity-propagation clustering on visual manifolds. *IEEE MultiMedia*, 16:34–43, 2009. 1