# Real Time Multi-Vehicle Tracking and Counting at Intersections from a Fisheye Camera

Wei Wang[1] Tim Gee[2] Jeff Price[2] Hairong Qi[1]
[1]University of Tennessee, Knoxville [2]Aldis Inc., Knoxville, TN
{wwang34,hqi}@utk.edu, {tim.gee,jeff.price}@aldiscorp.com

## Abstract

*This paper presents an approach for real-time multi-vehicle tracking and counting under fisheye camera based on simple feature points tracking, grouping and association. Different from traditional cameras, the main challenge under fisheye cameras is that the objects being tracked suffer from severe distortion and perspective effects in even adjacent frames. As a result, the points can be stably matched by a point tracker are much fewer, and the points even lose tracking completely quite occasionally. Firstly, to preserve points discrimination in dynamic grouping, we propose an approach based on motion similarity and neighbor-weighted grafting to transfers motion knowledge between long and short point trajectories. Moreover, to deal with cases such as points losing tracking completely or incorrect points grouping, we also propose a concept of points "identity-appearance" that integrates constrained motion for association between vehicle tracklets and segmented point groups. Our approach also overcomes several common challenges in traffic surveillance such as stopping vehicles, pedestrians and counting of linked (partially occluded) vehicles. Finally, extensive experimental results are provided on challenging fisheye image sequences to demonstrate the robustness and effectiveness of the approach.*

## 1. Introduction

Vehicle detection and tracking for traffic surveillance is an active research topic in computer vision, due to its promising potential for the reliable intelligent transportation systems. Compared to other existing technologies, vision-based systems are more attractive not only for their lower cost and easier installation, but also for their capability to capture the rich visual information for traffic analysis, such as vehicle counting, traffic flow speed estimation, statistics based signal timing and other safety applications.

To detect vehicles in a scene, the literature reports basically two groups of approaches: (i) background/foreground subtraction [5, 7, 10, 19], which first reconstructs a static background hypothesis from a sequence of images, then finds the foreground objects by calculating the difference between the background hypothesis and the current image. (ii) model-based approach, that uses an initialized *priori* [4, 12, 24]. For example, a recent model-based approach is the mixture of deformable part models [3], which was introduced in the community of object detection in images and proved to be effective to handle variations between classes, illumination and occlusion. However, the latter approach has a high computational complexity and barely meets the real time requirements in traffic applications. Therefore, most of the commercial systems are still based on the background subtraction algorithms.

When the objects to be tracked are isolated by detection, a one-to-one correspondence between objects and detected blobs is likely to appear. However, this assumption does not hold when multiple objects occlude one another, resulting in a foreground blob contains multiple objects. To solve the common challenges in tracking, many approaches have been proposed that can be categorized into three main categories: feature-based tracking [2, 10], region-based tracking [13,19] and contour-based tracking [22]. These tracking approaches have been demonstrated to be effective to solve some specific problems in certain scenarios, but a general solution is still a research target to achieve.

In this paper, we focus on the problem of real-time vehicle tracking and counting in *streaming* hemispherical image sequences. As a wide view vision sensor, the fisheye lens camera is becoming increasingly used in networked robotics [8] and human activity monitoring systems [14]. Although they have great advantage of reducing the number of required cameras to cover large space for traffic surveillance (*e.g.* at an intersection), the acquired images suffer from severe distortion and perspective effects, demanding non-standard algorithms for vehicle tracking. In order to cope with the issue of multi-vehicle tracking and counting, we propose an approach that integrates low level feature-point based tracking and higher level "identity appearance" and motion based real-time association, aiming at tackling

978-1-4799-6683-7/15 $31.00 © 2015 IEEE
DOI 10.1109/WACV.2015.10


Figure 1. Exemplar challenge of points tracking in fisheye image. Left: vehicle appearance rotated in adjacent frames. Right: only a few points are matched in tracking: the points ('o') in upper are the ones wait to be matched; the points in lower ('+') are the ones matched in tracking; while the blue points ('.') are newly detected.

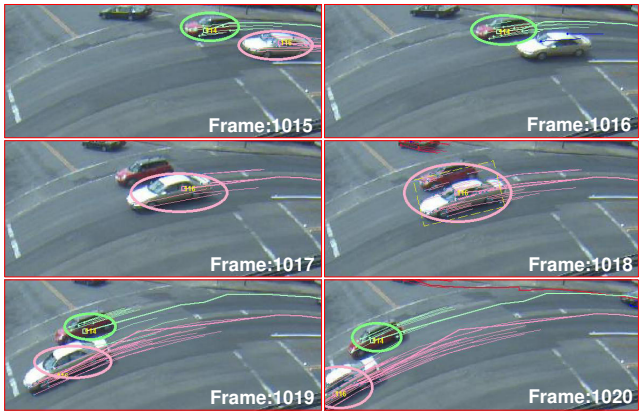


Figure 2. Exemplar challenge of association between Tracklets (id:114 (green), id:116 (pink)) and point groups, *e.g.* each vehicle lost tracking in the 2nd and 3rd frames, while points from 2 vehicles incorrectly grouped together in the 4th frame.

several major tracking problems under fisheye camera.

The feature-based tracking is realized by detecting and tracking individual corner points using the Kanade-Lucas-Tomasi (KLT) tracker [17], and then grouping the points based on their featured trajectories. However, the challenge in applying this point tracker under fisheye camera is that it is generally not easy to accurately track a corner point over even a short period of time. As shown in Figure 1, because of the distortion in fisheye camera, the flat ground becomes hemispherical and the vehicle between frames appears quite rotated. Therefore, much fewer points can be matched between adjacent frames than that in traditional cameras, *e.g.* the number of matched points ('+') only counts for a small portion of the points ('o') that need to be tracked. As a result, most of the point trajectories are quite short with limited historical information, which will not be discriminative enough to achieve good performance in grouping. To preserve more motion knowledge for points trajectories, we propose to use **grafting** between point trajectories for motion knowledge transfer and preservation, such that we can still have plenty of historical motion knowledge for feature grouping within fisheye camera. In addition, to guarantee the quality of tracked points, the bidirectional matching [6] and a smoothness constraint are also incorporated in generation of the point trajectories.

The goal of multi-vehicle tracking is to recover trajectories of all vehicles while maintaining identity labels consistent, such that the counting can be done more precisely. Although the point trajectory grafting helps to enhance the discrimination between feature points, incorrect grouping still happens because of the illumination change and vehicle appearance variation under fisheye camera. Figure 2 shows two vehicles being tracked in 6 frames. In the 2nd and 3rd frames, the points on each car lose tracking, respectively; while in the 4th frame, the points on the two vehicles are grouped together. To meet the requirement of real time processing, no complicated features are used other than the corner points. Therefore, either the case that all the points on a vehicle lose tracking or the case that points from two vehicles are grouped together, the resultant effect will be similar to the scenario that one vehicle disappears, but the identity of each vehicle still needs to be correctly re-found when points from the two vehicles be correctly separated again (even after a long time disappearance). To solve this challenge of possible identity losing or switching, we propose a new concept "**identity appearance**" combined with constrained **motion** for real time association between *Tracklet* at vehicle level and *point group* at point level. Each point will memorize its identity (i.e., which tracklet it belongs to) at each frame since its beginning, and the "identity appearance" of a segmented point group is represented as the histogram of points historical identities. Based on the affinity model derived from both identity-appearance and motion, the Hungarian algorithm [1, 11, 21] is used to find the global optimum and associate tracklets and detections robustly.

In addition, it is also difficult to tackle problems caused by stopped vehicles or interrupted pedestrians, *e.g.*, the stopped vehicles at red light will be eventually treated as background, and the slow moving vehicles due to traffic will be more difficult to be separated. The interrupted pedestrians also might be falsely detected and counted as a smaller vehicle. We propose to use movement to prevent these possible false results. Any point with movement less than a threshold will not be considered in grouping and not be updated in association, although their point trajectories are still kept and updated at the point level. Further more, for those vehicles appear side-by-side and travel through the camera view in parallel, the feature-based tracking has limited capability to separate them and count as multiple vehicles. We propose to use size and shape analysis of point

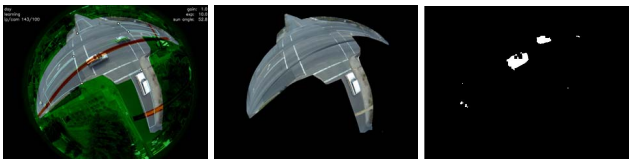

Figure 3. Left: raw image frame; Middle: estimated background; Right: segmented foreground blobs.

cloud to discriminate whether the big size point group corresponds to a single big track or multiple sedans.

In summary, our work presents an integrated approach that combines feature-based tracking and model-based association to solve challenges in fisheye camera for multi-vehicle tracking and counting at traffic intersections. The contributions are that:

- A grafting approach that transfers motion knowledge between point trajectories under fisheye camera, preserving feature points discrimination in grouping.
- A new concept of "identity appearance" that combines motion for robust and efficient association between vehicle Tracklets and detection of feature point groups.
- Simple criteria to relieve negative effects in grouping and counting caused by stopping vehicles, pedestrians and linked (partially occluded) vehicles.

## 2. Feature Point Tracking and Grouping

This section introduces how the point trajectories generated and how to dynamically group the feature points based on their motion knowledge in trajectories. A typical corner point trajectory ($T_p$) records motion information at every frame $t$, including: point position ($P_{pos}$) in image space and location ($P_{loc}$) in world coordination space, point velocity ($P_{vel}$) in world space, point blob membership ($P_{bm}$), trajectory age ($P_{age}$), point identity ($P_{id}$) indicating which vehicle the point belongs to, and grafted trajectory ($P_{gft}$).

### 2.1. Background Subtraction

The background subtraction always requires a relatively small computation time. We use a typical background subtraction algorithm that applies Kalman filter to the pixel intensities to estimate the background efficiently [9, 10].

An example result is shown in Figure 3, we use the green mask to cover the unrelated part of background and only reconstruct the road zone to further reduce computation. The middle sub-figure shows the reconstructed background, where a stopped vehicle is recognized as background. However, we will use a movement based criterion to remove the negative effect brought by stopped vehicles in feature point grouping. With the estimated background, the foreground of two moving vehicles are segmented out in the right sub-figure. Then, the feature point detection and tracking will only be applied on the foreground zone, in order to reflect the motion of vehicles only.

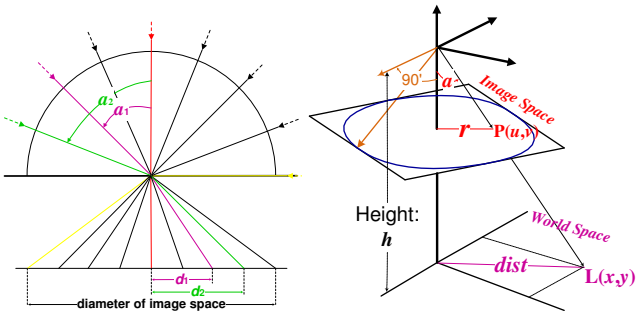


Figure 4. Left: Projection model of fisheye camera; Right: Geometrical relationship between point in image space and its location in world space.

### 2.2. Fisheye Calibration

In order to perform a fair evaluate on the motion of feature points, it is important to calibrate their image positions into world coordination space since the same distance at the edge of a fisheye image is much larger than that at the central part of the image. Moreover, the vehicle's velocity also needs to be described by an uniform unit (foot) in the world space. As shown in the left of Figure 4, the calibration is based on the property of fisheye projection that the distance $d$ between an image point and the principal point is linearly dependent on the angle of incidence $a$ of the ray from the corresponding object point. Therefore, the value of the angle per pixel $\varrho = \frac{a}{d}$ is a constant and $\frac{a_1}{d_1} = \frac{a_2}{d_2}$. Then, given a point $P(u, v)$ in image space, as in right of Figure 4, the point location in the world space $L(x, y)$ is calculated as:

$$x = h\tan(u \cdot \varrho), \quad y = h\tan(v \cdot \varrho) \tag{1}$$

where $h$ is the mounting height of the camera. Similarly, any point in the world space can be easily calibrated back into the image space.

### 2.3. Point Tracking and Grafting

As introduced in Sec.1, we use KLT tracker for feature point tracking. Similar to some previous work [10], we also use the proximity and motion history of point trajectories for dynamic grouping. However, different from previous works is the challenges brought by the fisheye camera. Objects in fisheye images suffer from severe distortion even in adjacent frames. Therefore, it is much more difficult to correctly and stably track a corner point over frames.

First of all, the quality of the point tracking is key for the trajectories grouping. There will be too few tracked points to describe the motion of a vehicle if we apply a high threshold value for KLT matching, while too many false tracking will generate if we apply a lower threshold. Therefore, we

use the forward-backward error checking [6] to find enough number of reliably tracked points while also remove tracking failures. Moreover, a smoothness constraint is also applied to point trajectories to further remove tracking failure, *e.g.* a point is matched to a patch on the ground result from illumination change. The smoothness is calculated with the point velocity $P_{vel}$ as below, where $P_{vel_{(t)}}$ is a vector.

$$VAD(t) = \cos^{-1}\left(\frac{(P_{vel_{(t)}})^T(P_{vel_{(t-1)}})}{\|P_{vel_{(t)}}\|\|P_{vel_{(t-1)}}\|}\right) \leq 75^\circ \quad (2)$$

where VAD measures the vector angle divergence between adjacent moving directions. Any point trajectory with the new VAD($t$) not meeting this constraint will be removed.

The unique property of fisheye images also create the challenge that many feature points cannot be stably tracked over frames, shown in Figure 1, therefore a great portion of the point trajectories are of short temporal length, resulting in historical motion knowledge outflow and discrimination weakening. We assume that the points on the same vehicle will have almost identical motion at any time $t$. The historical motion can be better kept if we clone a long point trajectory $Tp_i$ to others of short length before the point $Tp_i$ loses tracking (disappear). To guarantee the two points in grafting clone are from the same vehicle, we also expect that they are spatially close to each other and enforce they are from the same foreground blob. Therefore, also explained in Figure 5, the point selected for grafting clone based on motion similarity and spatial distance to another $Tp_j$ is found as:

$$\begin{aligned} &\underset{i}{\operatorname{argmin}}\,(vel_{diff}(i,j)\cdot exp\,(loc_{diff}(i,j)/\delta)) \\ &vel_{diff}(i,j) = \sum \|P_{vel_{(t,i)}} - P_{vel_{(t,j)}}\|_2/T \\ &loc_{diff}(i,j) = \sum \|P_{loc_{(t,i)}} - P_{loc_{(t,j)}}\|_2/T \\ &T = \min(P_{age_{(i)}}, P_{age_{(j)}}), \quad s.t., \quad P_{bm_i} = P_{bm_j} \end{aligned} \quad (3)$$

Then, the motion history and trajectory discrimination can be well preserved with grafting unless all the feature points lose tracking completely in certain frame.

## 2.4. Point Dynamic Grouping

The next step is to cluster the point trajectories into different groups on each frame (dynamic grouping). Each segmented group will be represented by a unique color in our results. The grouping algorithm we used is the Normalized-Cut [15, 16] – each feature point is represented as a node on a graph $G = (V, E)$ and the edge weights $w(\mu, \nu)$ between nodes $\mu$ and $\nu$ are set as the distance between two featured trajectories. Then, the graph cut solution measures both the total dissimilarity between the different groups as well as the total similarity within groups to achieve robust segmentation. We use five metrics to measure the distance between any two point trajectories, i.e., the spatial distance $\varpi_c$ in

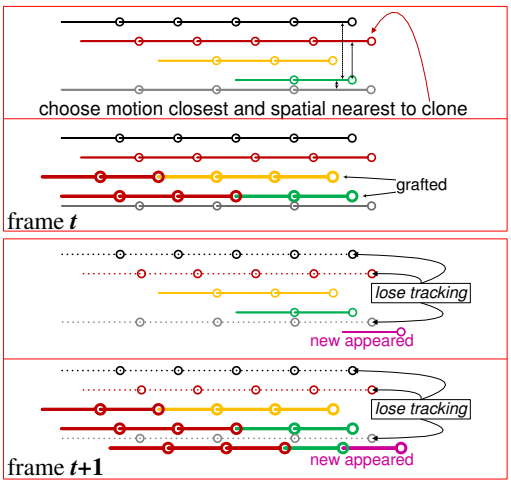


Figure 5. Illustration for grafting. The upper and lower are without and with grafting clone in frames $t$ and $t+1$, respectively. Notice that all the points must in the same foreground blob.

current frame $t$, the largest historical distance $\varpi_h$, and the averaged velocity divergence $\varpi_v$ in world space; the recent movement difference $\varkappa_m$ at frame $t$ and the historical blob membership divergence $\varkappa_b$ in image space. The weighted measure of pairwise distance $w(\mu, \nu)$ is calculated as:

$$exp(\frac{-\varpi_c}{\eta_c})\cdot exp(\frac{-\varpi_h}{\eta_h})\cdot exp(\frac{-\varpi_v}{\eta_v})\cdot exp(\frac{-\varkappa_m}{\eta_m})\cdot exp(\frac{-\varkappa_b}{\eta_b}) \quad (4)$$

where the $\eta$s are experientially predefined.

When a new vehicle appears, a *Tracklet* ($T_g$) at vehicle level will be generated with several descriptions including: group position ($G_{pos}$) in image space and location ($G_{loc}$) in world space, group velocity ($G_{vel}$) in world space, group age ($G_{age}$), and an unique group ID ($G_{id}$), *et al*. The group position is determined from its current "member" feature points, and the identity of the points in this group will also be updated as $P_{(id,t)} = G_{id}$. For those groups that contain tracked points with only one $P_{id} = k$ from the beginning of the $T_p$, it is quite easy to link the point group to a Tracklet $T_g(k)$. Otherwise, we will need to make use of the "identity-appearance" and motion based model to associate a point group to some existing Tracklet $T_g$, since the error cases such as completely losing tracking (no point matched in KLT tracking) or error segmentation always occurs.

In point dynamic grouping, it is important to eliminate any negative effect brought by stopping vehicles. The points on stopping vehicles are tend to be mis-segmented together, resulting in false update of the point ID. We propose to use the *movement* of point as a simple but effective criterion that any point with movement at $t$ less than a threshold $\beta$ (2 feet) will not be considered in grouping at $t$, though their point trajectories are still there with just $P_{id}$ not updated. Then, when those vehicles restart to move, they can be re-

detected and easily re-linked to a right tracklet, unless the vehicles stopped for a long time that be mis-recognized as background and lose tracking of all the points. Moreover, this criterion also helps to exclude pedestrians. Though the pedestrians can be detected, they cannot form any Tracklet (resulting from grouping) to be tracked and counted because of their obvious slower moving speed.

## 3. Affinity based Association

Because of the challenges in fisheye camera, it is also not easy to link the segmented point group to some Tracklet purely based on the points Id, $P_{id}$. For example, the points on a vehicle will be all newly detected with $P_{id}$ being null if the vehicle completely loses tracking in previous frames but re-detected again (similar to fully occluded). Moreover, the grouping algorithm also works not so reliably under fisheye camera as that in traditional cameras. Then, the point trajectories might contain multiple values in their historical recordings of $P_{id}$ if the points from different vehicles are grouped together in several previous frames but separated again (merge and split, similar to the situation of partial occlusion). Therefore, instead of generating a new tracklet $T_g$ for the re-appeared group, we need to re-link the point group to its corresponding Tracklet ended several frames before, such that the vehicles label $G_{id}$ can be preserved with consistency and the counting can be done more precisely.

### 3.1. Identity-Appearance

To make the detection and tracking more efficient for real time processing, instead of using computationally intensive detection algorithms to find vehicles, we adopt background subtraction and feature points based tracking. Therefore, the only "*appearance*" we can make use of is the identities of the points scattered on a vehicle. In our approach, the identity of individual point is no longer a single value but a list of historical recordings. As shown in Figure 6 (b), a histogram $H_c$ can be derived from the lists of $P_{id}$ in point group $c$, with $h_c(k)$ indicating how possible the point group $c$ belongs to Tracklet $k$ in association. Benefit by this "identity-appearance", point groups will be linked to correct tracklets after "merge and split", and the possible identity switch or false new tracklet generation can be reduced.

### 3.2. Constrained Motion

Motion is quite useful in targets tracking and has been already explored in many previous works [18, 20, 21]. As shown in Figure 6 (c), the motion affinity is estimated by 3 metrics, including: distance $\Delta_d$ between the predicted Tracklet tail location $G_{preloc,k}$ and the detected point group $\overline{G}$, vector angle $\Delta_m$ between the directions of predicted Tracklet moving and point group moving, and vector angle $\Delta_s$ between the most 2 recent $(t, t-1)$ movements after link for tracklet smoothness. The Tracklet tail location at

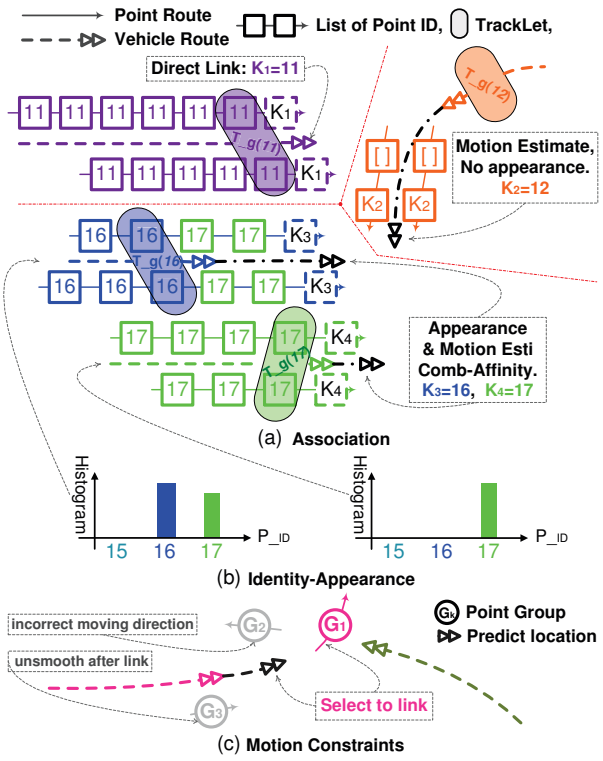


Figure 6. (a): upper left, association using points ID only; other, association using appearance and motion combined affinity model. (b): illustration of the identity-appearance. (c): illustration of motion constraints (distance, moving direction, smoothness).

$t$ is predicted by a momentum-based method. A small $\Delta_m$ guarantees the consistent moving direction while a small $\Delta_s$ guarantees a Tracklet is not linked to a group $\overline{G}$ that behind the Tracklet tail location of $(t-1)$.

Finally, the association based on affinity of both identity-appearance and constrained motion is calculated as:

$$P_{link}(T_g(k), \overline{G}(c)) = exp(\frac{-h_c(k)}{\gamma_h}) \cdot exp(\frac{-\gamma_d}{\Delta_d}) \cdot exp(\frac{-\gamma_m}{\Delta_m}) \cdot exp(\frac{-\gamma_s}{\Delta_s}) \quad (5)$$

where $P_{link}(T_g(k), \overline{G}(c))$ indicates the possibility to associate $T_g(k)$ and $\overline{G}(c)$. A Hungarian algorithm is used to find the global optimum. Any $\overline{G}$ that is not linked to a $T_g$ because of an over threshold $(\xi)$ link cost will be determined as a beginning of a new Tracklet.

Notice that the association is done in two steps for efficiency purpose. Firstly, for those groups that only a single unique point identity $P_{id} = k$ is contained in their "identity-appearance" histograms $H$, *e.g.* the right one of Figure 6 (b), they will be linked to Tracklet $k$ directly. Then, the association of the remaining groups and Tracklets will rely on the affinity based model as Eq. 5.

Figure 7. Left: multi-vehicle in oversized blob; right: Single large vehicle in oversized blob; Middle: blob featured shape.

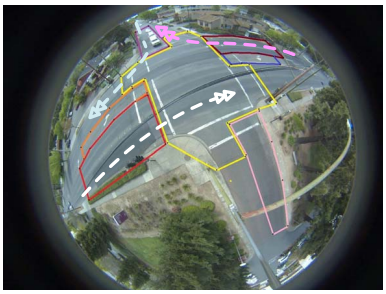

Figure 8. Vehicle counting based on valid Tracklets.

## 4. Vehicle Counting Scheme

Vehicle counting statistics is a real application in traffic surveillance based on tracking. Generally, the better the tracking, the more accurate the counting. However, if multi-vehicle appear together and stay as partially occluded all the time when they pass through the camera view, as shown in left of Figure 7, they will always be in the same foreground blob and generate very similar motion. Therefore, it will be very difficult to segment them and track them individually. Although the width of the point group provides a cue to discriminate single or multi-vehicle, the cue is confused by large vehicles with extraordinary height, *e.g.* big truck, large bus, which will also appear over-wide when projected onto the lane plane. Instead of using complicated detection approach to discriminate single or multi-vehicle for precise counting, a simple scheme is proposed for the counting purpose. The width of each point group $\overline{G}$ will be estimated in the world space when it starts to be counted. Then, blob shape analysis is applied on any suspicious point group of over-width. Based on the observation as shown in Figure 7 that the blob shape of multi-vehicle appears as overlapped blobs while single large vehicle appears as a huge chunk, we calculate the ratio $\Gamma$ of foreground area over the size of a fitted rectangle along the moving direction. Then, any group $\overline{G}$ of over-width with $\Gamma$ smaller than a threshold for more than 2 frames will be counted as multiple vehicles.

The vehicle counting mainly based on valid Tracklet $T_g$. As shown in Figure 8, six counting zones and one central zone are deployed in the fisheye camera view. Any Tracklet passing through the counting zone and entering into the central zone will be treated as valid and be counted.

| Method | GT | RC | PS | VT | IDE |
|---|---|---|---|---|---|
| BaseLine1-DS1 | 194 | 91.6% | 89.3% | 90.3% | — |
| BaseLine2-DS1 | 194 | 98.5% | 90.0% | 100% | 7.7% |
| Ours-DS1 | 194 | 98.5% | 97.9% | 38.1% | 0% |
| Ours-DS2 | 212 | 98.6% | 96.8% | 34.4% | 1.4% |
| Ours-DS3 | 177 | 100% | 96.2% | 38.4% | 1.1% |

Table 1. Quantitative evaluation of tracking performance.

## 5. Experimental Results

We present the tracking and counting results of our approach on three intersection fisheye videos (DS-1,2,3) collected at different times. The tracking performance is evaluated based on five metrics: Ground Truth (GT), the number of vehicles; Recall (RC), the correctly detected and tracked vehicles divided by the total number of vehicles; Precision (PS), the correctly detected and tracked vehicles divided by the total number of counted Tracklets; Virtual Tracklets (VT), the appeared invalid Tracklets (not counted) divided by the total number of vehicles; and identity exchange between different vehicles (IDE). The tracking purpose is for vehicle counting, which is also evaluated by four metrics: counted number (CT), accuracy ratio (AC), falsely counted number (FA), and miss count number (MS).

The tracking performance is compared to the BasedLine-1 that without either the grafting clone or the affinity-based association, while the BaseLine-2 is just without affinity association. Then, the counting performance is compared to an approach (3dM) that already embedded in a traffic monitoring product of Aldis Inc. The 3dM approach formulates the tracking as searching for the multi-vehicle configuration that maximizes the posterior probability of features when project a 3D vehicle model to a 2D foreground zone.

In our implementation, many parameters need to be predefined. Fortunately, all the parameters are not sensitive that can be roughly fixed by experience, *e.g.*, $\delta$=15 in Eq.3; $\eta_c$=$\eta_h$=200, $\eta_v$=1.0, $\eta_m$=10, $\eta_b$=0.1 in Eq. 4; $\gamma_h$=0.1, $\gamma_d$=6, $\gamma_m$=60, $\gamma_s$=90 in Eq. 5. For the cost matrix $P_{link}$, instead of using the matrix $(n \times n)$ directly, we use the augmented matrix $(2n \times 2n)$ in [23] as input of Hungarian that enables us to set a threshold $\xi$ to find initial and terminating tracklets. The processing is frame-by-frame in an on-line manner of counting. The average processing time for each frame is about 750 ms on a laptop with a 2.7G CPU and 3G memory in Matlab environment. Much shorter processing time can be expected if the code can be optimized and implemented with C++ and OpenCV for industrial work.

The quantitative performance of tracking is shown in Table 1. We can find that the point trajectory grafting increases discrimination between points and improves the recall ratio, while the affinity-based association reduces false alarms and increases the precision. The virtual Tracklets (VT) ap-

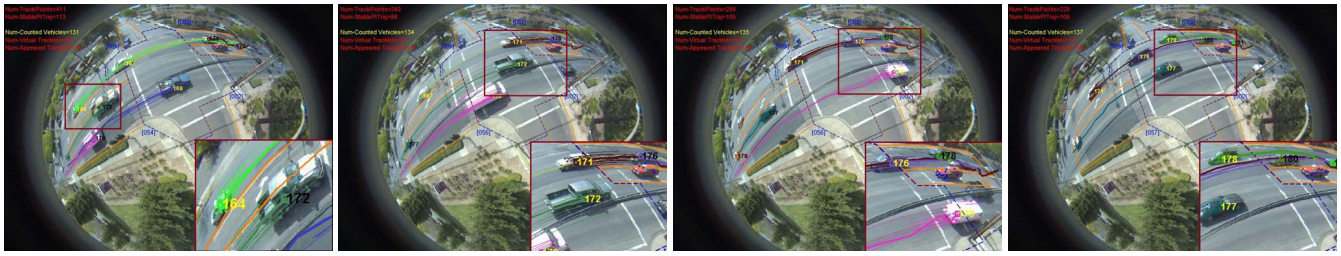

Figure 9. Frames: 1409,1414,1419,1424. Illustration of counting scheme: vehicles passed counting zone are shown in yellow ID, otherwise in black. Also, the stationary vehicle at the lower right lane never affects the grouping of other moving vehicles.

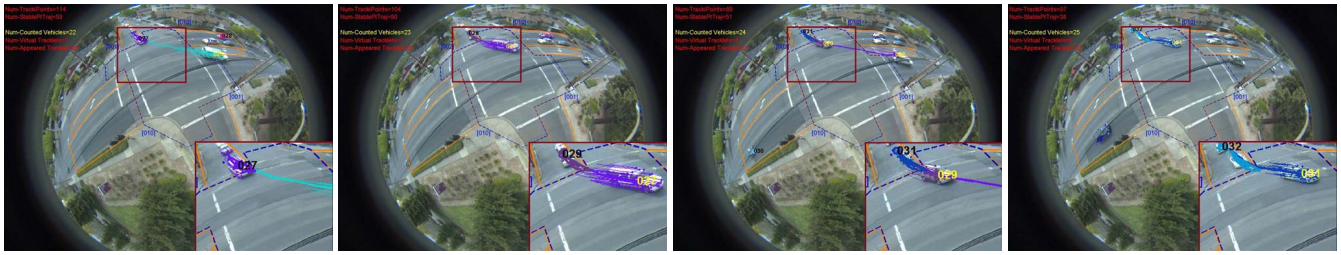

Figure 10. Frames: 243,253,263,278. Successfully tracking and counting of vehicles emerging from successive group split.

| Method | GT | CT | AC | T-B-L-R | FA | MS |
|---|---|---|---|---|---|---|
| 3dM-DS1 | 194 | 170 | 87.6% | | | |
| Ours-DS1 | 194 | 195 | **99.5**% | 12-2-83-98 | 4 | 3 |
| 3dM-DS2 | 212 | 192 | 90.6% | | | |
| Ours-DS2 | 212 | 216 | **98.1**% | 6-1-97-112 | 7 | 3 |
| 3dM-DS3 | 177 | 165 | 93.2% | | | |
| Ours-DS3 | 177 | 184 | **96.1**% | 14-3-76-91 | 7 | 0 |

Table 2. Quantitative evaluation of counting performance.


Figure 13. Frames:1545,1550. The pedestrian is detected but not tracked because of slower movement than vehicles.

pears when a Tracklet does not pass any counting zone, so it is also known as the number of times a complete Tracklet being interrupted. VT happens when a vehicle loses tracking over too many frames due to illumination reasons or a traffic signal, then a new Tracklet will be generated to represent motion of the vehicle. Although we desire a complete vehicle Tracklet instead of several fragments or VTs, in fact it does not affect counting performance because there will be only one valid Tracklet passing through the counting zone for each vehicle. Also, preventing the exchanging of vehicles ID among different vehicles is important for us to retrieve any vehicle's trajectory for other statistics. Fortunately, the schemes in our approach help reduce both the VT and IDE. The performance of counting is shown in Table 2. From the comparison on data set (DS) 1,2,3, we observe that our approach improves the counting accuracy about 7.5% on average. In addition, based on the valid Tracklets, we also provide the counting number of vehicles from different directions. The T-B-L-R in Table 2 represents vehicles from top, bottom, left and right, respectively. Sample tracking and counting results are shown in Figure 9 to Figure 13.

## 6. Conclusion

This work presented an approach for real time multi-vehicle tracking and counting under fisheye camera that integrates the low level feature point tracking and the higher level affinity based association. The points motion knowledge can be transferred between points via grafting clone, such that the discrimination of points is well preserved and the performance of dynamic grouping is improved. In association, the "identity-appearance" was proposed to describe the historical membership of point groups, and combined with the constrained motion to select the correct vehicle tracklets in merge and split. Extensive experiments showed that the proposed approach provided a promising way to achieve good performance for vehicles tracking and counting based on traditional and simple feature points tracking and grouping under the challenging fisheye camera.

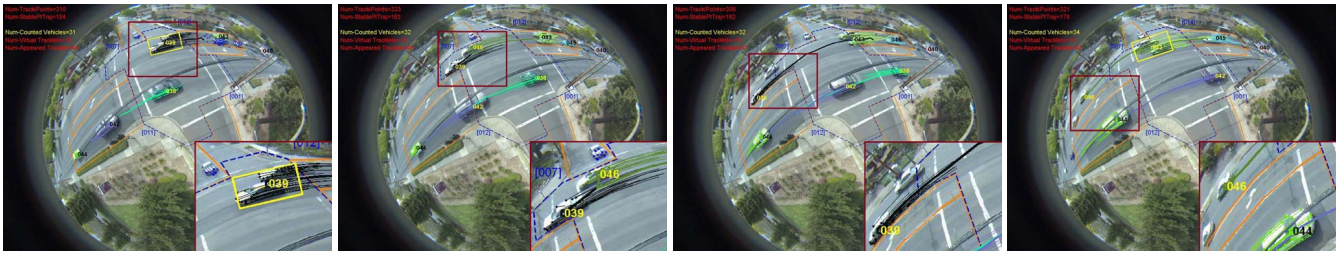

Figure 11. Frames: 377,382,387,392. Successfully tracking of vehicle (**046**) after completely lose points tracking over multi-frames.

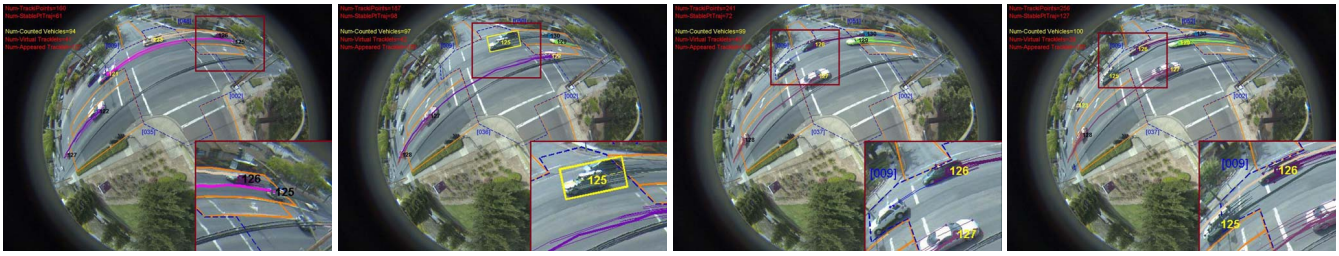

Figure 12. Frames: 1042,1049,1052,1053. Successfully tracking of vehicles (**125,126**) after incorrect merge and split over multi-frames.

## References


[1] X. Chen, Z. Qin, L. An, and B. Bhanu. An online learned elementary grouping model for multi-target tracking. In *CVPR*, 2014. 2

[2] A. Dore, A. beoldo, and C. Regazzoni. Multitarget tracking with a corner based partical filter. In *ICCV Workshops*, 2009. 1

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 1

[4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003. 1

[5] J. Jodoin, G. Bilodeau, and N. Saunier. Urban tracker: Multiple object tracking in urban mixed traffic. In *WMVC*, 2014. 1

[6] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *ICPR*, 2010. 2, 4

[7] N. Kanhere, S. Pundlik, and S. Birchfield. Vehicle segmentation and tracking from a low angle off-axis camera. In *CVPR*, 2006. 1

[8] K. Kemmotsu, T. Tomonaka, S. Shiotani, Y. Koketsu, and M. Iehara. Recognizing human behaviors with vision sensors in a network robot system. In *ICRA*, 2007. 1

[9] M. Kilger. A shadow handler in a video based real time traffic monitoring system. In *WACV*, 1992. 3

[10] Z. Kim. Real time object tracking based on dynamic feature grouping with background subtraction. In *CVPR*, 2008. 1, 3

[11] C. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, 2011. 2

[12] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *Int. Journal of Computer Vision*, 77(1-3):259–289, 2008. 1

[13] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *CVPR*, 2005. 1

[14] M. Saito, K. Kitaguchi, G. Kimura, and M. Hashimoto. People detection and tracking from fisheye image via probabilistic appearance model. In *SICE Annual Conference*, 2011. 1

[15] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, 1998. 4

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, 3(2):119–133, 2000. 4

[17] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994. 2

[18] B. Song, T. Jeng, E. Staudt, and A. Roy. A stochastic graph evolution framework for robust multi-target tracking. In *ECCV*, 2010. 5

[19] X. Song and R. Nevatia. Detection and tracking of moving vehicles in crowded scenes. In *WMVC*, 2007. 1

[20] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *CVPR*, 2009. 5

[21] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *CVPR*, 2012. 2, 5

[22] M. Yokoyama and T. Poggio. A countour based moving object detection and tracking. In *PETS*, 2005. 1

[23] Q. Zen and C. Shelton. Improving multi-target tracking via social grouping. In *CVPR*, 2012. 6

[24] W. Zheng and L. Liang. Fast car detection using image strip features. In *CVPR Workshops*, 2009. 1