An Efficient System for Vehicle Tracking in Multi-Camera Networks

Michael Dixon, Nathan Jacobs, Robert Pless
Washington University
St. Louis, MO, USA
(msd2|jacobsn|pless)@cse.wustl.edu

***Abstract*—The recent deployment of very large-scale camera networks has led to a unique version of the tracking problem whose goal is to detect and track every vehicle within a large urban area. To address this problem we exploit constraints inherent in urban environments (i.e. while there are often many vehicles, they follow relatively consistent paths) to create novel visual processing tools that are highly efficient in detecting cars in a fixed scene and at connecting these detections into partial tracks. We derive extensions to a network flow based probabilistic data association model to connect these tracks between cameras. Our real time system is evaluated on a large set of ground-truthed traffic videos collected by a network of seven cameras in a dense urban scene.**

I. Introduction

Tracking vehicles as they pass through a city is a classic surveillance problem. An extreme version of this problem is to track every vehicle in an urban area. Urban areas are interesting problem domains; they are complex because they often have high vehicle density and frequent occlusion, but also simple because vehicles follow very predictable paths. In this paper we develop a probabilistic tracking system that exploits this predictability to perform real time multi-camera, multi-vehicle tracking. In particular, we focus on the development of image processing tools that are fast enough to support real time tracking within each camera, and a probabilistic model that supports combining partial tracks within and between cameras.

This work is inspired by recent work [6] that exploits the urban environment to track within a camera very efficiently. Their insight is that since cars usually travel within a lane, it is possible to decompose the 2D tracking problem to a set of 1D tracking problems; creating tracklets that capture vehicle motion within one lane, and creating complete tracks (including, for example lane changes) by connecting appropriate tracklets. However, the previous work uses an ad hoc scoring mechanism for connecting tracklets and is limited to single camera.

The current work offers three major contributions. First, we cast the lane-based tracking framework within a strict probabilistic framework, mapping image coordinates to geo-referenced coordinates so that we can express priors in globally consistent units, rather than ad hoc units of pixels per frame. Second, we offer a novel measure for the likelihood that two tracklets come from the same vehicle. Finally, we offer quantitative results on a publicly available dataset of hundreds of hand-tracked cars traveling through a series of (slightly overlapping) camera views. Figure 1 highlights one of the 622 objects tracked by our multi-camera tracking algorithm, showing the individual tracklets linked together to make a complete track.

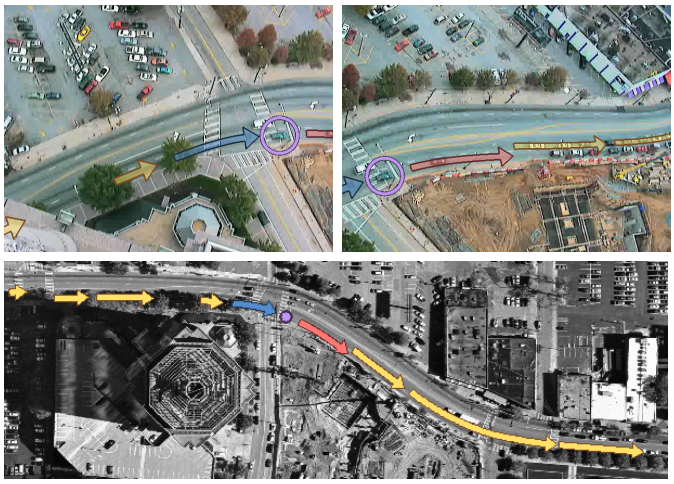

Fig. 1. Our tracking framework creates short high-confidence tracklets of cars while they remain in one lane and links them with an efficient MAP optimization using min-cost network flow. The top shows the tracklets for one vehicle in two of the cameras; the bottom shows the tracklets from all seven cameras that were associated with that vehicle.

These results highlight the benefits of tracking within a collaborative camera network — errors in each camera are fewer within the system than they are using the same algorithm within each camera. We believe this framework offers a scalable approach to city-scale, real time tracking of vehicles within a city. This type of tracking could support optimization of traffic light patterns, grounds models of typical traffic behavior that is useful for civic planning, and supports data mining approaches for unusual behavior detection.

A. Related Work

Using Space-Time Sheets: Reasoning about full spatio-temporal volumes has become extremely popular with the increasing memory capacity and speed of computers. Explicit reasoning about spatio-temporal slices has received only limited attention. Spatio-temporal slices were first used in the context of regularizing structure from motion when the camera motion stays constant [3], and more recently to reason about specular reflections [4]. Within the context of human activity recognition, spatio-temporal slices have been shown to be an

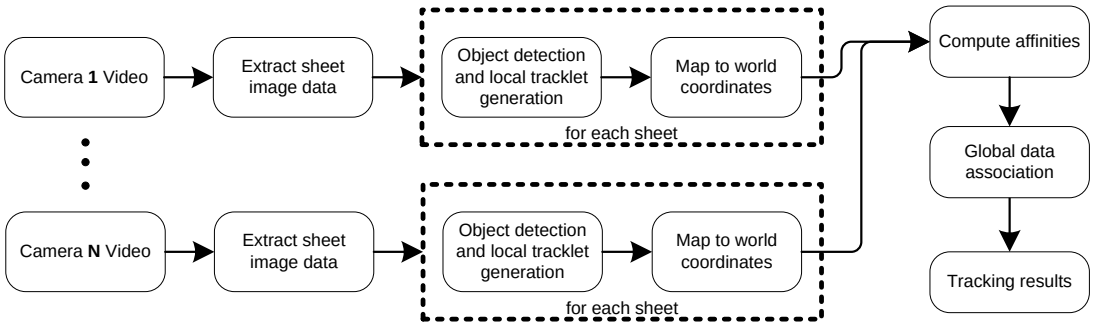

Fig. 2. A block diagram of the data flow within our tracking system. Our system extracts a set of space-time sheets from each video (separately) by sampling the image data along each lane, processes the sheets to generate a set of partial object tracks in world coordinates, then solves a global MAP data association problem to create complete object tracks.

effective and compact representation for applications such as person detection and people counting [17], tracking [14], [18], and behavior recognition [10]. Spatio-temporal slices have also been used in traffic flow measurement sensors to measure vehicle velocity [**?**].

Multi-Object Tracking: There is a large body of work on visual object tracking; see the article by Yilmaz [20] for an overview. Most recent work on multi-object tracking uses data-association based methods [21], [15], [2], [8], [11]. These methods use a wide variety of optimization techniques to build object tracks from lower-level primitives. A common element of all these approaches is the use of a pair-wise score function to help determine which primitives to associate. These scores are typically based on a combination of kinematic, appearance, and other, often ad hoc, terms. In this work we describe an association score that is suitable for linking overlapping and non-overlapping tracklets.

Tracking with Multiple Cameras: Using multiple-camera views for object tracking can significantly reduce ambiguities present in the single-view case. A broad range of camera deployment scenarios including overlapping [5], partially overlapping [9], and non-overlapping [7] fields of view have been considered. Our system uses a standard approach of mapping objects to the ground plane and performing data association in world coordinates.

B. System Overview

Our proposed system (shown in Fig. 2) for tracking multiple objects in multiple camera views works in three stages. First, given the location of roads in each camera view, we extract a set of space-time sheets from each video by sampling the image data along these curves (see Section II). Next, we process the sheets to generate a set of partial object tracks, or tracklets (see Section III). Finally, we perform data association, in world coordinates, to combine tracklets from all sheets into complete object tracks (see Section IV).

II. Space-Time Sheets

A space-time sheet, or simply a sheet, is an image constructed from video data extracted along a single curve in image space over time. In this work, the curve, which we call a sheet curve, c, is typically drawn along a lane of traffic. Pixels in a $sheet$ are parametrized by a temporal coordinate t and a spatial coordinate p, where $c(p)$ is a position along the sheet curve in image space. To construct a space-time sheet for a sheet curve c and a video I, we extract the colors of pixels under the curve, sampling every one pixel width, for each frame t of the video such that $sheet(p,t) = I(c(p),t)$. Fig. 3 shows example sheets extracted from video of a traffic scene highlighting that relevant video data for tracking over hundreds of frames can be summarized within a single image. Thus tracking a car while it travels within a lane reduces to a simpler 1D tracking problem.

In a single camera view there may be one or many space-time sheets of interest depending on the number of lanes in view. In this work, we manually specify the location of sheet curves in image space. While it is possible to automatically find sheets [12], [19], [6] we believe that the work of finding sheets is largely orthogonal to the work of understanding and using the sheets. Additionally, we assume that we have a mapping of point locations $c(p)$ along each sheet into a world coordinate system, consistent across all cameras. In our case this mapping is computed by estimating a homography between each camera and an aerial site image using manually clicked corresponding points. We acknowledge that this registration problem is challenging but also consider it to be orthogonal work in the remainder of this paper.

A. Benefits of the Sheet Representation

The choice of image measurement schemes has typically been constrained by the *full-frame world view*. In this predominant view, an entire frame of video is captured by an imaging sensor and transported to a processor system. Advances in sensor design and the pressing need for low-power distributed systems necessitates exploring alternatives.

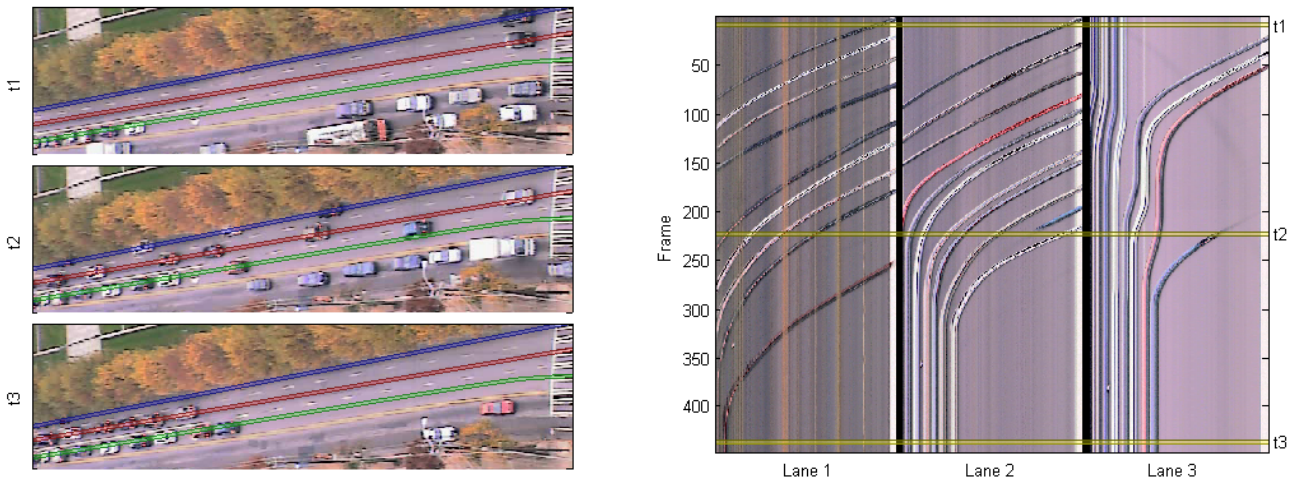

Fig. 3. The image on the left shows three example frames from a video of a traffic scene. Three sheet curves are highlighted in red, green and blue, respectively. The image on the right shows an example of sheets extracted from the three marked lanes for 450 frames of video. The curved streaks visible in the sheets show the cars' progress moving from right to left in the image, and coming to a stop for a light (just outside the field of view). The sheet on the left corresponds to the top lane; occlusions in that lane are visible as dominant vertical streaks. A lane change is also visible in this figure: a blue car enters the scene in lane two, its blue streak disappears as it changes lanes at about frame 225, and it then appears in the sheet for lane three. This is visible in the original video frame on the left at time t2, where the blue car can be seen between the red and green lanes.

A significant benefit of working with the sparse sheet representation is the greatly reduced bandwidth requirements. For example, a 640×480 video with five lanes of traffic can be reduced to approximately 1% of the original data. It is conceivable that this data reduction could be integrated deeper into the imaging system to reduce the image data bandwidth directly in the imaging sensor. This has the potential to greatly reduce the power requirements for visual tracking applications.

The ability to inspect a long temporal window can also significantly increase the accuracy of vision algorithms. In the *full-frame world view*, memory constraints significantly limit the possible duration of the temporal window. In our system, using the sheet representation allows temporal windows that are approximately 100 times longer than traditional models.

Thus these sheets offer a novel representation, allowing long temporal windows with low memory requirements and new approaches for a number of distributed camera applications, including object counting, scene activity analysis, and anomaly detection. In the remainder of this work we focus on the challenging problem of multiple object tracking. We describe a system capable of fast and accurate tracking of a large number of objects in a challenging urban environment. We begin with a description of our method for extracting object tracklets directly from sheets.

III. Local Tracking within Space-Time Sheets

The second stage of our tracking system (see Fig. 2 for a pictorial overview) extracts a set of tracklets, separately, from each space-time sheet. The tracklets from all sheets are used as input to the global data association algorithm described in Section IV. We follow the approach of [6], which we summarize in this section.

The tracklet generation process consists of two main steps: generating candidate moving object detections and temporally extending candidate locations. The first stage generates a set of 1D moving object detections. Each detection marks the possible location and scale of a moving object. The features used for detection are based on a color background model and local-motion consistency [16]. The second stage filters out false-positive detections and temporally extends detections. The temporal extension process uses a template-based appearance model coupled with an acceleration penalty to track forward and backward in time from the detections. This enables tracklets to be generated in regions in which few detections were found.

The output of the tracklet generation stage is the set of tracklets for each sheet, mapped onto world coordinates.

IV. Data Association between Sheets and Cameras

The final stage of our system combines tracklets detected from different sheets throughout the camera network into a set of complete object tracks. The input is a set of tracklets, which we denote as $\mathcal{S} = \{S_j\}$, where each tracklet, S_j, contains a sequence of world-space position observations $Y_j = \{\mathbf{y}_j^1, \mathbf{y}_j^2, ..., \mathbf{y}_j^{l_j}\}$, an appearance descriptor, $\mathbf{a}_i$, and its start and end frames, t_j^s and t_j^e. Our desired output is a set of tracks $\mathcal{T} = \{T_k\}$, where each track is a sequence of tracklets, $T_k = \{S_{k_1}, S_{k_2}, ..., S_{k_{|T_k|}}\}$.

We follow the formulation presented by Zhang et al. [21], but extend it from single-frame detections to tracklets. The data association problem is expressed as a MAP estimation

over the posterior probability of $\mathcal{T}$ given the tracklets $\mathcal{S}$:

$$\mathcal{T}^* = \arg\max_{\mathcal{T}} P(\mathcal{T}|\mathcal{S})$$

Assuming the likelihood probabilities are conditionally independent given $\mathcal{T}$ and that objects move independently of one another, this can be simplified to

$$\begin{aligned}\mathcal{T}^* &= \arg\max_{\mathcal{T}} P(\mathcal{S}|\mathcal{T})P(\mathcal{T}) \\ &= \arg\max_{\mathcal{T}} \prod_{S_i \in \mathcal{S}} P(S_i|\mathcal{T}) \prod_{T_k \in \mathcal{T}} P(T_k) \\ &\quad \text{s.t. } T_k \cap T_l = \emptyset, \forall k \neq l\end{aligned}$$

The first term is defined as follows

$$P(S_j|\mathcal{T}) = \begin{cases} 1-\beta_i, & \text{if } \exists T_k \in \mathcal{T}, S_j \in T_k \\ \beta_i & \text{otherwise.} \end{cases}$$

where β_i is the probability that S_i is a false positive. The second term is the probability of track T_k, which can be modeled as a Markov chain

$$\begin{aligned}P(T_k) =& P(\{S_{k_1}, S_{k_2}, ..., S_{k_n}\}) \\ =& P(I(k_1))P(L(k_1,k_2))...P(L(k_{n-1},k_n))P(T(k_n))\end{aligned}$$

with indicator functions

$$\begin{aligned}I(i) &\equiv \exists T_k \in \mathcal{T}, S_i \text{ is the first tracklet in } T_k \\ T(i) &\equiv \exists T_k \in \mathcal{T}, S_i \text{ is the last tracklet in } T_k \\ L(i,j) &\equiv \exists T_k \in \mathcal{T}, S_j \text{ immediately follows } S_i \text{ in } T_k\end{aligned}$$

representing the initialization, termination, and ordering of tracks.

Given estimates of β_i, $P(I(i))$, and $P(T(i))$ for each tracklet S_i and $P(L(i,j))$ for each pair of tracklets S_i and S_j, this formulation of the data association problem can be cast as a minimum-cost flow problem and solved optimally and efficiently.

A. Data association model

We now describe our model of the four probability terms described above.

False positives: The generation of tracklets can be modeled as a random process, where the tracklet generation stage initiates a false tracklet with some probability and then iteratively tracks forward and backward from this false initialization until a termination condition is reached. Under this model, the probability that a tracklet S_i of length l_i is a false positive is

$$\beta_i = B_0 e^{\lambda_B \cdot l_i}$$

where B_0 is the probability of initiating a false tracklet and λ_B is the log probability of continuing to extend a false tracklet for another frame.

Track initialization/termination: Assuming a track can enter or exit the scene at any point, then the initialization and termination probabilities of each tracklet are defined as

$$P(I(i)) = P(T(i)) = \frac{E(|\mathcal{T}^*|)}{|\mathcal{S}|}$$

where $E(|\mathcal{T}^*|)$ is the expected number of objects in the scene and $|\mathcal{S}|$ is the number of tracklets. As mentioned in [21], when a suitable estimate of $|\mathcal{T}^*|$ is not available, an EM approach can be used to estimate $P(I(i))$ and $P(T(i))$ during the optimization.

Tracklet linking: To derive our link scores, we start by assuming $P(L(i,j))$ depends only on S_i and S_j, and apply Bayes' rule to get

$$\begin{aligned}P(L(i,j)|S_i,S_j) &= \frac{P(S_i,S_j|L(i,j))P(L(i,j))}{P(S_i,S_j)} \\ &= \left(1 + b \cdot \frac{P(S_i,S_j|\neg L(i,j))}{P(S_i,S_j|L(i,j))}\right)^{-1}\end{aligned}$$

where

$$b = \frac{P(\neg L(i,j))}{P(L(i,j))} = \frac{1-P(L(i,j))}{P(L(i,j))}$$

is the prior odds against any two tracks being linked.

Assuming that S_i and S_j are independent when $L(i,j)$ is false, and assuming that the tracklet motion, appearance, and start/end frames are independent of each other, we can decompose this term further:

$$\begin{aligned}P(L(i,j)|S_i,S_j) &= \left(1 + b \cdot \frac{P(S_i)P(S_j)}{P(S_i)P(S_j|S_i)}\right)^{-1} \\ &= \left(1 + b \cdot \frac{P(S_j)}{P(S_j|S_i)}\right)^{-1} \\ &= \left(1 + b \cdot \frac{P(Y_j)}{P(Y_j|Y_i)} \frac{P(a_j)}{P(a_j|a_i)} \frac{P(t_j^s)}{P(t_j^s|t_i^s,t_i^e)}\right)^{-1}\end{aligned}$$

reducing it to its kinematic, appearance, and temporal terms.

Kinematic model: To compute the kinematic terms, we model each object's motion as a linear dynamic system, where the object's state at each frame is represented as a 4-dimensional vector comprised of its 2-D position and velocity in world space, and the state is updated according to the process matrix

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and with Gaussian process noise

$$Q = \sigma_a^2 \cdot \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{bmatrix}.$$

Each tracklet's trajectory through world-space, Y_i, can be expressed as a sequence of observations drawn from this model according to the measurement matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and with Gaussian measurement noise

$$R = \sigma_m^2 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Under a linear dynamic model, the probability of observing a given sequence of measurements, $P(Y_j)$, can be computed using a Kalman filter. To compute $P(Y_j|Y_i)$ under our model, we first use Y_i to estimate the state of the model at the start of S_j, $\mathbf{x_{t_j^s}}$, after which a Kalman filter can be used to compute the probability of observing Y_j using the new estimate of the initial state. When t_j^s is after t_i^e, an optimal estimate of $\mathbf{x_{t_j^s}}$ from Y_i can be computed using a Kalman filter; however, when t_j^s is before t_i^e (i.e. when two tracklets overlap in time), the Kalman filter's estimate of $\mathbf{x_{t_j^s}}$ does not take into account measurements after t_j^s. In this case, the optimal state estimate can be computed using a Kalman smoother.

Because $P(L(i,j))$ must be computed for all pairs of i and j, it is important that this probability can be computed quickly. While $P(Y_j)$ can be precomputed for all S_j, $P(Y_j|Y_i)$ must be computed for each pair, and for long tracklets, where the set of measurements Y_j is large, this can be expensive to compute. We note that

$$\frac{P(\{\mathbf{y}_j^1, ..., \mathbf{y}_j^N\})}{P(\{\mathbf{y}_j^1, ..., \mathbf{y}_j^N\}|Y_i)} \approx \frac{P(Y_j)}{P(Y_j|Y_i)}$$

as N approaches $|Y_j|$. Thus, in practice, a close approximation can be obtained without filtering over the entire tracklets. In our experiments, we terminate after 30 frames.

Appearance model: We represent the appearance of each tracklet, S_i, with a vector, $\mathbf{a}_i$, denoting the mean color of all pixels covered by the tracklet. Letting $N(\mathbf{x}; \mu, \mathbf{\Sigma})$ denote a Gaussian distribution with mean μ and covariance $\mathbf{\Sigma}$, we define our appearance terms as follows:

$$P(\mathbf{a}_j) = N(\mathbf{a}_j; \mu_a, \mathbf{\Sigma}_a)$$

where μ_a and $\mathbf{\Sigma}_a$ describe the distribution of appearance vectors over all tracklets, and

$$P(\mathbf{a}_j|\mathbf{a}_i) \begin{cases} N(\mathbf{a}_j; \mathbf{a}_i, \mathbf{\Sigma}_s) & c_i = c_j \\ N(\mathbf{a}_j; \mathbf{a}_i, \mathbf{\Sigma}_d) & c_i \neq c_j \end{cases}$$

with $\mathbf{\Sigma}_s$ and $\mathbf{\Sigma}_d$ representing how consistent a single object's color is between two tracklets. Because color cues are typically much less reliable between cameras than they are within a single camera, the distribution of $\mathbf{a}_j$ given $\mathbf{a}_i$ is conditioned on whether or not the two tracklets were observed by the same camera.

Temporal model: We represent the temporal extent of each tracklet with its start and end frames, t_i^s and t_i^e. The first temporal term $P(t_j^s)$ expresses the prior probability that a tracklet will begin at a particular frame. Assuming tracklets are uniformly distributed throughout video sequence of length T, this is simply

$$P(t_j^s) = \frac{1}{T}.$$

The second term describes the temporal relationship between subsequent tracklets. We model this relationship with three cases.

$$P(t_j^s|t_i^s, t_i^e) = \begin{cases} 0 & t_j^s < t_i^s \\ \frac{1}{Z_t} & t_i^s \leq t_j^s < t_i^e \\ \frac{1}{Z_t} e^{\lambda_r \cdot (t_j^s - t_i^e)} & t_j^s \geq t_i^e \end{cases}$$

The first case enforces the ordering of tracklets, expressing that the second tracklet in a sequence will never begin before the first. In the second case, the two tracklets are in the proper order but overlap in time; this is common in networks with overlapping camera views, where the same object may be detected in two different views at the same time. The final case models the distribution of frame-gap $t_j^s - t_i^e$ between two tracklets of the same object, which we represent as an exponential fall-off, where λ_r is the miss rate of the tracklet detector and Z_t is a normalizing constant.

Despite our extensions to make this MAP data association apply to tracklets, it remains in the same format as that of [21], and therefore is solvable by an efficient network flow algorithm.

V. Experiments

We validated our approach on the NGSIM Peachtree data set [13]. This data set was captured through a collaboration of researchers interested in modeling vehicle behavior [1]. A series of cameras were set up viewing consecutive parts of Peachtree Street in Atlanta, Georgia, and 15 minutes of data were simultaneously recorded in each. Figure 4 shows a view from each camera and its associated field of view on an aerial photograph.

Extensive ground truth is freely available within the data set, including geo-referenced lane centerlines, intersection locations and traffic light cycles. Trajectories of many hundreds of vehicles were tracked through the scene using a semi-automated tracking system with extensive hand corrections.

A. Tracking Metrics

To provide a quantitative evaluation of our system, we measure the *track completeness factor* and *track fragmentation* as defined by Perera et al. [15]. We compute these metrics as follows:

Given a set of estimated tracks, T, and ground truth tracks, G, we find an optimal-cost association, A^*, from each track in T to a corresponding track in G. The cost of an association is defined as the sum of distances between the estimated tracks and their associated ground truth tracks, where the distance between two tracks is defined as the average Euclidean distance between the estimated object position and ground truth object position over all frames. The optimal association, A^*, is the association that minimizes this cost. From the optimal association, we compute two performance metrics. The first

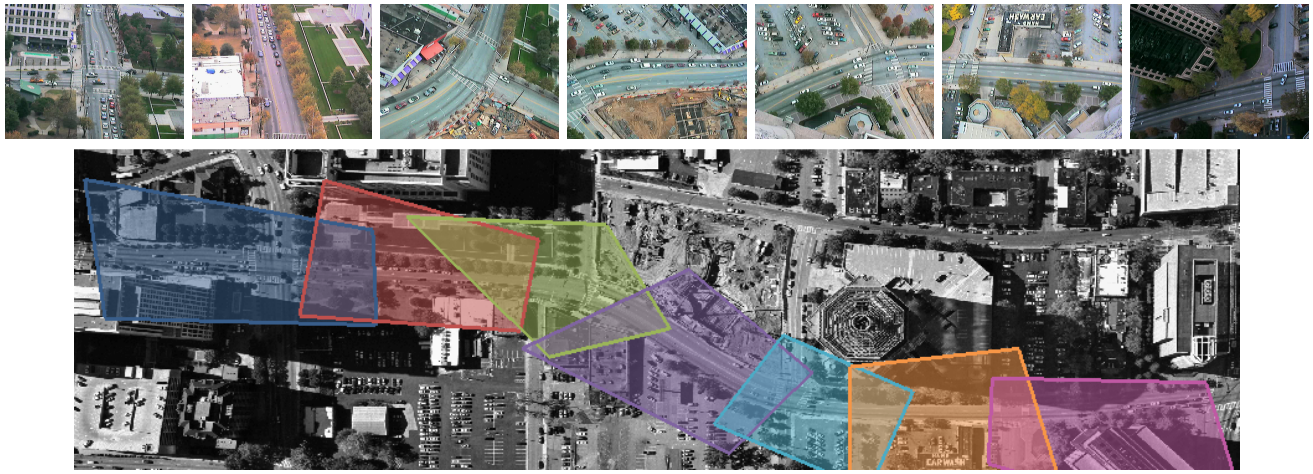

Fig. 4. The top row shows a view from seven cameras in the NGSIM Peachtree camera network. Cameras simultaneously captured 15 minutes of data along consecutive parts of Peachtree Street in Atlanta, Georgia. The bottom row shows each camera's field of view plotted on an aerial photograph of the area.

is the *track completeness factor* (TCF), which provides a measure of how well each object is detected. The second is the *track fragmentation* (TF), which provides a measure of how well the identity of a track is maintained. These are defined as follows:

$$TCF = \frac{\sum_i \sum_{T_j \in A(G_i)} |O(T_j, G_i)|}{\sum_i |G_i|}$$

$$TF = \frac{\sum_i |A(G_i)|}{|\{G_i | A(G_i) \neq \phi\}|},$$

where $A(G_i)$ is the set of estimated tracks associated with track G_i in the optimal association, A^*; $O(T_j, G_i)$ denotes the frames where T_j and G_i overlap; and $|\cdot|$ denotes the cardinality operator.

B. Tracking Results

We evaluated our system on seven videos from the NGSIM Peachtree data set. Ground truth vehicle positions are provided in world-space coordinates for the first 10,000 frames of the data set. However, any vehicles present in the scene at frame 1 were not included in the ground truth set. Because of this, we discard the first 2,000 frames of video, ensuring that all unlabeled vehicles have left the scene. We also note that ground truth positions and identities were only provided for vehicles traveling on or through Peachtree Street. Thus, while vehicles on cross streets are tracked by our system, these tracks do not appear in the ground truth and cannot be empirically validated. This may artificially increase the track fragmentation score at times, as vehicles without a corresponding ground truth track may be erroneously associated with another ground truth track during evaluation, even when the estimated vehicle is tracked accurately.

For our experiments, we divided the sequence into four 2,000-frame subsequences and performed tracking on each independently. We then compared our estimated tracks to the ground truth tracks and computed the TCF and TF scores. We report the mean and standard deviation over the four trials in Table I. The bottom shows the overall performance in tracking cars throughout the entire network. In addition, we also report the TCF and TF scores for each camera. To compute this, we compare our estimated tracks to the ground truth tracks over only the portions of the tracks that fall within a particular camera's field of view. We emphasize that the tracking results were obtained by performing global data association over the entire camera network; the scores reported by individual camera simply reflect how well the global tracking algorithm performed at different parts of the network. Breaking down the performance metrics by camera also enables us to make a direct comparison to previously reported results [6] for the single camera case.

	Single-camera		Multi-camera	
Camera	TCF	TF	TCF	TF
1	0.36 ± 0.17	1.36 ± 0.32	0.60 ± 0.04	1.28 ± 0.28
2	0.61 ± 0.05	1.25 ± 0.37	0.80 ± 0.03	1.25 ± 0.22
3	0.71 ± 0.12	1.38 ± 0.31	0.91 ± 0.05	1.31 ± 0.35
4	0.77 ± 0.07	1.14 ± 0.15	0.92 ± 0.02	1.23 ± 0.23
5	0.62 ± 0.07	1.35 ± 0.22	0.70 ± 0.03	1.16 ± 0.17
6	0.63 ± 0.04	1.29 ± 0.18	0.57 ± 0.09	1.14 ± 0.17
7	0.46 ± 0.07	1.40 ± 0.10	0.42 ± 0.11	1.13 ± 0.20
All	N/A	N/A	0.67 ± 0.04	1.39 ± 0.32

TABLE I
THE TRACK COMPLETION FACTOR (TCF) AND TRACK FRAGMENTATION (TF) FOR THE NGSIM PEACHTREE DATA SET.

Our system processed each video at approximately 20-30 FPS running in Matlab 2007b on a desktop workstation with a 2.66 GHz Intel Xeon processor. Figure 5 shows three example complete trajectories through the camera network.

VI. DISCUSSION

The NGSIM dataset, a publicly available data set with hand-coded ground truth, provides the best data set we know of to quantitatively evaluate multi-camera, multi-vehicle tracking algorithms on real data. This data set includes multiple views of a dense traffic scene with stop-and-go driving patterns, numerous partial and complete occlusions, and several inter-

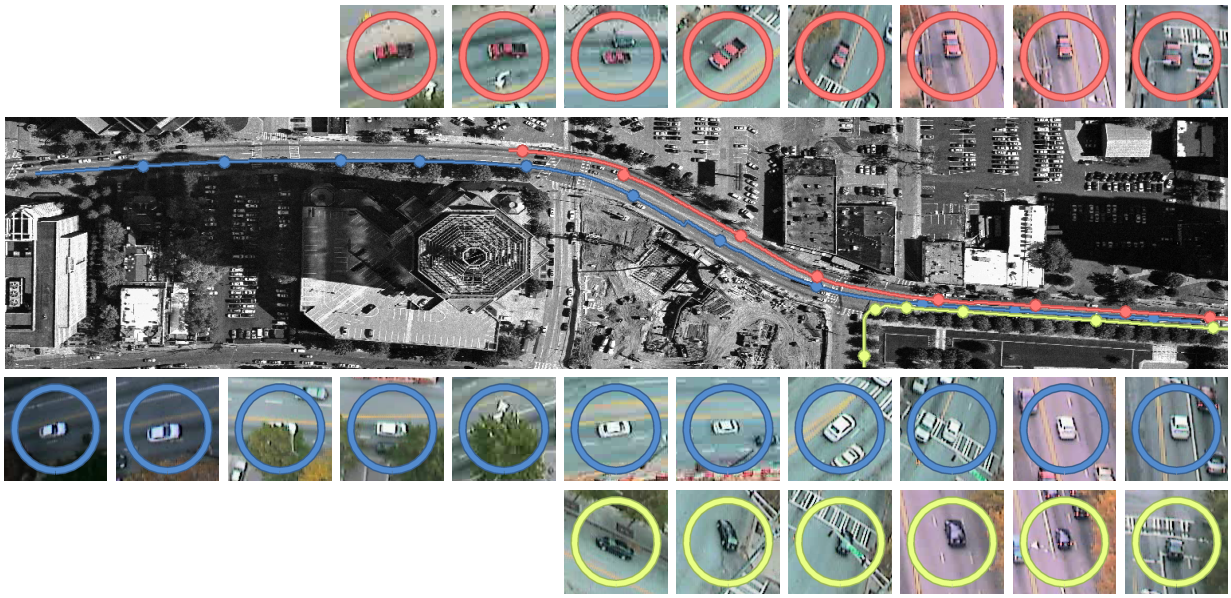

Fig. 5. A visualization showing three of the 622 complete tracks found by our system. The first vehicle (highlighted in red) enters the scene on the right, travels through 5 cameras and turns into a parking lot. The second vehicle (highlighted in blue) enters the network on the left, is tracked through several occlusions and exits the scene at the far end of the network. The third vehicle (highlighted in green) enters from a side street and is tracked to its exit through three cameras. Lane changes are visible in both the red and blue tracks, denoted as a lateral shift.

sections.

We find the most interesting result in the comparison between the left and right half of Table I. In each case, the cameras run the same algorithm to discover tracklets within their own data. The improvement is due to higher confidence made possible by long tracks; the neighboring cameras provide strong boundary conditions for the tracks in a given camera. Additionally, we note that our overall tracklet linking and fragmentation scores, across all seven cameras, compares favorably to results reported in the literature for these error metrics on a single camera [15].

VII. Conclusion

This framework decomposes a 2D tracking problem into a collection of 1D tracking problems, exploiting the fact that in urban areas vehicles often remain within lanes. This framework is ideally suited to implementation on a camera network because it provides a way to extract data from a live video stream which is real time and which gives a small set of tracklets. We do not know of other methods that perform both these tasks. Template tracking can give long high quality tracks as objects move within a scene, but remains relatively expensive to compute. Detect-and-connect trackers have low front end costs, but generate many detections which potentially need to be shared across cameras. While there are many potential approaches to this problem, the approach presented here offers one concrete probabilistic framework that fits within current computational capabilities. The framework runs in real time, offers a concrete way of distributing the image processing workload, and demonstrates system performance on par with single camera systems that track over much shorter distances. This suggests that this framework is likely to scale to much larger camera networks.

Finally, we would like to acknowledge the foresight of the NGSIM organization in making the video data set and the hand annotations publicly available. This makes the types of quantitative evaluations we show here possible, and supports quantitative comparisons between algorithms in this problem domain, where frequently the data collection and data annotation challenges make such sharing less common.

References

[1] Vassili Alexiadis, James Colyar, and John Halkias. A model endeavor: A public-private partnership is working to improve traffic microsimulation technology. *FHWA Public Roads*, 2007.

[2] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 744–750, June 2006.

[3] Robert C. Bolles, H. Harlyn Baker, and David H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[4] Antonio Criminisi, Sing Bing Kang, Rahul Swaminathan, Richard Szeliski, and P. Anandan. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Comput. Vis. Image Underst.*, 97(1):51–85, 2005.

[5] Ran Eshel and Yael Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[6] Nathan Jacobs, Michael Dixon, Scott Satkin, and Robert Pless. Efficient tracking of many objects in structured environments. Technical Report WUCSE-2009-9, CSE, Washington University, St. Louis, MO, USA, April 2009.

[7] Omar Javed, Khurram Shafique, Zeeshan Rasheed, and Mubarak Shah. Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision Image Understanding*, 109(2):146–162, 2008.

[8] Hao Jiang, S. Fels, and J.J. Little. A linear programming approach for multiple object tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[9] Gabin Kayumbi, Pier Luigi Mazzeo, Paolo Spagnolo, Murtaza Taj, and Andrea Cavallaro. Distributed visual sensing for virtual top-view trajectory generation in football videos. In *Proceedings of the International conference on Content-based image and video retrieval*, pages 535–542, 2008.

[10] V. Kellokumpu, G. Zhao, and M. Pietikäinen. Human activity recognition using a dynamic texture based method. In *Proc. of The British Machine Vision Conference (BMVC 2008)*, 2008.

[11] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Proc. IEEE International Conference on Computer Vision*, pages 1–8, Oct. 2007.

[12] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor. Detection and classification of highway lanes using vehicle motion trajectories. *Intelligent Transportation Systems, IEEE Transactions on*, 7(2):188–200, June 2006.

[13] Summary report, ngsim peachtree street (atlanta) data analysis. http://www.ngsim.fhwa.dot.gov/, June 2007.

[14] S.A. Niyogi and E.H. Adelson. Analyzing and recognizing walking figures in xyt. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–474, 1994.

[15] A. G. Amitha Perera, Chukka Srinivas, Anthony Hoogs, Glen Brooksby, and Wensheng Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–673, Washington, DC, USA, 2006. IEEE Computer Society.

[16] Robert Pless. Detecting roads in stabilized video with the spatio-temporal structure tensor. *Geoinformatica*, 10(1):37–53, 2006.

[17] Yang Ran, Rama Chellappa, and Qinfen Zheng. Finding gait in space and time. *Proc. International Conference on Pattern Recognition*, 4:586–589, 2006.

[18] Yann Ricquebourg and Patrick Bouthemy. Real-time tracking of moving persons by exploiting spatio-temporal image slices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):797–808, 2000.

[19] T.N. Schoepflin and D.J. Dailey. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 4(2):90–98, June 2003.

[20] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.

[21] Li Zhang, Yuan Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.