Inference of Non-Overlapping Camera Network Topology by Measuring Statistical Dependence

Kinh Tieu Gerald Dalley W. Eric L. Grimson

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

We present an approach for inferring the topology of a camera network by measuring statistical dependence between observations in different cameras. Two cameras are considered connected if objects seen departing in one camera are seen arriving in the other. This is captured by the degree of statistical dependence between the cameras. The nature of dependence is characterized by the distribution of observation transformations between cameras, such as departure to arrival transition times, and color appearance.

We show how to measure statistical dependence when the correspondence between observations in different cameras is unknown. This is accomplished by non-parametric estimates of statistical dependence and Bayesian integration of the unknown correspondence. Our approach generalizes previous work which assumed restricted parametric transition distributions and only implicitly dealt with unknown correspondence. Results are shown on simulated and real data. We also describe a technique for learning the absolute locations of the cameras with Global Positioning System (GPS) side information.

1. Introduction

Consider the problem of wide-area surveillance, such as traffic monitoring and activity classification around critical assets (*e.g.*, an embassy, a troop base, critical infrastructure facilities such as oil depots, port facilities, airfield tarmacs). We wish to monitor the flow of movement in such settings from a large number of cameras, typically with non-overlapping fields of view. To coordinate observations in these distributed cameras, we need to know the connectivity of movement between fields of view (*i.e.*, when an object leaves one camera, it is likely to appear in a small number of other cameras with some probability). A simple example with two cameras imaging the upstream and downstream sections of a road is shown in Figure 1. We want to infer that objects leaving the upstream view are likely to transition to the downstream view. We also want to infer the distribution of transition times between the two views.

upstream downstream

Figure 1: Upstream to downstream movement of objects.

In some instances, one can carefully site and calibrate the cameras so that observations are more easily coordinated. However even with calibrated cameras, the departure/arrival locations, connectivity, and transition time distribution still have to be learned. In many cases, cameras must be rapidly deployed and may not last for long periods of time. Hence we seek a passive way of determining the topology of the camera network. That is, we want to determine the network structure relating cameras, and the typical transitions between cameras, based on noisy observations of moving objects in the cameras. Departure and arrival locations in each camera view are nodes in the network. An arc between a departure node and an arrival node denotes connectivity (likely transition). We want to infer both the topology (*i.e.*, which arcs exist) and the transition times. For the example in Figure 1 we want to infer that the views are connected and estimate the transition time distribution.

If we could identify the same object in different cameras (*e.g.*, using a license plate reader or face recognition system), then learning the topology and transitions would be easy. In practice, in wide-area surveillance, the correspondence between observations in different cameras is difficult to obtain because cameras may be widely separated and the observations may occupy only a few pixels. A key feature we can exploit is time of arrival and departure. It can be measured accurately by tracking in individual cameras. Other features such as object appearance can also be used.

In this paper, we hypothesize that given the correspondence, the transition time distribution is highly structured (*i.e.*, has low entropy). For example, we expect different modes in the transition time distribution for vehicles and pedestrians with possibly some outliers. The contributions of this paper are a formal definition of connectivity in terms of statistical dependence and an algorithm for integrating out the unknown correspondence. We show encouraging results on synthetic and real data.

2. Previous Work

Previous work on tracking across multiple cameras generally either assumed known camera topology or known correspondence. Methods which use assignment algorithms for tracking across multiple cameras such as [7, 9, 8] assume the transition models are known or fit them with hand-labeled correspondences. Other work [3, 14] for calibration also assume known correspondence.

Makris *et al.* [11] have tackled the problem of estimating a multi-camera topology from observations. They assume a single mode transition distribution and exhaustively search for the location of the mode. Their method assumes all departure and arrival pairs within a time window are implicitly corresponding. The distribution of transition times obtained from this correspondence is examined for a peak by thresholding based on the mean and standard deviation. Essentially the correlation between arrival and departure times is computed using a loose, implicit notion of correspondence. They show promising results using this method.

Correlation is effective for monotonic relationships in general, but is not flexible enough to handle multi-modal distributions. Makris *et al.* [11] have acknowledged this fact, which can occur when both cars and pedestrians are part of the observations. Their approach essentially assumes a Gaussian transition distribution and implicit true correspondences within a chosen time window. However for a given departure observation, the true correspondence is a single arrival observation. So for all observations within a time window, the true and false correspondences generate a mixture of the true and false transition distribution. The time window size and distribution of observations determines the number of false correspondences versus the single true correspondence. In general the more dense the observations and the longer the transition time, the more false correspondences. Thus their method suffers from assuming a unimodal transition distribution, and only implicitly dealing with correspondence.

Our method generalizes their approach to more flexible, multi-modal transition distributions, and explicitly handles correspondence. This is accomplished by using a more general information theoretic notion of statistical dependence, and integrating out the uncertain correspondence in a Bayesian manner. Our approach makes very few assumptions and does not require supervision.

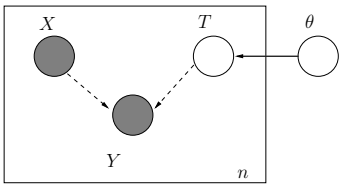


Figure 2: Graphical model of departures X, arrivals Y, and transformation T parameterized by θ.

3. Problem Formulation

To infer the topology of a camera network, assume we have identified arrival and departure locations and observations in each camera. For example, this can be done with a blob-based tracker in each camera separately [16]. For each pair of cameras, we want to infer whether they are connected and the distribution of transition times. Recall that this is made more difficult because the correspondence between observations in different cameras is unknown.

Suppose we are given observations of departure $x_1, .., x_n$ and arrival $y_1, .., y_n$ times in two connected cameras, respectively. Also, assume that the correspondence between the observations is given by a permutation π of the indices such that $(x_i, y_{\pi(i)})$ is a corresponding pair. We formalize this by writing

$$y_{\pi(i)} = t_i(x_i), \tag{1}$$

where the distribution of transformation t_i is parameterized by θ. This is illustrated in the graphical model shown in Figure 2. Shaded vertices are observed, plates (boxes) denote repetitive structure. Solid arrows denote direct probabilistic dependence, while dashed arrows denote direct functional dependence. Note that each observed pair $(x_i, y_{\pi(i)})$ is associated with its own hidden transformation t_i. In this model, a fixed transformation distribution is chosen by sampling a θ. Then, n observations are generated by repeatedly sampling an x_i, a t_i given the chosen θ, and finally generating $y_{\pi(i)}$ as function of x_i and t_i. Given true correspondences we would know that two cameras are dependent. Thus with unknown correspondence, strength of statistical dependency is a good measure for jointly inferring correspondence and camera connectivity.

For departure and arrival time observations X and Y, the transformation T is an additive transition time between cameras: $Y = X + T$. Our formulation also captures other transformations such as color variations between cameras. We will show this in the experiments. Based on our formulation, both the degree and nature of dependence is determined by the distribution of T, which is determined by θ. Basically T tells us how X and Y are related, and the randomness in T indicates strength of dependence. In particu-

lar, strong dependence means that observations y are highly predictable given x. This will be reflected by low entropy in the distribution of T. So the degree of statistical dependence measures how connected two cameras are, and the nature of this dependency is encoded in the corresponding distribution of transition times.

3.1. Mutual Information as a Measure of Dependence

Recall that our problem is to measure the degree and nature of statistical dependence between observations in two cameras. The mutual information (MI) [15, 18]

$$I(X;Y) = \int\int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy. \tag{2}$$

between two random variables X and Y is a natural measure of dependence. It is the average log likelihood ratio between the joint distribution (dependent case) and the product of marginal distributions (independent case). Wolf [19] has shown that MI is the dominating term in a Bayesian measure of independence. When the joint distribution is very different from the independent distribution, there is a strong dependence between X and Y, and $I(X;Y)$ is large.

We can write MI in terms of entropy,

$$I(X;Y) = h(Y) - h(Y|X), \tag{3}$$

where entropies are defined as

$$h(Y) = -E_Y[\log p(Y)] \tag{4}$$

$$h(Y|X) = -E_X\{E_Y[\log p(Y|X)]\}. \tag{5}$$

Thus MI is a measure of the reduction in uncertainty of one random variable given knowledge of another.

We can use equation (3) to compute MI between the observations in two cameras. We can relate $h(Y|X)$ to the entropy of T by recalling the graphical model shown in Figure 2. Then using the chain rule for entropy,

$$h(Y,T|X) = h(T|X) + h(Y|T,X) \tag{6}$$

$$= h(T) + h(Y|T,X) \tag{7}$$

because X is independent of T, and

$$h(Y,T|X) = h(Y|X) + h(T|Y,X). \tag{8}$$

Assume that $Y = T(X)$ can be determined given X and T up to fidelity $\epsilon > 0$, and that T can be determined given X and Y also up to fidelity ϵ [15]. Combining the above yields

$$h(Y|X) = h(T), \tag{9}$$

$$I(X;Y) = h(Y) - h(T). \tag{10}$$

So as previously asserted, the degree and nature of dependence between X and Y is governed by the distribution of transformations T, which for our model depends on the hidden parameter θ.

3.2. Entropy Estimation

To compute $I(X;Y)$ we need to compute $h(Y)$ and $h(T)$. We use the Parzen density estimator [12] to compute entropy from samples. We show this for T; it similarly applies to Y. The Parzen density estimate is

$$\hat{p}(t) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{t-t_i}{\sigma}\right) \tag{11}$$

where σ is the bandwidth, K is the kernel, and $t_1, ..., t_n \in R^d$ are iid samples. The Gaussian function makes a convenient kernel. Parzen density estimators are simple yet flexible enough to fit densities with multiple modes.

We can approximate the entropy

$$h(T) = -E[\log p(T)] \approx -\frac{1}{n}\sum_{i=1}^{n} \log \hat{p}(t_i). \tag{12}$$

This is just the negative of the average log likelihood of a set of samples. It shows how the entropy of the transformation is related to their likelihood which in turn determines the degree of statistical dependence in our model.

For uni-dimensional data we use the faster m-spacings estimate [17, 10]. The estimate is

$$h(T) \approx \frac{1}{n}\sum_{i=1}^{n-m} \log\left(\frac{n}{m}(t_{i+m} - t_i)\right) \tag{13}$$

where the subscripts for t denote ordering. The primary computation is sorting the data to obtain order statistics.

3.3. Posterior Expectation

Now we know how to compute statistical dependence in terms of MI and entropies. Given observations $O = \{x_i, y_i\}$ we write the posterior expectation as

$$E_\theta[I(X;Y)] = \int_{\theta'} \{h(Y) - h[T|\theta']\} p(\theta'|O) d\theta' \tag{14}$$

$$= h(Y) - \int_{\theta'} h[p(T|\theta')] \frac{p(O|\theta')p(\theta')}{p(O)} d\theta' \tag{15}$$

$$= h(Y) - \int_{\theta'} h[p(T|\theta')] \tag{16}$$

$$\prod_{i=1}^{n} \frac{\int_{t_i} p(x_i, y_i, t_i|\theta')p(\theta')dt_i}{p(O)} d\theta'. \tag{17}$$

Note that the dependence integrates over all transformations and all transformation distributions. We can choose the *maximum a posteriori* (MAP) θ^* and t^* given θ^* as the representative dependence and transformation. This is reasonable if the posterior distribution of θ and $p(T|\theta^*)$ is unimodal and sharply peaked. We expect this to be the case for simple transformations such as transition time. We could also sample transformation distributions and transformations from the posterior distribution of θ and $p(T|\theta)$ and compute their resulting statistical dependence. In practice,

we represent the MAP θ^* implicitly by modeling the distribution of T nonparametrically. The assumed functional relationship between X, Y, and T also simplifies the integral with respect to T to a single term. So the computation of $I(X;Y)$ only requires computing $h(Y)$ and $h(T)$.

4. Correspondence

So far we assume a known correspondence π between the observations X and Y in both cameras. This allows us to relate the nature of the statistical dependency between X and Y with the distribution of transformations T. We define statistical dependence as mutual information $I(X;Y)$, and compute it via the entropies $h(Y)$ and $h(T)$.

In reality, the correspondence π is unknown so we must integrate it out as a nuisance variable,

$$E_\theta[I(X;Y)] = E_\theta[E_\pi\{I(X;Y_\pi)\}]. \tag{18}$$

Using the single MAP correspondence to approximate the expectation is problematic because $I(X;Y_\pi)$, in general, does not vary smoothly with permutations π. The MAP correspondence may have incorrect correspondences which can dramatically change the estimated transformations T. This is because of the discrete structure of permutations. Thus the correspondence problem does not admit a simple MAP approximation to the full Bayesian solution as for MI.

4.1. Markov Chain Monte Carlo

The combinatorial nature of permutations makes computation of the expectation (18) by direct enumeration intractable. Markov Chain Monte Carlo (MCMC) [5, 13, 2, 4] is a way to perform approximate inference in this case. We want to approximate

$$E_\theta[E_\pi\{I(X;Y_\pi\}] \approx \frac{1}{n}\sum_{j=1}^{n} E_\theta[I(X;Y_{\pi_j})], \tag{19}$$

which requires samples from the posterior distribution of correspondences $p(\pi)$. We use the Metropolis-Hastings algorithm [6] (see Algorithm 1) to do this. The initial sample is a random correspondence (permutation). New samples are obtained by conditionally sampling a new correspondence given the current one via a proposal distribution $q(\pi'|\pi_j)$. We use three different types of proposals for sampling correspondences: (1) add a match, (2) delete a match, (3) flip two matches. This allows the algorithm to sample new correspondences by swapping matches. The ability to add and delete matches handles missing correspondences as described in next section. It also enables matches to be swapped without generating samples with highly improbable matches in the process. The proposals are simple to implement and work well in our experience. The new sample is accepted with probability proportional to the relative likelihood of the new sample vs. the current one. The likelihood of a correspondence is proportional to the log probability of the corresponding transformations, which we compute as $-h(T)$ as described in the previous section. The algorithm repeats this process for the desired number of samples. We compute $I(X;Y_{\pi_j})$ for each sample π_j and take the average as the expected posterior MI.

Algorithm 1 Metropolis-Hastings

1. Initialize π_0; $j = 0$.
2. **loop**
3. Sample π' from $q(\cdot|\pi_j)$.
4. Sample U from $U(0,1)$.
5. Let $\alpha(\pi_j, \pi') = \min\left(1, \frac{p(\pi')q(\pi_j|\pi')}{p(\pi)q(\pi'|\pi_j)}\right)$.
6. **if** $U \leq \alpha(\pi_j, \pi')$ **then**
7. $\pi_{j+1} = \pi'$.
8. **else**
9. $\pi_{j+1} = \pi_j$.
10. **end if**
11. $j \leftarrow j + 1$.
12. **end loop**

4.2. Missing Correspondences

Recall that our goal is to compute the statistical dependence between observations in two cameras. We have shown how to do this by computing MI using nonparametric density estimation and integrating out the unknown correspondence using MCMC. In real data, it is common for correspondences to be missing between X and Y. That is, some x_i's may not have corresponding $y_{\pi(i)}$'s. This can occur because some objects may move into a different camera view or because of tracking errors in either camera.

We consider missing data as outliers, and model the distribution of transition times as a mixture of the true and outlier distributions. We can still use the fast m-spacings estimate of entropy by minimizing an upper bound on the mixture entropy. Let $p_1 = p(T|\omega = \text{missing})$, $p_2 = p(T|\omega = \text{present})$ and $\lambda = p(\omega = \text{missing})$. The joint entropy is

$$h(T,\omega) = H(\omega) + h(T|\omega) \tag{20}$$
$$= h(T) + H(\omega|T), \tag{21}$$

where H denotes entropy for discrete variables. Thus,

$$h(T) = H(\omega) + h(T|\omega) - H(\omega|T) \tag{22}$$
$$= H(\lambda) + \lambda h(p_1) + (1-\lambda)h(p_2) - H(\omega|T) \tag{23}$$
$$\leq H(\lambda) + \lambda h(p_1) + (1-\lambda)h(p_2) \tag{24}$$

because $H(\omega|T) \geq 0$. In our experiments we have used a uniform outlier distribution.

In summary, our method relates statistical dependence between two cameras with mutual information between observations at each camera. MI is computed by calculating the entropy of the transition distribution. MCMC is used to integrate over the unknown correspondence between the observations, taking into account missing matches.

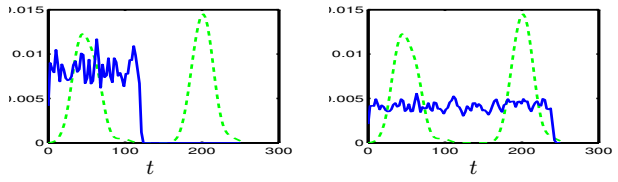


Figure 3: Transition distributions obtained using correlation with different time windows all fail to match the simulated multi-modal distribution (dashed plot). In addition, there is no clear maximum peak indicating statistical dependence.

5. Results

First, we show detailed results for a simulated and real road. In both cases, two cameras are positioned at two non-overlapping portions of the road. Finally, we show results for a simulated and real traffic network of cameras.

5.1. Simulated Road

To study the differences between our approach and previous work we simulated a data set of 100 points from a $\mathrm{Poisson}(0.1)$ departure process. The transition distribution is a mixture of $\mathrm{Gamma}(16.67, 0.33)$ and $\mathrm{Gamma}(266.67, 1.33)$. This generates a dense arrival process and two transition time modes with different means and identical variance. Real objects such as pedestrians and vehicles often exhibit this type of process.

Recall that the correlation method matches all observations within a transition time window. These assumed correspondences are used to estimate the distribution of transition times. Figure 3 shows the transition distributions estimated using the correlation method with various time windows. The number of false correspondences causes the transition distribution to differ greatly from the true distribution. It is difficult to choose a best correlation time window. Also, correlation weakens with increasing distance between the means of the mixture component distributions because it assumes unimodality. Although the transition distribution has low entropy, correlation fails to capture this.

Figure 4 shows our approach on the same data. Although we do not recover the transition distribution exactly, it is much closer in shape than the ones obtained from the correlation method. The estimated MI of 2.47 is close to the true value of 2.12. In general it is difficult to recover the true transition distribution, however our algorithm does find distributions that are qualitatively similar in structure (multi-modal) and quantitatively similar in MI. Departure and arrival times alone may not be able to resolve the ambiguity that can occur by correspondences which shift the modes of the transition distribution.

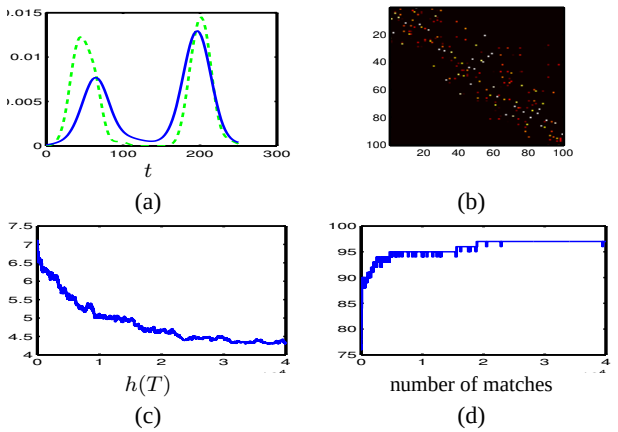


Figure 4: Our method on the simulated road. (a) Estimated transition distribution. (b) Samples from the posterior distribution of correspondences $p(\pi)$ (true correspondence along the diagonal). (c) Entropy of the transition distribution vs. MCMC iteration. (d) Number of correspondences vs. MCMC iteration.

5.2. Single Road

Consider the two views shown in Figure 1 of upstream and downstream portions of the same road. Cars and pedestrians passing through the scene will appear in one view and subsequently in the other. We hand-labeled 100 matches in one day of tracking data obtained from a blob-tracker [16]. The data also contained about 25% unmatched outliers.

Figure 5 shows the transition distributions estimated using the correlation method. As in the case for the simulated data, correlation cannot accurately recover the multi-modal nature of the transitions. The also results in a higher entropy distribution and less statistical dependence. Figure 6 shows the results of our approach on this data. Note how the number of matches changes rapidly initially but eventually converges. Our recovered transition distribution matches the true distribution fairly well. The sharpness of the posterior correspondences point to why we can recover the transition distribution fairly accurately. Figure 7 shows a sample of the correspondences we obtain from our method. In total, 14% of them were in error. This is using temporal information alone. We expect that adding other features such as color may improve the correspondences.

5.3. Simulated Traffic Network

We built a traffic simulator to generate data for a simulated network of cameras at intersections. The simulator was based on a real road network, and took into account real traffic patterns and vehicle dynamics. An example network is shown in Figure 8. We simulated 1000 car trips using shortest paths from start to end node with some noise in the path. Departure and arrival times were recorded.

We computed MI for each pair of cameras. For each

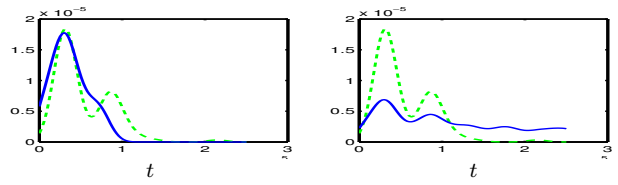


Figure 5: Transition distributions obtained using correlation with different time windows on the road data. The dotted distribution is the true one. The results vary widely for different time windows.

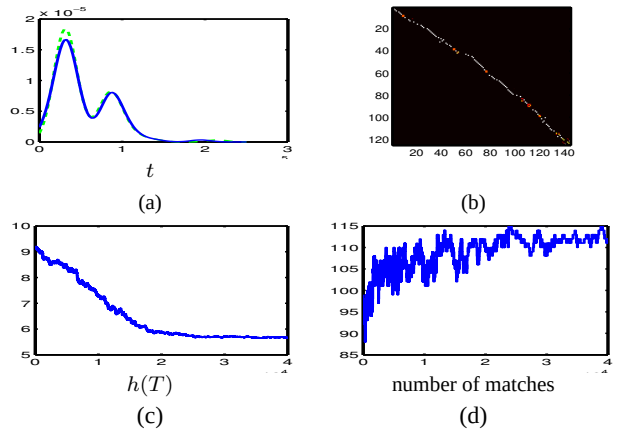


Figure 6: Our method on the road data. (a) Estimated transition distribution. (b) Samples from the posterior distribution of correspondences $p(\pi|O)$ (true correspondence near the diagonal). (c) Entropy of the transition distribution vs. MCMC iteration. (d) Number of correspondences vs. MCMC iteration.

camera, edges with MI values above a fixed percentage of all values were added. For a Markov chain $X \rightarrow Y \rightarrow Z$, the data processing inequality guarantees $I(X;Y) \geq I(X;Z)$ and $I(Z;Y) \geq I(Z;X)$ [1]. Thus directly connected cameras have higher statistical dependence (assuming roughly equal unconditional entropies). Examples of learned graphs based on different percentage thresholds are shown in Figure 8. In our experiments greedy selection closely approximates the correct topology.

5.4. Real Traffic Network

We obtained data from a real traffic network of five cameras, described in the next section. Once again we applied our method as for the simulated traffic network. For this experiment we also added color transformations from one camera to another. This is commonplace because cameras often have different sensor responses. In addition, for wide area surveillance the lighting conditions for vary dramatically across cameras. Color transformations are modeled as flows in RGB space, $c_y = c_x + \Delta c$, for an RGB vector c. Figure 9 shows estimated color flows for a good correspondence and an essentially random one. Note how the

Figure 7: Examples of objects matched by our method. The second from the right has been considered an outlier by the algorithm.

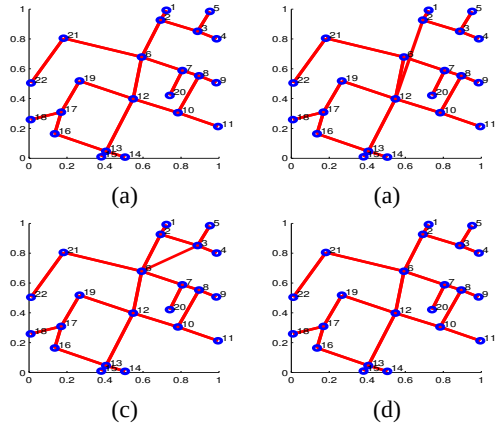
corresponding color flow is essentially a brightening, while the non-corresponding one is less unstructured. The total transformation entropy is the sum of the temporal and color transition entropies. In this case we had greater difficulty inferring the camera transition topology. Many of the primary transitions are recovered as shown in Figure 10. Weaker second order connections also show up. We believe these difficulties are primarily caused by the lack of data. Many of the links between cameras had only about 30 correspondences. In addition, accurate times of departure and arrivals were only available at frame resolution. Adding other visual features may help alleviate these problems.

6 Camera Location Learning

Thus far we have discussed a technique for learning the topology and transition time distributions for a network of cameras. A related problem is that of finding the geometry of the network. Specifically, we are interested in learning the actual geographic coordinates of the fields of view of the cameras, using GPS side information. By doing so, applications can take advantage of semantic information such as the relative locations of key protection assets and camera views. We envision extending the topology learning framework in the future to allow integrating the learning of geometry and topology jointly. In Figures 8 and 11, we would like to be able to learn the locations of the cameras (blue circles and red boxes, respectively).

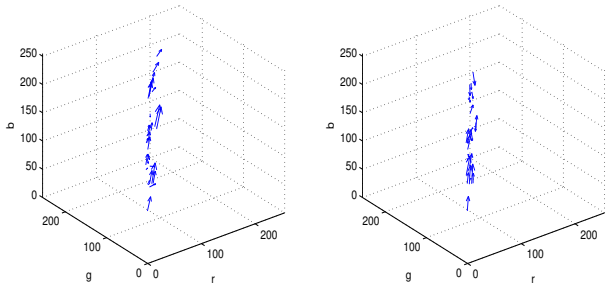
One method is to manually obtain the world coordinates of the camera fields of view at the original installation time. Unfortunately, in many situations this is impractical or insufficient. In some surveillance scenarios, rapid deployment of cameras is necessary to limit danger to those installing them. In a much wider set of long-term surveillance scenarios, cameras tend to get bumped out of calibration, new cameras are periodically added, and old cameras fail.

For our experiment, we assume that we have an installed network of cameras and at least one object moving through the surveillance area that is instrumented with a GPS receiver. Note that we do not have correspondence between the instrumented objects and camera observations, *e.g.* when we see an object pass through a camera, we do not



(a) (a) (c) (d)

Figure 8: (a) The true simulated network of cameras. (b)-(d) Examples of recovered graphs of the simulated traffic network for different MI thresholds. Here the camera locations are assumed known for visualization purposes, but our algorithm is agnostic to this information.



(a) (b)

Figure 9: (a) Corresponding color flow. (b) Non-corresponding color flow.

know to which, if any, instrumented object it corresponds.

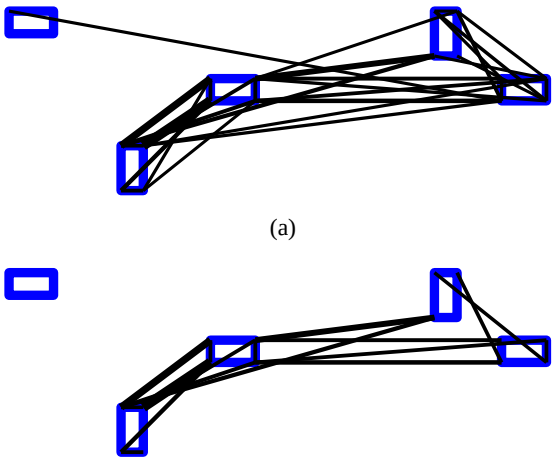
Under this setup, we know the latitude and longitude of each instrumented object at each point in time. We denote this data as the set $\{(x_{vi}, y_{vi}, t_{vi})\}$, where v indexes the vehicle and i indexes time. We may estimate

$$\hat{p}(x,y) \propto \sum_{v=1}^{N_v} \sum_{i=1}^{N_{vi}} \delta(x, x_{vi}) \delta(y, y_{vi}) \tag{25}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta function (see Fig. 11a for $\hat{p}(x,y)$ for the real traffic network we tested).

We separately have access to the recorded in-camera tracking data, $\{(t_{cj})\}$, that reports when vehicles enter and exit each camera's field-of-view, where c indexes the camera number and j indexes the times reports occurred. We seek to correlate the spatio-temporal GPS data with the camera report times. To do so, we wish to estimate $p(x, y|c)$, the probability that a vehicle will be at location (x, y) given that it is seen at the edge of camera c.

Although we cannot quite estimate $p(x, y|c)$ directly, we



(a) (b)

Figure 10: Links inferred for the real traffic network. Link width is proportional to strength of statistical dependence. (a) low MI threshold (b) high MI threshold.

can approximate it with the mixture distribution

$$\tilde{p}(x, y|c) \propto p(x, y|c) + \epsilon p(x, y) \tag{26}$$

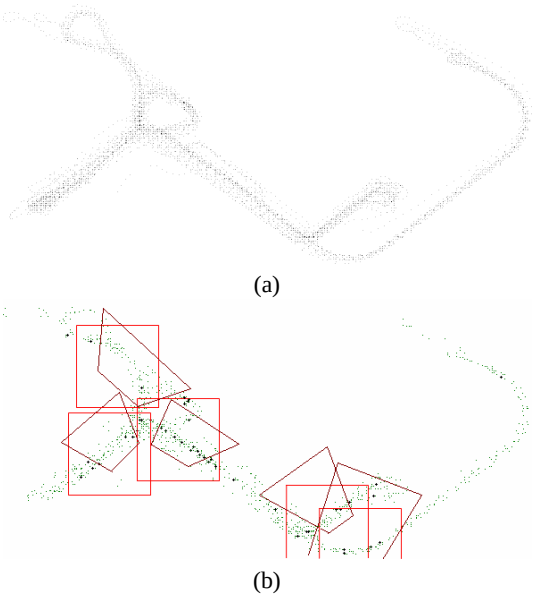
$$\propto \sum_{v=1}^{N_v} \sum_{i=1}^{N_{vi}} \sum_{j=1}^{N_{cj}} \delta(x, x_{vi}) \delta(y, y_{vi}) \delta(t_{cj}, t_{vi}) \tag{27}$$

where ϵ indicates how well $\tilde{p}(x, y|c)$ approximates $p(x, y|c)$. For our current implementation, we assume that $p(x, y|c) \approx \tilde{p}(x, y|c)$ and do not attempt to remove $\epsilon p(x, y)$.

To test this algorithm, we use a dataset of five cameras, five instrumented vehicles following scripted behavior, and approximately unplanned 17 vehicles that passed through the cameras during the data collection period. In Fig. 11b we show $\tilde{p}(x, y|c)$ as green and black (bold) dots, composited from all cameras. We threshold $\tilde{p}(x, y|c)$ and consider high values to be candidate entry/exit locations for the cameras. These are shown in the figure as large dark black dots. For each camera, we find the bounding square of a fixed size that contains the largest number of high values. We only consider the high values generated for the particular camera in question. The red bounding squares are our estimated camera locations. Overlayed as well are dark red trapezoids indicating the manually-drawn approximate ground-truth fields-of-view of the cameras. Note that with this data we are able to correctly identify the camera locations.

7. Discussion

We presented an approach to infer the topology of a camera network by measuring the degree and nature of statistical dependence between observations in different cameras. Unlike previous work, our method explicitly considers the



(a) (b)

Figure 11: **Camera Location Learning:** (a) plot of all GPS coordinates recorded for all vehicles using the real traffic network ($\hat{p}(x, y)$). (b) estimated camera locations, composited together ($\{\tilde{p}(x, y|c)\}$). Black (green) points are GPS coordinates of vehicles judged to (not) correspond to an entry or exit from a camera. Light red squares indicate the location of all automatically-estimated camera locations. Dark red trapezoids indicate the manually-estimated camera locations.

correspondence problem and handles general types of statistical dependence by using mutual information and non-parametric density estimates. It also recovers the transition distribution, not just an average transition time. We demonstrated how our method outperforms previous work on simulated and real data. The computations are approximate but give promising results and is relatively simple to implement.

Future work involves analyzing more than two cameras at a time. Our results have only used time of departure and arrival and color, but our approach is readily adapted to other features. We hope to integrate the learning of metric camera locations into our topology learning framework. We can also take advantage of the following sources of information to further constrain the solution space: (1) motion direction and scene entry/exit information, (2) vehicles in the scene that are not equipped with GPS devices, and (3) transition times between cameras.

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[2] F. Dellaert. Addressing the correspondence problem: A Markov Chain Monte Carlo approach. Technical report, Carnegie Mellon University School of Computer Science, 2000.

[3] R. B. Fisher. Self-organization of randomly placed sensors. In *European Conference on Computer Vision*, 2002.

[4] D. A. Forsyth, J. A. Haddon, and S. Ioffe. The joy of sampling. *International Journal of Computer Vision*, 41(1/2), 2001.

[5] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo In Practice*. Chapman & Hall/CRC, 1996.

[6] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 1970.

[7] T. Huang and S. Russell. Object identification in a Bayesian context. In *International Joint Conference on Artificial Intelligence*, 1997.

[8] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *International Conference on Computer Vision*, 2003.

[9] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Computer Vision and Pattern Recognition*, 1999.

[10] E. G. Learned-Miller and J. W. F. III. ICA using spacings estimates of entropy. *Journal of Machine Learning Research*, 4, 2003.

[11] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Computer Vision and Pattern Recognition*, 2004.

[12] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33, 1962.

[13] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *International Joint Conference on Artificial Intelligence*, 1999.

[14] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Computer Vision and Pattern Recognition*, 2004.

[15] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27, 1948.

[16] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition*, 1999.

[17] O. Vasicek. A test for normality based on sample entropy. *Journal of the Royal Statistical Society, Series B*, 38(1), 1976.

[18] P. Viola and W. M. Wells, IIII. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2), 1997.

[19] D. Wolf. Mutual information as a Bayesian measure of independence. Technical Report 94, Los Alamos Unlimited Release, 1994.