Scene Understanding by Statistical Modeling of Motion Patterns*

Imran Saleemi[1] Lance Hartung[2] Mubarak Shah[1]

[1]Computer Vision Lab
University of Central Florida
{imran,shah}@cs.ucf.edu

[2]Computer Sciences Department
University of Wisconsin-Madison
hartung@cs.wisc.edu

Abstract

We present a novel method for the discovery and statistical representation of motion patterns in a scene observed by a static camera. Related methods involving learning of patterns of activity rely on trajectories obtained from object detection and tracking systems, which are unreliable in complex scenes of crowded motion. We propose a mixture model representation of salient patterns of optical flow, and present an algorithm for learning these patterns from dense optical flow in a hierarchical, unsupervised fashion. Using low level cues of noisy optical flow, K-means is employed to initialize a Gaussian mixture model for temporally segmented clips of video. The components of this mixture are then filtered and instances of motion patterns are computed using a simple motion model, by linking components across space and time. Motion patterns are then initialized and membership of instances in different motion patterns is established by using KL divergence between mixture distributions of pattern instances. Finally, a pixel level representation of motion patterns is proposed by deriving conditional expectation of optical flow. Results of extensive experiments are presented for multiple surveillance sequences containing numerous patterns involving both pedestrian and vehicular traffic.

1. Introduction

The goal of this paper is scene modeling and understanding by unsupervised inference of motion patterns in static camera scenes, which is a key task in visual surveillance. The decreasing cost of surveillance systems has resulted in large amounts of video data, which cannot be adequately handled by human analysts alone. Various attempts have been made to automate commonly encountered surveillance problems of tracking, abnormality detection, activity analysis, and scene understanding [13, 4, 12].

*This research was supported in parts by grants from the NSF REU program, and Harris Corporation.

Figure 1. Examples of surveillance sequences from static cameras used in our experiments. Left to right: MIT, NGSIM, and Hong Kong sequences. The arrows show some desirable patterns of motion.

Scene understanding, in general, may refer to *scene layout* in structured environments (e.g., sidewalks, roads, intersections), *motion patterns* (e.g., vehicles turning, pedestrian crossings), and *scene status* (e.g., traffic light status, congestion). The proposed work is an attempt to model and learn motion patterns from static camera videos without any user intervention. Given that the related literature uses various terms to describe their representation (e.g., behavior, activity, event and variants), it should be mentioned here that the term 'motion pattern' in our work refers to a spatial segment of the image, that has a high degree of local similarity of speed as well as flow direction within the segment and otherwise outside. The patterns may not be disjoint in the image space, and at a semantic level describe the flow of object motion and ideally contain the source as well as the sink of the path described by the pattern. The representation of a pattern however, should not only list the pixels in the pattern, but it is very desirable to also have a statistical model of flow magnitude and direction at each pixel. Models representing scenes and the motion therein are useful in a variety of surveillance related tasks like object tracking and classification, abnormal behavior detection, etc.

Various approaches to scene modeling, diverse as they are in methods and applications, can be broadly categorized based on the observations made from the scene. The most commonly used features for scene modeling are low level motion and appearance features [18, 16, 14, 17]. Examples of such features include sparse or dense optical flows [7], spatiotemporal gradients [9], and object trajectories obtained after detection and tracking [13, 1, 15, 4, 12].

Scene modeling and activity perception approaches based on the traditional pipeline of object detection (based

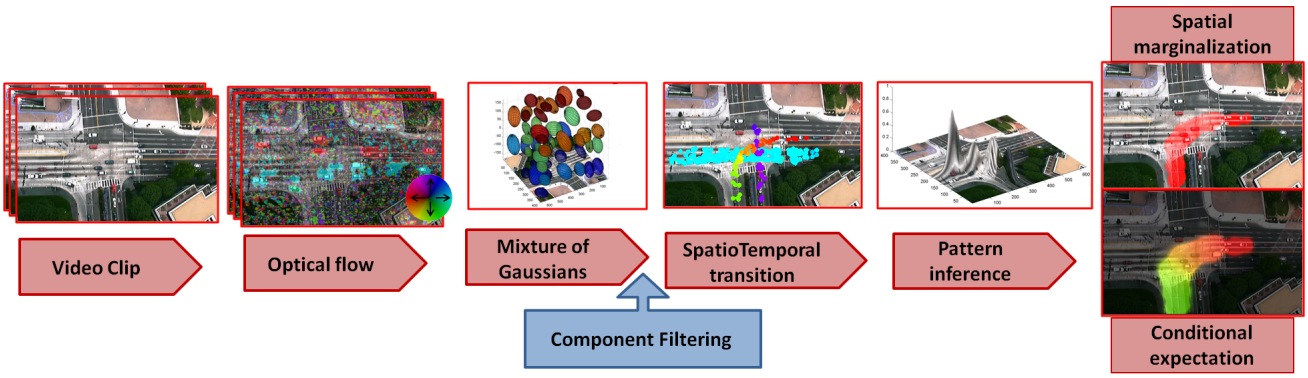

Figure 2. Process flow of our approach: Grouping of frames into video clips, optical flow computation, Gaussian mixture model learning by K-means, filtering of noisy Gaussian components, inter-component spatiotemporal transition computation for instance learning, pattern inference using KL-divergence, pattern representation as spatial marginal density, and computation of conditional expected value of optical flow given pixel location.

on appearance [2] or motion [13]), and subsequent tracking of those detections [1, 5], are well suited to surveillance scenarios involving far-field settings, where computation of complicated features, such as gestures, appearance or local motion may be infeasible, but sufficient number of pixels on object ensure motion based detection. Such methods are applicable to structured scenes containing clearly discernable distinct paths, like roads and sidewalks, and well defined entry and exit points. Proposed approaches to scene understanding employing trajectory features involve clustering or classification of whole trajectories using proximity in the space of different distance metrics [13, 15, 4, 8, 3, 6]. Recently proposed methods learn dense pixel to pixel transition distributions using trajectories obtained after tracking [12] and attempt abnormality detection, improved tracking, and even motion based foreground background segmentation. The reliance of all the above mentioned approaches on good quality object trajectories is however, a significant drawback.

The second kind of approaches for scene understanding and activity analysis use low level motion or appearance feature vectors directly, instead of trajectories. Such features include multi-resolution histograms [18], spatiotemporal gradients [9], and appearance descriptors and spatiotemporal volumes [10], etc. These approaches can perform action recognition and are useful for detection and separation of co-occurring activities, but are usually supervised, where the training stage involves manual isolation of activities or video clips into distinct activities.

Our proposed approach circumvents both problems of object based tracking in crowded scenes, and supervised learning of patterns or activities in the training stage, by using only pixel based local motion features. A few methods have recently been proposed that employ similar low level features for motion pattern inference [14, 9, 17]. Wang et al [14] obtain sparse optical flow [7] for pixels with frame difference above a threshold which is then used as the low level observations. Yang et al [17] also employ dense optical flow, while Kratz and Nishino [9] use spatiotemporal gradients as the most basic features. All these methods however share the severe drawbacks of spatiotemporal quantization of videos, and co-occurrence statistics based inference. While temporal division of long video sequences into tractable smaller length video clips is useful without loss of information, spatial division of video frames into cuboids (30×30 in [9], and 10×10 in [14, 17]) results in loss of resolution and therefore coarse estimates of motion at the super-pixel level, as opposed to pixel or sub-pixel levels. Since high density crowd scenes may not have regions of uniform motion, especially when observed over a significant length of time, and owing to the small number of pixels per object, this spatial resolution, which is usually already limited in surveillance scenarios, becomes extremely important. The quantization of optical flow direction into large intervals [14, 17] also poses a significant limitation on the potential applications by introducing abrupt boundary jumps and coarse pattern direction estimation. Furthermore, employment of co-occurrence statistics [17], while useful at higher levels of model abstraction, results in the fact that co-occurring patterns of motion, even when distant in image and motion feature space, cannot be discerned from one another. On the other hand, explicit high level temporal models of atomic patterns with non-quantized representations can easily be employed to deduce co-occurrence.

We present a new method for inference of motion patterns which overcomes the drawbacks and limitations of the above mentioned techniques, while employing simple yet powerful statistical modeling and learning methodologies. The process flow of our method is outlined in Fig. 2. In particular, the novel contributions of our approach include, (1) introduction of a statistical model of raw unquantized optical flow for motion patterns representation, (2) a hierarchical problem-specific learning method, (3) use of a co-occurrence free measure of spatiotemporal proximity and flow similarity between features, and (4) statistical inference of motion patterns by computation of conditional op-

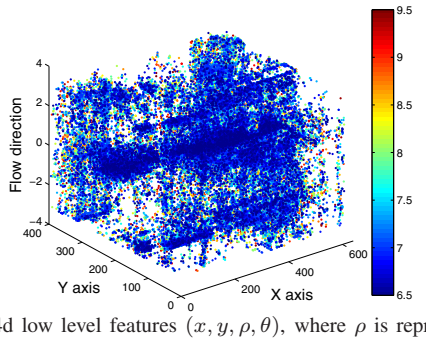

Figure 3. 4d low level features (x, y, ρ, θ), where ρ is represented by colors as per the legend. Note that $\theta \in [-\pi, \pi]$.

tical flow expectation.

2. Proposed Method

Our proposed method begins with introduction of a probabilistic model of motion patterns based on optical flow, such that although noise and clutter may still not be clearly separable, the statistical properties of our model of dense flow will help classify it into patterns when observed for extended periods of time. We first compute dense optical flow using the method of [7]. We can then define, $\mathbf{X} = (x, y, \rho, \theta)$, as a random variable representing optical flow at a particular pixel, where (x, y) is the location of the pixel on the image lattice, and (ρ, θ) is the magnitude and direction of the optical flow vector (u, v) such that, $\rho = \sqrt{u^2 + v^2}$ and $\theta = tan^{-1}\left(\frac{v}{u}\right)$. A number of possible values for the variable $\mathbf{X}$ are shown in Fig. 3, as observed in a small video clip. The goal of learning the model is to evaluate the probability $p_i(\mathbf{X} = \mathrm{x})$, i.e., estimate the likelihood of a pixel and optical flow pair x, of belonging to the motion pattern i.

Since the motion patterns need to be learned over extended periods of time, and the optical flow is dense, only a parametric approximation of the actual distribution is tractable. We employ a mixture of Gaussian model to this end. Furthermore, since estimation of mixture parameters is computationally intensive, we propose a hierarchical learning technique that allows us to bypass parameter optimization without compromising the quality of distribution. The hierarchical learning is performed in a bottom up fashion, starting from pixel level optical flow vectors, to a large number of distributions, which then represent mixtures of distributions. The mixtures then form the eventual representation of motion patterns.

It can be noticed that a single pattern of motion, as well as an 'instance' of that pattern (e.g., an object following the pattern), can be visualized as a worm in the 5-space, where time (frame number) is the fifth dimension, just like the trajectory of an object is a curve in the 3-space, (x, y, t). Obviously the description of a pattern itself is independent of the notion of time (although the *sequence* of inter-pattern

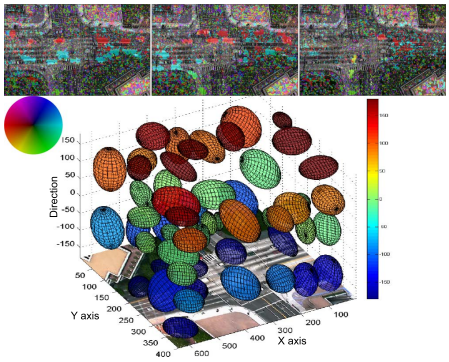

Figure 4. Top: 3 examples of optical flow frames in a video clip, where colors show flow direction as per the circle on the left, and magnitude is represented by image brightness. Bottom: A few Gaussian components representing the marginalized distribution $\int p(\mathbf{X})d\rho$ for the video clip, shown as error ellipses with surfaces at 1σ. The colors of ellipses also indicate the mean direction of optical flow, i.e. μ_θ as per the color bar shown on the right.

occurrences in general would be governed by time, i.e., non-random). However, multiple instances of the same pattern can be significantly spaced out in time, which is very likely to happen since most patterns by definition are recurrent. Therefore, in the learning phase, time is taken into account when learning instances of patterns, but ignored when learning the actual patterns. Since the smallest unit of time, a frame of optical flow, has too few observations (values for the random variable $\mathbf{X}$ to take) to contribute towards a meaningful group of mixture components, we therefore quantize time into disjoint segments of k frames each, referred to as 'video clips' where, in our experiments, k is typically set to the number of frames in 1 second of video.

2.1. Mixture Model

We begin by marginalizing out time in each video clip, and filtering out optical flow observations with unusually small or large magnitudes. The remaining observations can now be considered as data points in the 4d space (x, y, ρ, θ), an example of which, is plotted in Fig. 3. Our goal now is to learn this 4d distribution modeled as a Mixture of Gaussian components. The theoretically straightforward way for this is random or approximate initialization of a chosen number of components followed by a parameter optimization algorithm like EM. However, as we will later show, a careful initialization suffices for the purpose of our algorithm and the optimization process can be skipped (significantly improving computation efficiency as the number of data points in each video clip are $\sim 10^5$). We therefore estimate the parameters of the distribution by treating the observations as data points that need to be clustered into N clusters, where the centroid and error ellipse of each cluster serve as the parameters, mean μ, and standard deviation σ, in each dimension for each Gaussian component, which along with the orientation of the error ellipse determines the full covariance matrix, Σ of the component.

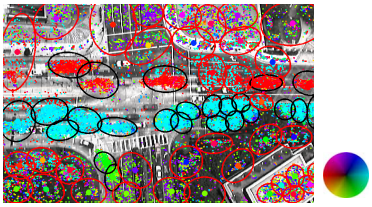

Figure 5. Component Filtering: Two dimensional marginalized Gaussian mixture components, $\int\int p(X)d\rho d\theta$, represented by component error ellipses at 1.5σ. Data points are shown as colored dots, where colors correspond to optical flow directions as per the circle shown as legend. Components detected as noise or clutter are shown as red ellipses, while components with low directional variance are depicted as black ellipses. Mean direction is shown as solid dot at each component centroid. Notice the variation in colors in noisy components.

We employ K-means clustering to estimate the parameters of these N components or clusters. Initialization is performed using multiple iterations of K-means on randomly sampled subsets of the data points, and the distance metric is Euclidean, modified to cater for boundaries of direction interval $[-\pi, \pi]$. We can then write, $\mathbf{X} \sim \sum_{i=1}^{N} w_i \mathcal{N}(\mathbf{X}|\mu_i, \Sigma_i)$, where, w_i, the weight of each component is the percentage of data points in the i^{th} cluster, and $\mathcal{N}$ represents Gaussian distribution.

As mentioned before one of the main goals of the proposed method of motion pattern learning is to be able to avoid object detection and tracking, which can be problematic especially in cases of crowd scenarios and dense traffic flow and clutter. We observe that by filtering non-moving and erroneous optical flow observations, the data points in the 4d space (x, y, ρ, θ) are segmented into spatially disjoint clusters generally corresponding to moving objects. Although raw optical flow is noisy, observations of true motion occur in dense, low variance clusters of consistent optical flow, which then become components of the mixture model. Since our objective is to find many dense clusters with small variance especially in the direction of optical flow, the number of components need only be large enough. Furthermore, optimization of parameter fitting is not required. An example of K-means clustering as component initialization can be seen in Fig. 4.

2.1.1 Gaussian Component Quality Assessment

Despite filtering of optical flow with nominal or unusually large magnitude ρ, a non-negligible number of remaining observations, and therefore Gaussian components will comprise optical flow from noise and background areas. Such components have x and y standard deviations that are similar to other components, but they have high variance in ρ and θ dimensions as shown in Fig. 5. The components with unusually high variances especially in flow direction, θ, can therefore be discarded. It can be seen in Fig. 5 that such components, shown as red ellipses, have observations with highly varying colors that depict flow direction. While discarding of these components can be construed as loss of

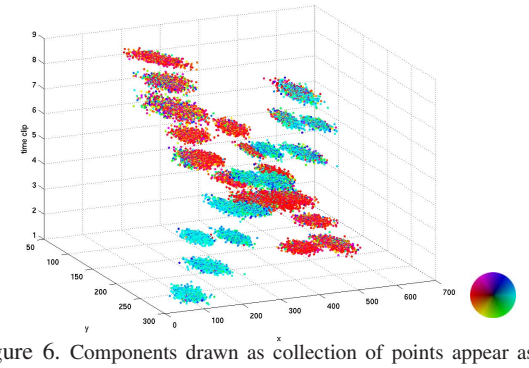

Figure 6. Components drawn as collection of points appear as worms in the spatiotemporal volume (x, y, t), where t is quantized into video clips. Colors represent the optical flow direction as per the legend, while magnitude is not shown. One instance each is shown for two patterns; eastward (cyan), and westward (red) traffic.

information, it should be noted that given the small value of k, the GMM of a single time clip represents instances of atomic events and even if a few correct components are discarded, these atomic examples are repeated periodically multiple times in a video sequence, and therefore do not have an adverse effect on learning of actual patterns. Fig. 5 shows an example clip where such components have been filtered out. The filtering is based on directional variance of a component being larger than other components (relative), or than a fixed threshold (absolute), e.g., mean variance.

Although learning the proposed model using a non-parametric distribution is intractable as explained earlier, it should be noticed that components in the Gaussian mixture behave as representative sampled data points of the underlying data, and adequately retain the multimodal structure of the true distribution.

2.2. Inter-Component Spatio-Temporal Transition

We now have the representation of a single video clip in the form of N Gaussian mixture components, our goal is to do the same for motion patterns using multiple video clips. We observe that since motion pattern instances should appear as continuous worms in the spatio-temporal volume of optical flow, and because each worm has been divided into components by mixture learning (as shown in Fig. 6), the clusters belonging to the same pattern form a representation of a pattern 'instance'. We therefore, propose a method of linking components in temporally adjacent (or proximal) video clips to form multiple mixtures corresponding to instances of motion patterns.

Since the GMM components capture the local shape of a pattern instance, by linking components through time, the full structure of the pattern can be recovered. We treat components from all video clips as nodes in an undirected graph and place an edge between two nodes according to the following conditions: (1) the components belong to proximal video clips, and (2) one component is 'reachable' from the

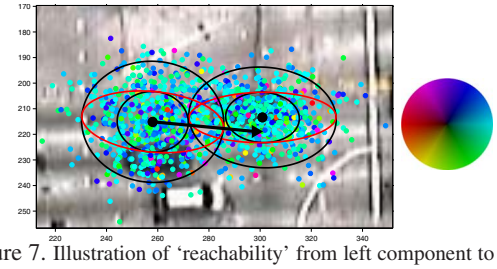

Figure 7. Illustration of 'reachability' from left component to right component. Two sets of black concentric ellipses show 1σ and 2σ error ellipses for each component, with mean location shown as large black dots in the center. Cyan colored dots show data points in each component, and represent eastward motion. The red ellipses depict hypothetical transformed error ellipses as described in text. The black line starting from left component's centroid shows the prediction $(\hat{x}_i, \hat{y}_i)$, represented by the arrow.

other. The term 'reachability' can be defined in terms of transition probabilities of a random walk, where these transitions reflect an underlying motion model. A graphical illustration of this process can be seen in Fig. 7. Given components i and j from video clips t and $t+\xi$ respectively, where ξ is a small positive integer, the probability of transition of a feature from component i to j can be computed as the probability that a *predicted* data point $\hat{\mathrm{x}}_i$ belongs to component j, and can be written as,

$$p_j(\mathbf{X} = \hat{\mathrm{x}}_i) = \frac{1}{4\pi^2\sqrt{|\Sigma_j|}} exp\left[-\frac{1}{2}\left(\hat{\mathrm{x}}_i - \mu_j\right)^\top \Sigma_j^{-1}\left(\hat{\mathrm{x}}_i - \mu_j\right)\right], \quad (1)$$

and $\hat{\mathrm{x}}_i = (\hat{x}_i, \hat{y}_i, \rho_i, \theta_i)$ is the transition prediction for component i, where, $\hat{x}_i = x_i + \xi k \rho_i \cos\theta_i$, and $\hat{y}_i = y_i + \xi k \rho_i \sin\theta_i$.

The transition prediction is set up such that the underlying motion model follows the constant velocity model, in that smoothness of motion is exhibited in terms of both magnitude and direction of velocity, whereas the factor ξk, the number of frames between the video clips ensures that the prediction is meaningful for evaluation using component j. The weight of the edge connecting nodes i and j is computed using both forward (Eq. 1) and backward likelihoods, where the backward likelihood is computed using the opposite direction $(\theta_j + \pi)$. Since these two events are independent, the edge weight is defined as their product.

The edge weights are subsequently thresholded by using Mahalanobis distance between the destination node's distribution and the source node's prediction. However, for the purpose of computing the edge weight, the destination node's covariance, Σ_j is transformed so that while the error ellipse's orientation in 4d remains unchanged, the spatial variance of the component (variance in x and y dimensions), reflects more variation in the direction parallel to its mean optical flow direction, compared to the direction orthogonal to mean flow. In other words, component j is reachable from component i if its prediction falls within a

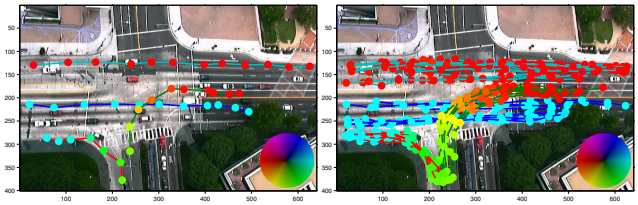

Figure 8. Examples of pattern instances. Left: one instance each of four patterns, westward, eastward, left and right turns. Colored circle locations represent Gaussian component's X and Y means, while their colors represent mean optical flow direction as per the circle in the bottom right. Lines connecting the components represent membership in an instance. Right: Multiple instances for each pattern shown in the left image.

transformed error ellipse of cluster j, where the transformed ellipse reflects standard deviation 2σ in direction of mean flow and 1σ in the perpendicular direction. This reachability test is illustrated in Fig. 7, where the transformed covariance is depicted by red ellipses. When all edges have been placed between appropriate nodes, the connected components of the graph represent distinct motion pattern instances that occurred throughout the video. Some examples of such pattern instances are shown in Fig. 8.

2.3. Motion Patterns Inference

We now have numerous instances of each motion pattern and the final goal is to merge all instances of each distinct pattern into a single mixture of Gaussian components, which can be achieved by a comparison between all pairs of instance probability distributions (represented by GMMs). We use the Kullback-Leibler (KL) divergence to compare these distributions. Note that the previous step of computing pattern instance distributions incorporated a time constraint, such that each instance is bounded by, and is continuous across a temporal segment. However, it is now reasonable to compare instances from different time periods in the video. In other words, if we consider two mixtures of temporally linked Gaussian components, the KL divergence between them will be low if they represent the same action (eg. a car moving eastward) because of similarity in location (x, y), as well as motion (ρ, θ).

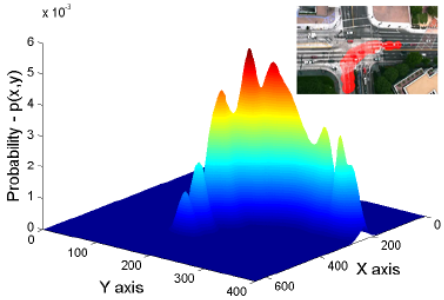

Figure 9. Probability density for a single motion pattern, left turn. Assuming this is the m^{th} pattern in the set of all patterns, the figure shows the marginal density surface, $\int\int p_m(x, y, \rho, \theta) d\rho d\theta$. Inset: The same density thresholded and overlayed on the scene image, shows the extent of the motion pattern.

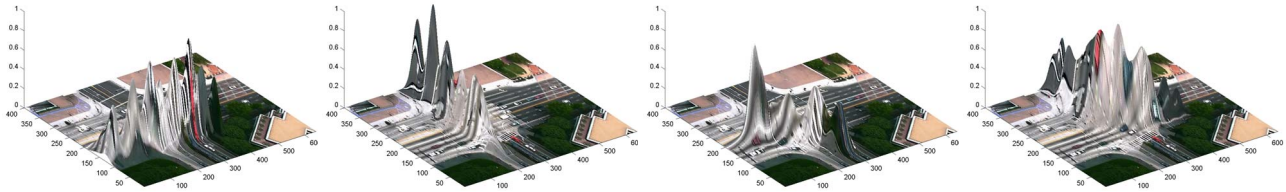

Figure 10. Probability surfaces of four motion patterns, textured by an image of the scene showing, (L-R) eastbound motion, southbound motion, right turning traffic, and left turn.

The KL divergence between two Gaussian mixtures is only computable through approximation, for which we use Monte Carlo sampling of instance distributions. We employ Eq. 2 where p_f and p_g are probability density functions representing two pattern instances f and g, that we wish to compare. We draw n samples $\{\mathrm{x}_i\}_{i=1}^{n}$ from the first distribution p_f. If the samples also have a high probability of occurring in distribution p_g, then the divergence is low, and we consider the instance g to be a subset of the same pattern as f. We can therefore write for a two instance divergence,

$$D(f \parallel g) \approx \frac{1}{n}\sum_{i=1}^{n} \log \frac{p_f(\mathrm{x}_i)}{p_g(\mathrm{x}_i)}. \tag{2}$$

The divergence between all possible instance pairs is computed and assembled into a non-symmetric square matrix which can be considered as edges of a fully connected graph, where the edges can be binarized by choosing different thresholds. A graph connected component analysis of the matrix then gives multiple final motion patterns. The automatic or manual choice of various binarization levels represent a multiscale analysis of the scene, such that a high level corresponds to few, large patterns, and vice versa. The components of individual instances are merged into the connected component (the motion pattern) that they belong to. Patterns with a single or very few instances can be discarded.

The result is a number of self-similar groups of motion instances that have occurred numerous times throughout the video. Each such group then represents an individual Gaussian mixture. As shown in Fig. 9, a marginalized probability map of each of these new GMMs indicates the probability of a pixel belonging to that motion pattern. By thresholding this probability map, it is possible to give a binary label to each pixel in the image, indicating membership in the pattern, as shown in Fig. 9(inset).

2.4. Conditional Expectation of Optical Flow

The proposed representation of a motion pattern is richer than giving a binary label to each pixel because it also contains per pixel motion information. The majority of applications in scene understanding and surveillance require that probability is bound to pixels, i.e., the statistical distribution is computable conditional on spatial location. The proposed probability distribution of a motion pattern given a pixel is easily computable by marginalization over the optical flow dimensions, ρ and θ, that is $\int\int p(x, y, \rho, \theta) d\rho d\theta$ as can be seen in figures 9 and 10. However, we are not only interested in the probability as shown in Fig. 9, but also in the distribution of different optical flow values given a pixel.

Furthermore, the most useful representation of a pattern is not the probability density function of optical flow given a pixel but the value of optical flow that is most expected for that pattern at a given location. The estimation of such a value involves the computation of conditional expected value of optical flow magnitude and direction given a pixel location. Assuming conditional independence given a pixel, we compute the conditional expectations for magnitude ρ and direction θ separately. The expectation for direction given a pixel location, is a two dimensional function with (possibly real valued) domain equal to the size of the image, and range in the interval $[-\pi, \pi]$, and for a motion pattern comprised of M Gaussian mixture components, indexed by $i \in \{1, \ldots, M\}$, this function can be computed as follows:

$$\begin{aligned}
\mathbb{E}[\Theta|\mathbf{X}, \mathbf{Y}] &= \int_{-\infty}^{\infty} \theta p(\theta|x, y) d\theta = \int_{-\infty}^{\infty} \frac{\theta p(x, y, \theta)}{p(x, y)} d\theta \\
&= \frac{1}{p(x, y)} \int_{-\infty}^{\infty} \theta \left[\sum_{i=1}^{M} w_i p_i(x, y, \theta)\right] d\theta \\
&= \sum_{i=1}^{M} w_i \frac{\int_{-\infty}^{\infty} \theta p_i(x, y, \theta) d\theta}{p_i(x, y)} = \sum_{i=1}^{M} w_i \mathbb{E}_i[\Theta|\mathbf{X}, \mathbf{Y}] \qquad (3) \\
&= \tan^{-1}\left\{\frac{\sum_{i=1}^{M} w_i \sin\left(\mathbb{E}_i[\Theta|\mathbf{X}, \mathbf{Y}]\right)}{\sum_{i=1}^{M} w_i \cos\left(\mathbb{E}_i[\Theta|\mathbf{X}, \mathbf{Y}]\right)}\right\}, \qquad (4)
\end{aligned}$$

where $\mathbb{E}_i[\Theta|\mathbf{X}, \mathbf{Y}]$ is the expected direction at a pixel indicated by (x, y) as per the Gaussian distribution of the i^{th} component of the pattern. Eq. 3 shows that the conditional expectation of a weighted mixture distribution is the weighted mean of individual conditional expectations, whereas Eq. 4 is specific to direction expectation, so as to avoid problems at phase boundaries.

Fixing x and y by choosing a particular pixel, this distribution is a one dimensional density represented by values along a vertical line in the three dimensional space (x, y, θ). It is known that for any multivariate Gaussian distribution of dimension k, any sub-vector of dimension $l < k$, itself represents a multivariate Gaussian distribution, and it can be shown that the distribution of any dimension of the multivariate Gaussian density, given all the other dimensions is a one dimensional Gaussian probability density.

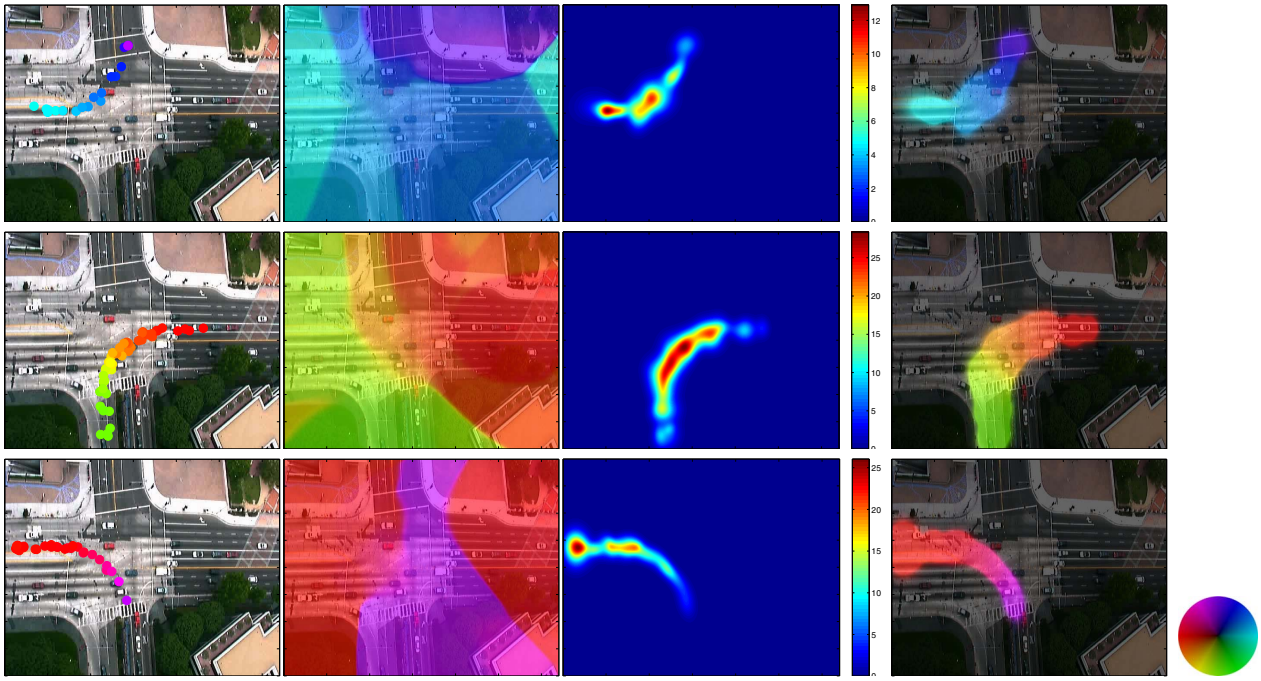

Figure 11. Three example patterns, with each row showing: (L-R) components in the pattern, expected orientation per pixel, expected magnitude per pixel, and combined expected optical flow per pixel, with magnitude indicated by brightness.

Therefore, the pdf of $p_i(\theta|x,y)$ is Normally distributed and its mean is given by $\mathbb{E}_i[\Theta|\mathbf{X},\mathbf{Y}] = \mu_\theta + \begin{bmatrix} \sigma_{\theta X}\sigma_{\theta Y} \end{bmatrix} \bullet \begin{bmatrix} \sigma_{XX}\sigma_{XY} \\ \sigma_{YX}\sigma_{YY} \end{bmatrix}^{-1} \bullet \begin{bmatrix} x-\mu_X \\ y-\mu_Y \end{bmatrix}$.

The conditional expectation of magnitude ρ can be computed in a similar fashion using Eq. 3. The conditional expected optical flow provides the dominant magnitude and direction of the motion pattern at the pixel level, instead of quantized or interpolated estimates of direction at super-pixel levels (quantized image space) as in [14, 17]. Since GMM pdf is a continuous function, the proposed representation can be used to computed probability or expectation with sub-pixel accuracy. Furthermore, given the range of expected optical flow magnitude over the entire image plane, the need for thresholding of probability is avoided as can be seen in Fig. 11. It is worth mentioning here that the boundary or segmentation of the patterns shown in Fig. 11 are not defined by any thresholding of the probability, but the rather sharp contours are actually a manifestation of the steep drop in expected flow magnitude values, reducing the brightness of the flow image.

3. Experimental Results

We tested the proposed model and learning method on three data sets. For the first, NGSIM data set [11], we used a 15 minute video of a segment of road that includes an intersection as can be seen in Fig. 1. This is a challenging video for scene understanding because, for the region in the center of the intersection, there is no single dominant pattern. Fig. 12 shows the expected optical flow of some of the patterns discovered, by overlaying them on an image of the scene field of view. Each pixel in the image shows the expected direction with a different color, and the expected flow magnitude with varying brightness. Fig. 12(a-d) are examples of, (a) southbound, (b) northbound, (c) east to south right turn, and (d) westbound motion patterns.

The second data set used in our experiments is the MIT data set [14]. The patterns discovered in this sequence are shown in Fig. 12(e-h), where the patterns shown are, (e) pedestrians entering building, (f) northbound traffic, (g) westward pedestrian traffic on crosswalk, and (h) south to east traffic left turn. The third data set we tested our algorithm on is also very challenging, depicting dense crowds of pedestrians walking in opposite directions through the same spatial regions. As can be seen in Fig. 12(i-l), the proposed method is able to detect multiple semantically meaningful patterns of motion in a completely unsupervised manner. This sequence is much more challenging than those used in [14, 17, 9], because of crowded traffic, and multiple co-occurring patterns in the same region. This sequence is only 250 frames which is a very small number for learning patterns. The results of this sequence also employ the multi-scale analysis of motion patterns by varying thresholds of the KL divergence graph edges. There are only three main patterns in the small sequence, i.e., pedestrians on crosswalk walking in opposite directions simultaneously, and the vehicles on the top right coming to a stop. However, a low threshold of KL divergences yields many small patterns, one of which is shown in Fig. 12(i), while a higher one re-

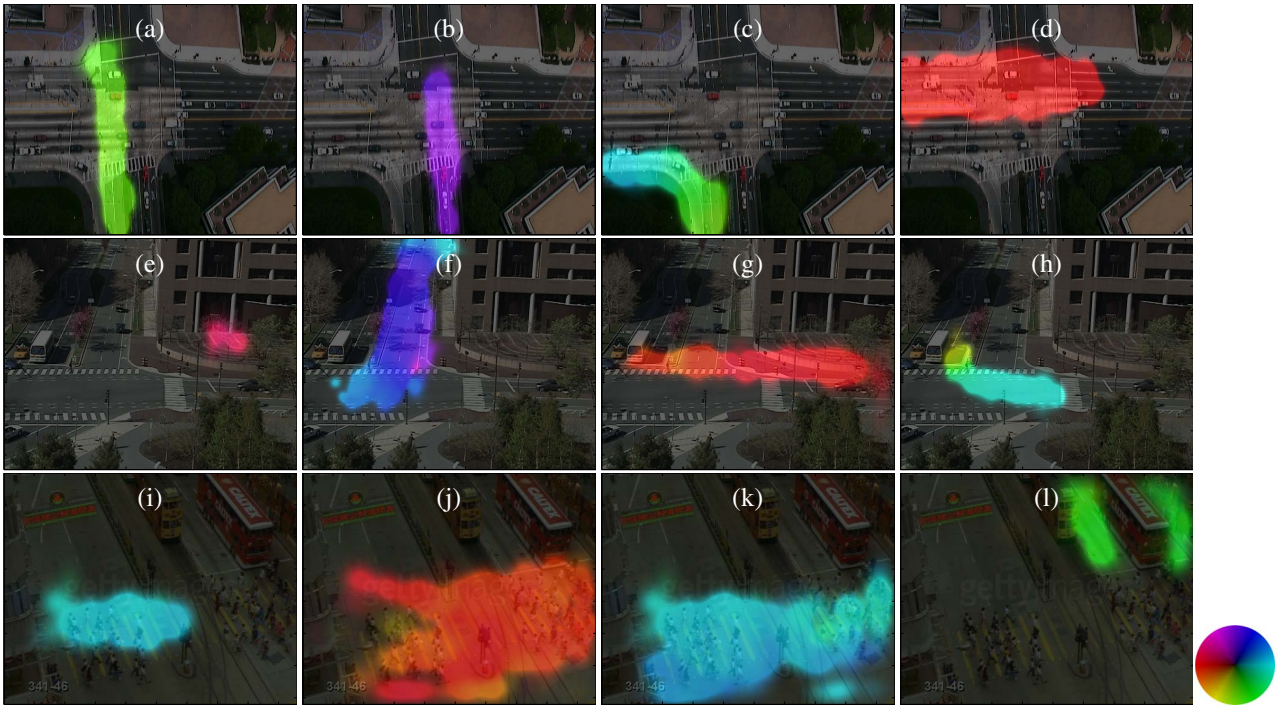

Figure 12. Some of the detected motion patterns, represented by expected optical flows, are shown for, (a-d) NGSIM dataset, (e-h) MIT dataset, and (i-l) Hong Kong sequence. The colors represent expected flow direction as per the circle on the bottom right, while the brightness in different regions indicate the expected magnitude of flow. See text for details about patterns' description.

sults in only three (j-l), that represent motion at a higher semantic level. This aspect of our approach is akin to the multiscale analysis of Yang et al [17]. The proposed method is readily usable in a variety of surveillance applications, for example anomaly detection, and use as prior motion model for tracking.

4. Conclusion

In conclusion, we have presented a novel scene understanding framework, and introduced a new representation and learning approach for motion pattern discovery in static scenes, where an optical flow based mixture model is representative of salient patterns of traffic, and is learned without parameter optimization. Our representation avoids any quantization and loss of information in the feature space, and we have presented results of motion patterns discovery in challenging scenarios.

References

[1] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *PAMI*, 22(8):844–851, 2000.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[3] Z. Fu, W. Hu, and T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *ICIP*, 2005.

[4] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *PAMI*, 28(9):1450–1464, 2006.

[5] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *BMVC*, 1995.

[6] I. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. In *ICPR*, 2004.

[7] T. Kanade and B. Lucas. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[8] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining applications. In *ACM Int. Conf. on Knowledge Discovery and Data Mining*, 2000.

[9] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *CVPR*, 2009.

[10] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[11] Next Generation Simulation (NGSIM) dataset. Available at http://www.ngsim.fhwa.dot.gov/.

[12] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *PAMI*, 31(8):1472–1485, 2009.

[13] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *PAMI*, 22(8):747–757, 2000.

[14] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception by hierarchical bayesian models. In *CVPR*, 2007.

[15] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. In *ECCV*, 2006.

[16] T. Xiang and S. Gong. Video behaviour profiling and abnormality detection without manual labelling. In *ICCV*, 2005.

[17] Y. Yang, J. Liu, and M. Shah. Video scene understanding using multi-scale analysis. In *ICCV*, 2009.

[18] H. Zhong et al, J. Shi, and M. Visontai. Detecting unusual activity in video. In *CVPR*, 2004.