Similarity Invariant Classification of Events by KL Divergence Minimization

Salman Khokhar, Imran Saleemi and Mubarak Shah
University of Central Florida
{skhokhar, imran, shah}@eecs.ucf.edu

Abstract

This paper proposes a novel method for recognition and classification of events represented by Mixture distributions of location and flow. The main idea is to classify observed events into semantically meaningful groups even when motion is observed from distinct viewpoints. Events in the proposed framework are modeled as motion patterns, which are represented by mixtures of multivariate Gaussians, and are obtained by hierarchical clustering of optical flow in the four dimensional space (x, y, u, v). Such motion patterns observed from varying viewpoints, and in distinct locations or datasets, can be compared using different families of divergences between statistical distributions, given that a transformation between the views is known. One of the major contributions of this paper is to compare and match two motion pattern mixture distributions by estimating the $similarity$ transformation between them, that minimizes their Kullback−Leibler (KL) divergence. The KL divergence between Gaussian mixtures is approximated by Monte Carlo sampling, and the minimization is accomplished by employing an iterative nonlinear least squares estimation method, which bears close resemblance to the Iterative Closest Point (ICP) algorithm. We present a robust framework for matching of high-dimensional, sampled point sets representing statistical distributions, by defining similarity measures between them, for global energy minimization. The proposed approach is tested for classification of events observed across several datasets, captured from both static and moving cameras, involving real world pedestrian as well as vehicular motion. Encouraging results are obtained which demonstrate the feasibility and validity of the proposed approach.

1. Introduction and Related Work

Action, event, and behavior recognition is one of the important high level tasks in classical computer vision. In particular, the ability to recognize and classify events across distinct views, is crucial to solving real world problems in practical scenarios. The proposed framework is a step forward in this regard and describes an intuitive approach to allow comparison between events captured across datasets, viewpoints, and locations.

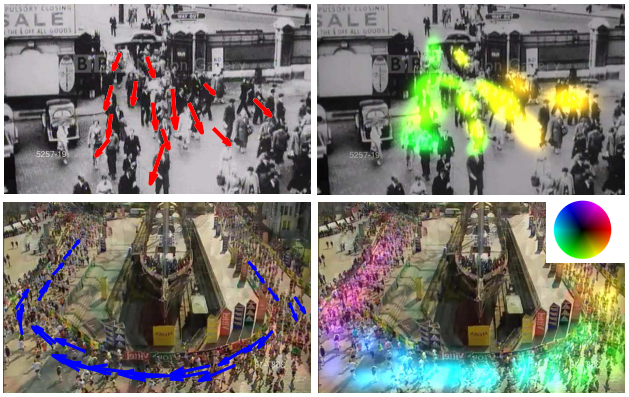


Figure 1. Two examples showing events, (top) 'divergence', and (bottom) 'right U-turn' respectively. The left column shows a few flow points as arrows indicating direction of motion. Right column displays the learned conditional expectation of flow, where the flow orientation is shown by color, and magnitude by brightness, as per the color wheel on the right.

Even though the area of action analysis has been explored in depth [13], and very reasonable performance results have been achieved for $human$ action recognition, numerous other kinds of behaviors and events are observable in videos that do not correspond to similar, articulated, human actions. Such behaviors include global, multi-agent, collective events, examples of which include both vehicular and pedestrian events, such as traffic turns, 'stop and start' events, queuing or lane formation, and convergence and divergence, etc. Techniques developed in human action recognition, are not well suited to the modeling of these high-level, global events (henceforth, the term 'events' corresponds to such activities and behaviors, as opposed to articulated human 'actions').

Since global events comprising human or vehicle agents are inherently different from articulated actions, the question of how they should be modeled is an important one. One of the simpler answers is the clustering and modeling of trajectories of objects involved in the event, where the

tracks are obtained using the conventional object detection, and tracking pipeline. This technique is exploited by Hu *et al*. in [4]. Most of the interesting events however, are observed in scenarios of dense, crowded motion, which exhibit rich and diverse behaviors. Such dense crowd videos impose a severe limitation on the pre-processing step in trajectory modeling based techniques, i.e., the ability to track individual objects.

In recent years, event and behavior modeling and understanding is increasingly being viewed as the problem of 'motion patterns' estimation [12, 14, 3, 6, 11, 7, 9], where motion patterns can be described as contiguous regions of the scene that contain a similar, smoothly varying motion flow field. These methods tend to move away from the object-level representation paradigm, and instead rely on raw, noisy, low-level motion features. Although they employ different low-level features (e.g., spatiotemporal gradients [6], and optical flow [12]), as well as diverse pattern recognition frameworks (e.g., Gaussian mixtures [6, 11], and topic models [12, 3]), the motion patterns in general are ideally suited to discriminatively represent a wide range of observable events, including but not limited to, road traffic patterns like vehicular turns, circular motion, acceleration, etc., and pedestrian behaviors such as entry and exit, and convergence and divergence, etc., behaviors that do not lend themselves easily to modeling using existing action representation frameworks. Two examples of such events are shown in figure 1.

Although location of observed flow is an important cue and constraint in grouping and analysis of spatiotemporal flow data, the same constraint transforms into the limitation of extremely view-dependent representations, which in turn makes comparison and recognition much harder. Due to the presence of spatial dimension in these representations, the above mentioned methods are unable to perform generalized event *recognition* or *classification*, for behaviors captured from varying view points, for videos of distinct datasets, or moving camera videos. However, Li and Chellappa [9] have recently proposed a technique to perform motion segmentation for group behavior recognition (GaTech Football Play dataset), albeit from the same viewpoint.

As opposed to human action videos which often capture frontal pose of the actors, from ground level viewpoints, events related to surveillance scenarios lend themselves appropriately to exploration of the task under consideration, i.e., transformation invariant event matching and recognition. This is due in part to the fact that surveillance videos depict a a reasonably wide field of view of the scene, and are often captured from high oblique to nadir viewpoints. The proposed similarity transformation therefore is well suited to view invariant matching of events, given that due to overhead views, the most dominant components of transformation are mostly related to rotation and scaling, in addition to translation, as opposed to perspective.

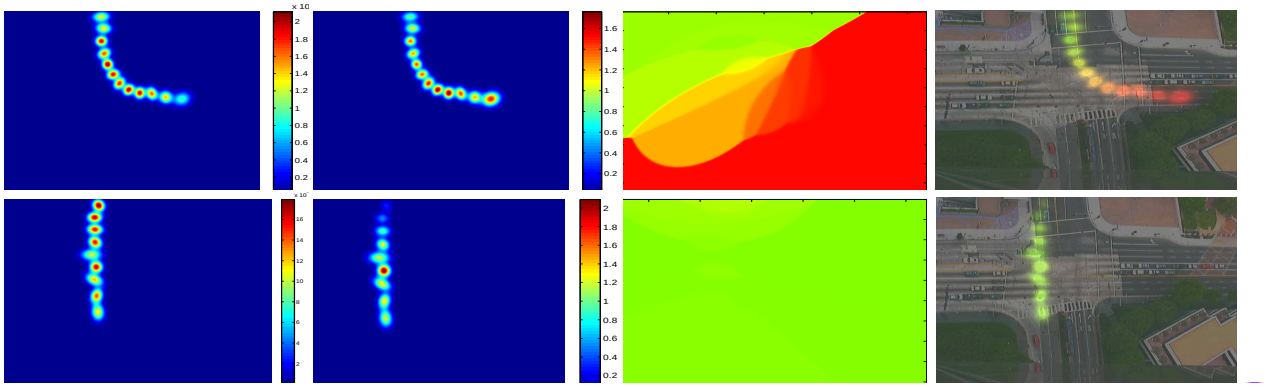
Given input videos, the goal of the proposed framework therefore, is to estimate a measure of similarity between representations of two observed event instances, F and G, regardless of the view from which they were observed. This goal can be achieved in three main steps:

(i) Automatic detection of events and their representation as probabilistic distributions, $f(\mathbf{x})$ and $g(\mathbf{x})$;

(ii) Analysis of the effect of a transformation, $\mathbf{T}$, on the multivariate distribution, $g(\mathbf{x})$, to analytically obtain a *transformed* distribution, $g_{\mathrm{T}}(\mathbf{x})$; and

(iii) Estimation of the transformation, $\mathbf{T}$, such that some measure of dissimilarity or divergence, $\mathcal{D}$, between f and g_{T} is minimized. A test example of an event can then successfully be matched against all learned event models for the purpose of classification or labeling.

The proposed framework begins with estimation of motion patterns that correspond to semantically meaningful events observed in the videos. Our choice for the representation is a mixture of Gaussian model, similar to [11], which is learned over the four dimensional space (x, y, u, v), where (x, y) corresponds to a pixel location, and (u, v) is the instantaneous flow observed at (x, y). The reason for this choice is that this is a rich generative representation which is ideally suited to obtaining an analytical form for the representation of transformed motion patterns, as opposed to histogram representations [14] and topic models [12]. Although the estimation of motion patterns is not claimed to be a novel contribution of our framework, we introduce a few noteworthy improvements and simplifications over the method of [11], as detailed in section 2. In the second main step, we propose to employ a similarity transformation, $\mathbf{T}$ to obtain parameters of a transformed motion pattern. This is a reasonable assumption given that the proposed event representation essentially captures motion in a plane, which is often observed from nadir viewpoints, essentially mimicking an orthographic projection. The goal in the third step is to estimate the rotation and scale parameters $\mathbf{R}$, and translation parameters $\mathbf{t}$, of the transformation, $\mathbf{T}$, that minimizes a measure of divergence between two Gaussian mixtures. In the proposed method, we attempt the minimization of Kullback−Leibler (KL) divergence. The KL divergence for Gaussian mixtures has no closed form expression, and its minimization is not trivial. We propose an approximation to the minimization, which bears resemblance to the Iterative Closest Point (ICP) algorithm [1], wherein we present novel application-specific cues and point set matching techniques.



(a) marginal probability (b) expected magnitude (c) expected direction (d) expected flow

Figure 2. Visualization of Gaussian mixture distributions representing two events, left turn (top), and acceleration (bottom). Each image shows, (a) probability of a pixel belonging to the pattern, $\int\int p(\mathbf{x})dudv$, (b) conditional expectation of flow magnitude, given pixel locations, $\mathrm{E}[\sqrt{u^2+v^2}|x,y]$, and (c) conditional expected flow orientation, $\mathrm{E}[tan^{-1}(v/u)|x,y]$. The image in (d) shows (b) and (c) in terms of brightness and color respectively, according to the color wheel. Notice that (a) and (b) will not always be similar, for example, in the second row, although all pixels in the pattern have high likelihood, the magnitude increases as the objects travel downward (acceleration).

2. Proposed Framework

We begin the description of the proposed framework with the method used for detection and estimation of motion patterns representing real world events. We then investigate the effect of a similarity transformation on the 4d motion pattern probability distribution, and finally derive our algorithmic framework for simultaneous estimation of the transformation and matching score between two pattern distributions.

2.1. Motion Patterns Estimation

For the purpose of representing events as distributions, we first need to categorize pixels of a video into clusters that correspond to an event. This step employs the method of [11] after a few modifications, to perform a hierarchical clustering of data points representing pixel locations and their flows. The first stage in this step is to estimate optical flow from consecutive frames of the video. Since we also propose to detect and compare events in aerial (UAV) videos, e.g., the CLIF dataset [2], which the related methods mentioned earlier do not deal with, image alignment and warping is performed as a pre-processing step to remove camera motion.

Given a large set of flow vectors, $\mathbf{x} = (x, y, u, v)$, within a short video clip containing a few frames, obtained via optical flow [5], K-means clustering is first performed in the four dimensional space. The cluster means, covariance matrices, and weights (percentage of points belonging to each cluster), eventually become the parameters of the components in the Gaussian mixture. Our goal is to cluster these components again, across *all* video clips, so that each high level cluster of components represents a single event as a Gaussian mixture. As opposed to performing an expensive, multistage approach for this goal, as suggested in [11], which involves point sampling and computation of KL divergences, we obtain the Gaussian mixtures in a single step. In this step, a planar graph, $C = (V, E, W)$, is constructed, where V, the set of vertices is the set of *all* clusters obtained from K-means, E, the set of edges connects k nearest neighbors of each vertex, and W represents a sparse weight matrix with non-zero entries corresponding to the edges in E. The weight between two, four dimensional Gaussian components, $(\mu^q, \mathbf{\Sigma}^q)$, and $(\mu^r, \mathbf{\Sigma}^r)$, is a *squared weighted Mahalanobis distance*, given as,

$$\begin{aligned} w_{qr} = \quad & \alpha\mathcal{M}\left(\mu^r_{xy}|\mu^q_{xy}, \mathbf{\Sigma}^q_{xy}\right) + \beta\mathcal{M}\left(\mu^r_{\rho}|\mu^q_{\rho}, \sigma^q_{\rho}\right) \\ + \quad & (1-\alpha-\beta)\mathcal{M}\left(\mu^r_{\theta}|\mu^q_{\theta}, \sigma^q_{\theta}\right), \end{aligned} \tag{1}$$

where,

$$\mathcal{M}\left(\mu^r_{xy}|\mu^q_{xy}, \mathbf{\Sigma}^q_{xy}\right) = \left(\mu^r_{xy} - \mu^q_{xy}\right)^\top {\mathbf{\Sigma}^q_{xy}}^{-1} \left(\mu^r_{xy} - \mu^q_{xy}\right), \tag{2}$$

such that the Gaussian component parameters are represented in polar form, (x, y, ρ, θ), instead of cartesian form (x, y, u, v). The underlying idea is that location and flow are heterogenous features with distinctly varying influences, and an unweighted distance involving the full covariance matrix, either in polar or cartesian coordinates for flow, will diminish the influence of flow, compared to spatial locations (x, y). The weight matrix, W is then simply binarized, and the connectivity of the graph C, determines the set of components in each Gaussian mixture. We observed in our experiments, that the decision to ignore the effect of temporal information (as opposed to [11]) does not effect performance. We therefore, obtain a representation of event G, as a motion pattern, which can be written as,

$$g\left(\mathbf{x}\right) = \sum_{n=1}^{N_g} \omega^n \mathcal{N}\left(\mathbf{x}|\mu^n, \mathbf{\Sigma}^n\right), \tag{3}$$

where $\mathbf{\Sigma}$ is the 4×4 covariance matrix, and ω is the mixing proportion for the N_g mixture components. The next step is

to obtain the representation of the event, $g(\mathbf{x})$, undergoing a similarity transformation.

2.2. Transformation of Motion Patterns

Our goal is to obtain the analytical expression for the distribution, $g_{\mathrm{T}}(\mathbf{x})$, which corresponds to the distribution, g, undergoing a transformation $\mathbf{T}$. In other words, we need to estimate the transformed parameters, $(\mu_{\mathrm{T}}, \mathbf{\Sigma}_{\mathrm{T}})$. We begin by observing that the data point, $\mathbf{x} = (x, y, u, v)$ can be interpreted as relating two distinct pixels, (x, y) and (x', y'), such that,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix}, \tag{4}$$

and therefore, instead of transforming $\mathbf{x}$, we can equivalently transform the two points independently, using the same transformation. Since we propose using a similarity transformation, a 2d point p is transformed into p′ as, $\mathrm{p}' = \mathbf{R}\mathrm{p} + \mathbf{t}$, where $\mathbf{R}$ is a 2×2 matrix representing 2d rotation and scale, while $\mathbf{t}$ is a 2d vector of translation. If the data point $\mathbf{x}$, is represented as 5d vector in homogenous coordinates, we can write a 5×5, rigid, linear transformation matrix, $\mathbf{T}$, compactly as,

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{0} & \mathbf{t}^{\top} \\ \mathbf{0} & \mathbf{R} & \mathbf{0}^{\top} \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

such that the transformed data point, $\mathbf{x}'$ is given by $\mathbf{T}\mathbf{x}$. Similarly, the mean of a transformed component, μ_{T}, is given as, $\mathbf{T}\mu$. For the derivation of transformed covariance matrices of the mixture components, assume that the set of original 4d data points generating the Gaussian mixture distribution, are written as a $4 \times n$ matrix, $\mathbf{X}$, and their mean, μ is $\mathrm{E}[\mathbf{X}]$. The covariance of the point set can then be written as,

$$\mathbf{\Sigma} = \mathrm{E}\left[(\mathbf{X} - \mathrm{E}[\mathbf{X}]) (\mathbf{X} - \mathrm{E}[\mathbf{X}])^{\top} \right]. \tag{6}$$

By application of the transformation $\mathbf{T}$ to the point set $\mathbf{X}$, a set of transformed points $\hat{\mathbf{X}} = \mathbf{T}\mathbf{X}$ is obtained, the mean of which has previously been computed as $\mathbf{T}\mu$ by transformation of μ, and can also be written as $\mathrm{E}[\mathbf{T}\mathbf{X}]$. The covariance of the transformed set, $\hat{\mathbf{X}}$ can be derived by simple manipulation as,

$$\mathbf{\Sigma}_{\mathrm{T}} = \mathrm{E}\left[\left(\hat{\mathbf{X}} - \mathrm{E}\left[\hat{\mathbf{X}}\right] \right) \left(\hat{\mathbf{X}} - \mathrm{E}\left[\hat{\mathbf{X}}\right] \right)^{\top} \right] = \mathbf{T}\mathbf{\Sigma}\mathbf{T}^{\top}. \tag{7}$$

It should be noticed in the above derivation, that the data point sets are not in homogenous coordinates and $\mathbf{T}$ is the first 4×4 entries of the matrix shown in equation 5, i.e., $[\mathbf{R} \quad \mathbf{0};\ \mathbf{0} \quad \mathbf{R}]$. This is meaningful since the translation parameter, $\mathbf{t}$, obviously does not effect covariance.

2.3. Divergence Minimization

We recall that the goal of the proposed framework is to find the similarity between two events, represented as Gaussian mixtures, after computing a potential spatial transformation between them. We propose to estimate this transformation, by minimizing KL divergence between the distribution representing the first event, and the distribution of the second event after transformation. We have already explained how to obtain the Gaussian mixture representation, as well as how to compute the parameters of a distribution undergoing transformation. The divergence minimization will not only estimate the transformation, but the minimum divergence serves as the required similarity measure. Therefore, given two events, represented by $f(\mathbf{x})$ and $g(\mathbf{x})$ respectively, we can compute a measure of dissimilarity, $\mathcal{Z}$, between them as,

$$\mathcal{Z}(f, g) = \mathcal{D}(f \| g_{\mathrm{T}^*}), \tag{8}$$

where $\mathcal{D}$ is chosen to be the KL divergence. There however, is no closed form expression for KL divergence between Gaussian mixtures and various approximations are often used. In this paper, divergence is approximated by Monte Carlo sampling over the first distribution, f. A large set of N, 4d points, $\{\mathbf{x}_j\}_{j=1}^{N}$, is thus sampled from the mixture distribution f, where N is typically equal to 1000 in our experiments. A straightforward algebraic simplification of KL divergence differentiation with respect to T allows us to see that,

$$\begin{aligned} \mathbf{T}^* &= \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{j=1}^{N} f(\mathbf{x}_j) \ \log\left(\frac{f(\mathbf{x}_j)}{g_{\mathrm{T}}(\mathbf{x}_j)} \right), \\ &= \underset{\mathbf{T}}{\operatorname{argmax}} \ \log \prod_{j=1}^{N} \sum_{n=1}^{N_g} \omega^n \mathcal{N}\left(\mathbf{x}_j | \mathbf{T}\mu^n, \mathbf{T}\mathbf{\Sigma}^n\mathbf{T}^{\top} \right), \end{aligned} \tag{9}$$

because the term $f(\mathbf{x}_j)$ is independent of $\mathbf{T}$. We see from equation 9, that the problem of estimating the optimal transformation that minimizes the KL divergence, is equivalent to maximum likelihood estimation (MLE) of the Gaussian mixture parameters given the set of sampled points. One notable exception is that in our case, the set of parameters $\{(\omega^n, \mu^n, \Sigma^n)\}_{n=1}^{N_g}$, has previously been computed during motion pattern estimation, and we only seek to estimate the transformation, $\mathbf{T}$. Another equivalent way of describing this problem is that the goal is to estimate the transformation, $\mathbf{T}$, such that the points $\{\mathbf{T}^{-1}\mathbf{x}_j\}_{i=j}^{N}$, (inverse-transformed sample points), have the maximum joint likelihood, given the parameters of g (not g_{T}). Notice however, that this is not an easier problem compared to estimation of mixture parameters (e.g., using Expectation Maximization), and an analytical expression of the joint likelihood derivative, with respect to even the linear transformation $\mathbf{T}$, is non-trivial. We therefore propose an approximate, iterative maximization algorithm that shares similarities with the Iterative Closest Point (ICP) algorithm [1].

Given the point set $\{\mathbf{x}_j\}_{j=1}^{N}$ sampled from f, which we now write as $\mathbf{X}_f$, we assume the initial value of the

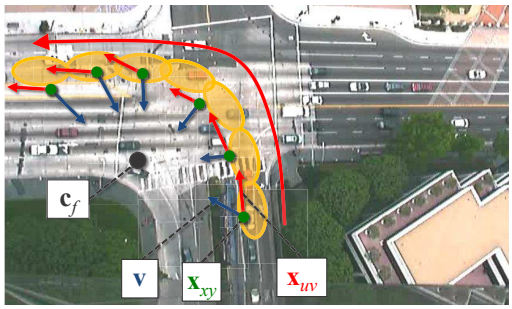


Figure 3. Illustration of the "path-context" cue $\mathrm{s} = \mathbf{x}_{uv}.\mathbf{v}$. The green dots represent the spatial component of 4d points sampled from the mixture of Gaussians, represented as yellow error ellipses. The red arrows depict flow direction of each sampled point, and blue arrows represent $\mathbf{v}$, pointing towards the center of the mixture, $\mathbf{c}_f$, shown as black dot. The scalar 's' acts as a path-context for each sampled point, and is a cue towards its location *within* the event distribution.

transformation to be identity, and write it as $\mathbf{T}_0$. In other words, $\mathbf{R}_0 = \mathbf{I}_{2\times 2}$, and $\mathbf{t}_0 = [0\ 0]$. The proposed maximization algorithm then begins by first sampling another set of N points, $\mathbf{X}_{g_{\mathrm{T}_0}}$, from $g_{\mathrm{T}_0} \equiv g$. At each iteration, $i \geq 1$, the goal of the maximization process is to iteratively find 1-1 correspondence between the point sets, $\mathbf{X}_f$ and $\mathbf{X}_{g_{\mathrm{T}_{i-1}}}$, and compute intermediate transformation, $\hat{\mathbf{T}}_i$, so that, $\mathbf{T}_i = \hat{\mathbf{T}}_i \mathbf{T}_{i-1}$. The underlying idea can be visualized by considering the ideal case, where a perfect correspondence result implies, $\mathbf{X}_f \equiv \mathbf{X}_{g_{\mathrm{T}_i}}$, and therefore, as $N \to \infty$, $f \equiv g_{\mathrm{T}_i}$, and $\mathcal{Z}(f, g) = 0$.

As opposed to most methods employing ICP which, for each point in a set, find the closest corresponding point in the other set, we propose to minimize the global cost of matching by defining a meaningful weighting mechanism. We first define a graph, $H = (V, E, W)$, where, $V = \mathbf{X}_f \bigcup \mathbf{X}_{g_{\mathrm{T}_i}}$, $E = \{e_{jk}\}$, where $1 \leq j, k \leq N$, and $e_{jk} \in \{0, 1\}$, and $e_{jk} = 1$ iff $V_j \in \mathbf{X}_f \bigwedge V_k \in \mathbf{X}_{g_{\mathrm{T}_i}}$. We define W to be an $N \times N$ weight matrix, even though V is size $2N$. This is because by definition of E, $\mathbf{X}_f$ and $\mathbf{X}_{g_{\mathrm{T}_i}}$ are independent sets of H.

We observe that even the choice of similarity as a transformation can make the optimization process end up in a local minima, as shown in figure 7. Therefore, to incorporate an additional cue during optimization, we compute a vector for each distribution sample that points towards the distribution's spatial center. We compute the center as,

$$\mathbf{c}_f = \sum_{n=1}^{N_f} \omega^n \mu_{xy}^n, \tag{10}$$

and for each point, $\mathbf{x} \in \mathbf{X}_f$, we then compute a vector, $\mathbf{v} = \mathbf{c}_f - \mathbf{x}_{xy}$, which points from each point to the mixture's center. We then take the inner product of this vector with the sample point's optical flow and use it as an additional

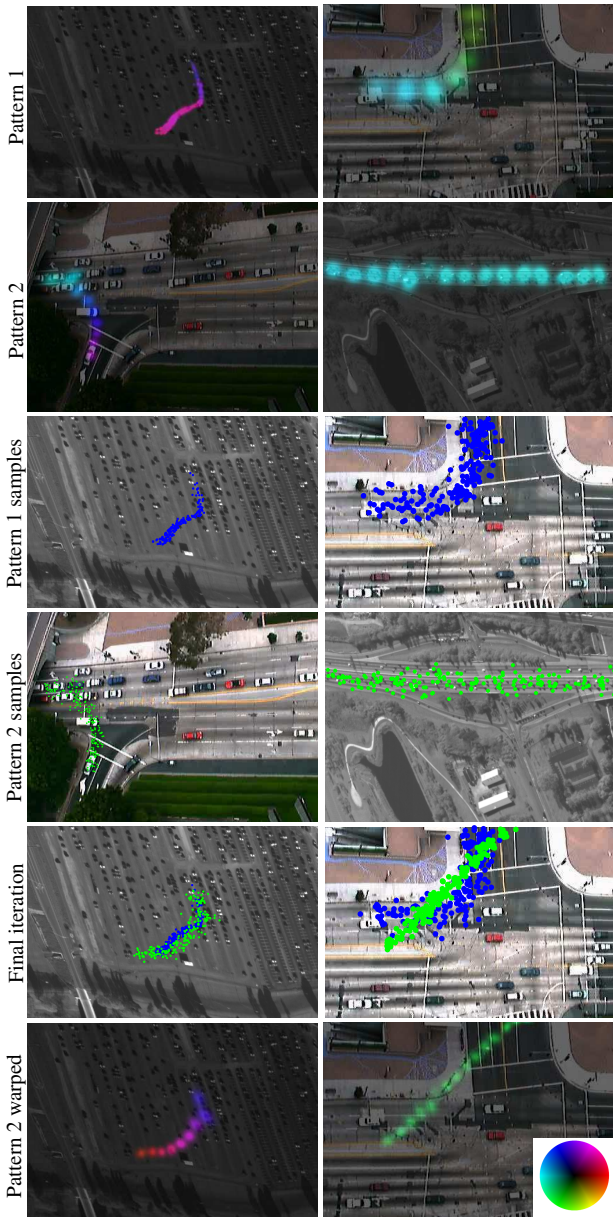


Figure 4. Transformation estimation for two pairs of events. The pair shown in the 2nd column is a mismatch. Notice that the sampled points shown in rows 3, 4, and 5, are actually 4D (flow not depicted). Compare the last row to first, to gauge quality of estimated transformation.

weight in the graph G. We call this product "path-context", and write it as $\mathrm{s} = \mathbf{x}_{uv}.\mathbf{v}$. This constraint is applied to avoid the matching of patterns with a directional mismatch, which can easily happen, for instance with events that represent motion in straight line. Such a mismatch indicates the convergence of maximization process to a local minimum. We demonstrate by extensive experiments that the proposed framework avoids this scenario. The usefulness of this con-

Input: 4d Guassian mixture distributions, f and g, representing two motion patterns, or events.
Output: $\mathbf{T}$ and $\mathcal{D}\left(f \| g_{\mathrm{T}}\right)$, such that $\mathcal{D}$ is minimum.

▶ Initialize index, $i \leftarrow 0$
▶ $\mathbf{R}_i \leftarrow \mathrm{I}_{2\times 2}$, $\mathbf{t}_i \leftarrow [0 \;\; 0]$
▶ Construct matrix $\mathbf{T}_i$ as in eq. 5
▶ $\mathcal{Z}_i \leftarrow \infty$
▶ Compute $\mathbf{c}_f$ using eq. 10.
▶ Repeat while $\mathcal{Z}_i > \epsilon$ and $i < i_{\max}$,
- $\mathbf{X}_f \leftarrow \left\{\mathbf{x}_j | \mathbf{x}_j \sim f(.)\right\}_{j=1}^{N}$
- For each $\mathbf{x}_j \in \mathbf{X}_f$,
  - ▲ $\mathbf{v}_j \leftarrow \mathbf{c}_f - \mathbf{x}_{xy}$
  - ▲ $\mathrm{s}_j \leftarrow \mathbf{x}_{uv} \cdot \mathbf{v}_j$
- $\mathbf{X}_{g_{T_i}} \leftarrow \left\{\mathbf{x}_k | \mathbf{x}_k \sim g_{T_i}(.)\right\}_{k=1}^{N}$
- Compute $\mathbf{c}_{g_{T_i}}$ using eq. 10.
- For each $\mathbf{x}_k \in \mathbf{X}_{g_{T_i}}$,
  - ▲ $\mathbf{v}_k \leftarrow \mathbf{c}_{g_{T_i}} - \mathbf{x}_{xy}$
  - ▲ $\mathrm{s}_k \leftarrow \mathbf{x}_{uv} \cdot \mathbf{v}_k$
- Create $N \times N$ weight matrix, W
- Set w_{jk} as in eq. 11, for $1 \leq j, k \leq N$
- Find correspondences between $\mathbf{X}_f$ and $\mathbf{X}_{g_{T_i}}$ ([8])
- Convert point sets to $\mathbf{X}'_f$ and $\mathbf{X}'_{g_{T_i}}$, using eq. 4
- Perform linear least squares to estimate $\hat{\mathbf{T}}_i$
- $\mathbf{T}_i \leftarrow \hat{\mathbf{T}}_i \mathbf{T}_{i-1}$
- $\mathcal{Z}_i \leftarrow \mathcal{D}\left(f \parallel g_{T_i}\right)$
- $i \leftarrow i + 1$

▶ Return $\mathbf{T}_i$ and $\mathcal{Z}_i$

Figure 5. Algorithmic overview of the proposed framework. See text for details.

straint is illustrated in figure 3. Given the data points from the two sets and their path contexts, the entries of the $N \times N$ weight matrix, W, are defined as,

$$w_{jk} = \lambda \|\mathbf{x}_j - \mathbf{x}_k\|_2 + (1-\lambda)|\mathrm{s}_j - \mathrm{s}_k|, \qquad (11)$$

where, λ is a fixed value parameter that serves to balance the influence of Euclidean distance between heterogenous vector $\mathbf{x}$, and absolute difference between scalar s. The imbalance between these quantities also arise due to the vastly disparate ranges of values they take. The correspondences are then established by bipartite graph matching over H, using the Hungarian algorithm [8]. An illustration of establishing the first point correspondence between $\mathbf{X}_f$ and $\mathbf{X}_{g_{T_0}}$ can be seen in figure 6. Given the N correspondences, 2d point sets with $2N$ elements are obtained, where the N members correspond to locations (x, y), while the second half incorporates flow correspondences, by computing (x', y'), as in equation 4. A least squares fit over the combined set is then performed to obtain, $\hat{\mathbf{T}}_i$. We observed in our experiments, that the proposed maximization algorithm converged quickly, i.e., often within 5 iterations. It is also

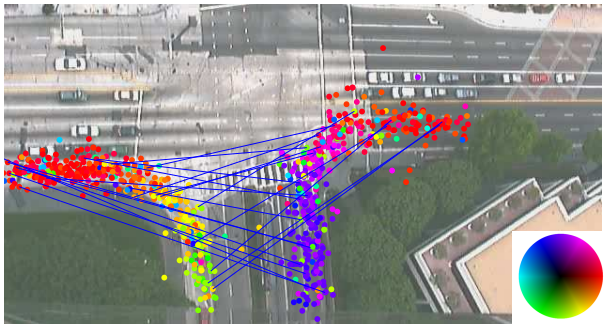


Figure 6. Illustration of the correspondence established between point sets $\mathbf{X}_f$ and $\mathbf{X}_{g_{T_0}}$, depicted as blue lines. Both sets are sampled form motion patterns corresponding to 'right turns', and only a few randomly chosen correspondences are shown to avoid clutter. The colored dots represent data points sampled from the two mixtures, and their orientation and magnitude are shown by the color and brightness as per the color wheel. Notice that although the correspondences cannot be exact, they give a very reasonable estimate of the transformation between the mixtures.

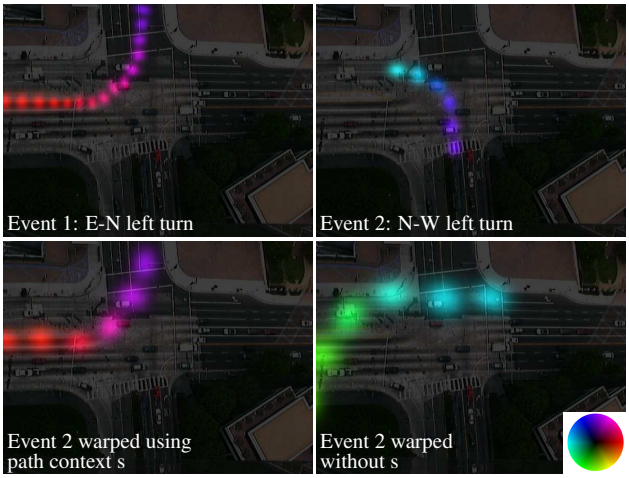


Figure 7. An example of a transformation of a 'left turn', onto another 'left turn'. Without path context optimization results in a local minimum (bottom right) whereas with the use of path context the correct transformation is achieved.

worth noting that in order to avoid sampling bias, both the sets $\mathbf{X}_f$ and $\mathbf{X}_{g_{T_i}}$, are sampled anew in each iteration. Moreover, the set $\mathbf{X}_{g_{T_i}}$ at the i^{th} iteration is not sampled from g and transformed, rather the parameters of g_{T_i} are computed and the points are sampled from the transformed distribution. An overview of the algorithm is listed in figure 5.

3. Experiments and Results

We performed extensive experiments on motion patterns estimation for a set of diverse events. A broad variety of videos were used, including the NGSIM dataset [10], CLIF dataset [2], and a collection of publicly available videos from Getty images. Notice that the CLIF dataset consists of wide area aerial videos, and therefore the proposed frame-

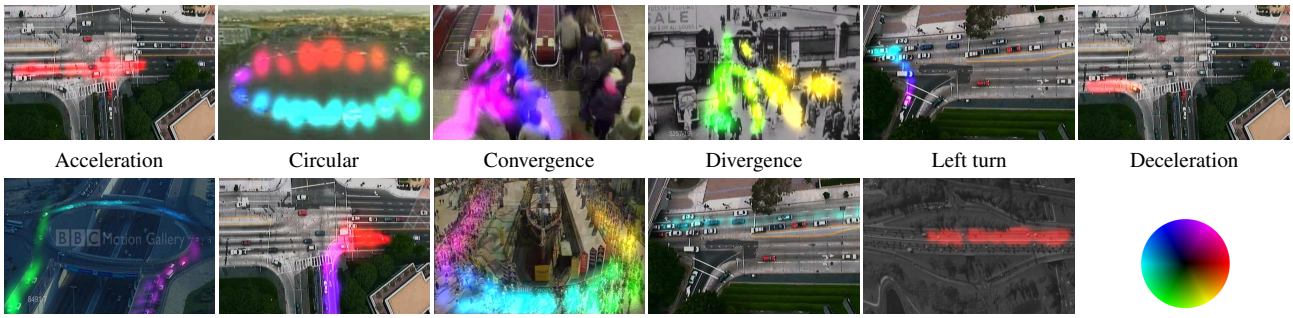


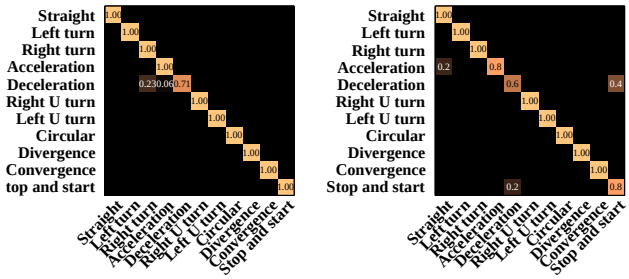
Figure 8. Examples of events learned using the proposed approach, visualized as conditional expected optical flow given pixel locations.

Event class / Error ($\mu \pm \sigma$)	$\mathbf{t}_x$ (% pixels)	$\mathbf{t}_y$ (% pixels)	**Scale** (%)	**Angle** (% rads)
Straight	-0.2 $\pm$ 0.3	0.1 $\pm$ 0.1	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
Left Turn	-0.2 $\pm$ 0.4	0.2 $\pm$ 0.4	0.1 $\pm$ 0.0	0.0 $\pm$ 0.1
Right Turn	-0.2 $\pm$ 0.2	0.0 $\pm$ 0.1	0.0 $\pm$ 0.0	-0.1 $\pm$ 0.2
Acceleration	-0.5 $\pm$ 0.8	0.0 $\pm$ 0.0	0.1 $\pm$ 0.0	0.0 $\pm$ 0.0
Deceleration	-0.1 $\pm$ 0.1	0.1 $\pm$ 0.1	0.1 $\pm$ 0.0	0.0 $\pm$ 0.0
Right U Turn	-1.9 $\pm$ 2.4	0.6 $\pm$ 1.6	0.2 $\pm$ 0.1	0.0 $\pm$ 0.1
Left U Turn	-3.7 $\pm$ 7.9	0.7 $\pm$ 3.4	0.3 $\pm$ 0.1	0.1 $\pm$ 0.2
Circular	-2.2 $\pm$ 4.5	0.1 $\pm$ 2.0	0.4 $\pm$ 0.2	-0.1 $\pm$ 0.2
Divergence	-0.9 $\pm$ 1.8	0.3 $\pm$ 1.1	0.1 $\pm$ 0.0	0.0 $\pm$ 0.0
Convergence	-0.3 $\pm$ 0.8	0.3 $\pm$ 0.5	0.1 $\pm$ 0.0	0.1 $\pm$ 0.1
Stop and Start	0.0 $\pm$ 0.2	0.1 $\pm$ 0.1	0.1 $\pm$ 0.1	0.0 $\pm$ 0.0

Table 1. Average transformation error for synthetically transformed event distributions. The manual transformation, followed by estimation of the transformation allows quantification of performance of the proposed method, by comparison against ground truth transformation parameters. Error is reported as 'error in value, divided by the value to be estimated'.

work is the first to perform event recognition by statistical motion patterns modeling in moving cameras, as opposed to related methods [12, 6, 11] which have experimented only with static camera surveillance scenarios. Moreover, the proposed similarity measure (equation 1) allowed us to learn and recognize complex events including convergence, divergence, move-stop-move conditions, and acceleration, deceleration patterns, in addition to relatively simplistic straight line and turn events learned in existing methods. A few examples of the 11 event categories discovered and classified using our approach are shown in figure 8.

One of the first experiments performed for testing of the proposed approach was to synthetically transform distributions of estimated event motion patterns. Given a mixture distribution, it is transformed using the analytical expression derived in section 2.2, and finding transformed parameters by equation 7, etc.. Arbitrarily chosen values of parameters given by $\mathbf{R}$ and $\mathbf{t}$ result in new events, using which the proposed matching framework is tested. The transformed pattern is then warped to the original distribution by estimation of $\mathbf{T}$ using the proposed iterative KL divergence minimization framework. This experiment allows us to directly compare the parameters of the estimated transformation against the ground truth parameters. The quantitative results of this experiment are reported in table 1.

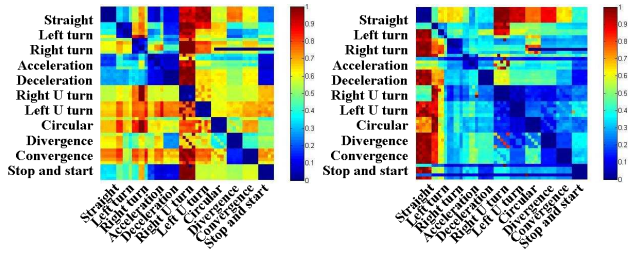


(a) KL Divergence (b) Graph matching cost

Figure 9. Confusion tables for classification of 132 event instances, into 11 classes. Using KL divergence based NN, the average accuracy was 96.21%. An accuracy of 92.73% was achieved when using graph matching cost at last iteration for classification.

The main goal of the proposed framework is the ability to recognize event classes by matching two event distributions after estimating the potential transformations between them. We tested this capability by classifying 132 event distributions into one of the 11 event classes. Some of these distributions were synthetically generated by applying random transformations to existing patterns as mentioned earlier. Each class was represented by a single mixture distribution chosen arbitrarily, and all 132 *test* event instances were attempted to be matched to each of the 11 *model* distributions, and the corresponding KL divergences at last iteration were noted. The test distributions were then given the label of the model distribution with the least KL divergence. The global correspondence framework employed in the proposed method resulted in very encouraging results for this experiment, and even though instances of the same class are not exactly the same shape, an average classification accuracy of 96.21% was obtained. The confusion table for this experiment is shown in figure 9(a).

We also tested a different measure of similarity or distance between the matched event distributions. While exploiting the same iterative transformation estimation framework, we used the cost of matching the graph, H, as the distance between the model and transformed test event patterns. This experiment allowed us to separate the influence of transformation and matching steps, by choosing differ-



(a) KL Divergence (b) Graph matching cost

Figure 10. Matrices depicting "post-transformation" dis-similarity between all pairs of event instances. High similarity blocks along diagonal illustrate discrimination between classes.

ent metrics for each step. The results of this experiment were also very satisfactory and largely similar to the first, as shown in figure 9(b).

To quantify and visualize the discriminative nature of our rich statistical event distributions, and the proposed transformation estimation framework, we also computed exhaustive, pair-wise distances between 55 event instances, taken evenly from 11 classes, resulting in a self-similarity matrix for each classification metric. The two matrices obtained using KL divergence and graph matching cost as distance measures are shown in figure 10(a) and (b) respectively. The clearly visible 5×5 low value blocks along the diagonal conclusively depict low intra-class, and high inter-class distances after transformation estimation.

4. Conclusion

In conclusion, we propose a framework for discovery, representation and learning of a broad range of typically observable pedestrian and vehicular events, in static as well as moving camera surveillance videos. More importantly, given the statistical representation of event instances as Gaussian mixture distributions, we proposed a principled, rigorously derived framework for estimation of a potential similarity transformation between two distributions, such that the KL divergence between them is minimized. The same divergence measure after estimation of the transformation is used to provide a measure of similarity for nearest neighbor classifier based event recognition. The proposed framework is not only the first to learn generative model of location and flow for generalized events in moving camera videos using optical flow while avoiding the need for tracking, but is also the first to perform similarity invariant recognition of those events. Results obtained by performing experiments on a wide range of videos and events, validate our approach.

Acknowledgements

This research was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-10-C-0112. Any opinions, findings and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

[1] P. Besl and N. McKay. A registration of 3d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] Columbus Large Image Format (CLIF) 2006 Dataset. Available at https://www.sdms.afrl.af.mil/datasets/clif2006/.

[3] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *IEEE Int. Conf. on Computer Vision*, 2009.

[4] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. "A system for learning statistical motion patterns". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, September 2006.

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

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

[7] D. Kuettel, M. Breitenstein, L. Van Gool, and V. Ferrari. What's going on? discovering spatio-temporal dependencies in dynamic scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[8] H. Kuhn. The hungarian method for the assignment problem. *Naval Res. Logistics Quarterly*, 2(1-2):83–97, 1955.

[9] R. Li and R. Chellappa. Group motion segmentation using a spatio-temporal driving force model. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

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

[11] I. Saleemi, L. Hartung, and M. Shah. Scene understanding by statistical modeling of motion patterns. In *CVPR*, 2010.

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

[13] D. Weinland. A survey of vision-based methods for action representation, segmentation and recognition. *CVIU*, 2010.

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