Gaussian Process Regression Flow for Analysis of Motion Trajectories

Kihwan Kim   Dongryeol Lee   Irfan Essa

{kihwan23, dongryel, irfan}@cc.gatech.edu

Georgia Institute of Technology, Atlanta, GA, USA

`http://www.cc.gatech.edu/cpl/projects/gprf/`

Abstract

Recognition of motions and activities of objects in videos requires effective representations for analysis and matching of motion trajectories. In this paper, we introduce a new representation specifically aimed at matching motion trajectories. We model a trajectory as a continuous dense flow field from a sparse set of vector sequences using Gaussian Process Regression. Furthermore, we introduce a random sampling strategy for learning stable classes of motions from limited data. Our representation allows for incrementally predicting possible paths and detecting anomalous events from online trajectories. This representation also supports matching of complex motions with acceleration changes and pauses or stops within a trajectory. We use the proposed approach for classifying and predicting motion trajectories in traffic monitoring domains and test on several data sets. We show that our approach works well on various types of complete and incomplete trajectories from a variety of video data sets with different frame rates.

1. Introduction

Motion trajectories of moving targets are essential for providing information about an object's movement patterns over time. Visual analysis of such moving objects in videos, is therefore a considerably important topic [6, 10, 14, 20, 24]. A key element in such analysis is representing the trajectory patterns of objects in a manner that allows for effective discrimination of these trajectories from specifically known and unknown patterns. Such forms of discrimination are usually evaluated by comparing spatial proximity between trajectory points with known (or learned) reference patterns or by clustering all the input trajectories to find similar patterns [6, 11, 25].

Such approaches for analysis of trajectories are useful for recognizing motion patterns that are representative of the data, but do not necessarily support (1) detecting complex time-varying patterns, caused by subtle changes of acceleration or sudden stops, which may be important for predicting changes in headings or detecting unknown or anomalous patterns, or (2) recognizing incomplete trajectories for instant and online monitoring. Such instances require modeling and analyzing motion trajectories in a spatio-temporal domain. Furthermore, constructing reference models from videos having (a) sufficiently *representative* patterns for sequences of varying lengths, and (b) different *temporal sampling* (frame-rates) remains a difficult task. The goal of this
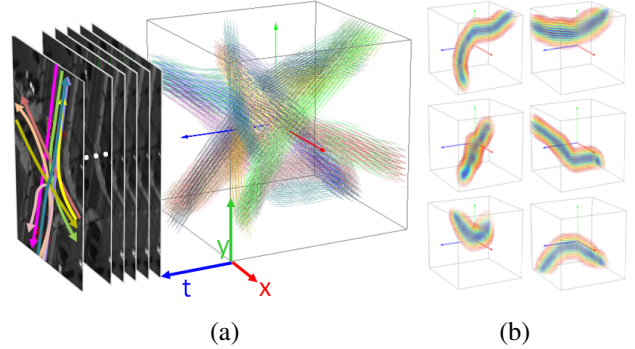

Figure 1. **Gaussian Process Regression Flows (GPRF)**: **(a)** Video frames having trajectories, and the normalized *mean flows* representing 17 different *spatio-temporal patterns* of motion trajectories are visualized in different colors. Each mean flow is generated from trajectories extracted from videos using Gaussian process regression. We classify each online track by traversing inside the flow fields and collecting posterior densities along their paths. **(b)** Some of individual mean flows are shown : blue indicates that the mean flows in the region have lower variances (and thus higher certainty), and red indicates higher variances.

paper is to present a representation that allows for matching of complex trajectories. We demonstrate the validity of this representation for observing and modeling traffic patterns from video.

Specifically, in this paper, we propose a novel representation of motion trajectories using Gaussian Process Regression (GPR). This representation supports effective recognition of normal and anomalous patterns by generating a continuous vector field, a Gaussian Process Regression Flow (GPRF) (Fig. 1). We demonstrate using this framework (Fig. 2) a process to take visual observations (trajectories) and generating a flow field that provides *representative motion tendencies* even with a minimal amount of data in each training set. Furthermore, this representation is invariant to temporal sampling constraints (*i.e.*, frame-rates of data can vary from trajectory to trajectory). Our use of flow fields, which describe statistical certainties in terms of their spatial shape and temporal tendency, can effectively classify and predict motion patterns, and detect anomalous tracks from incremental online trajectories. We show that the proposed spatio-temporal flow can efficiently model complex motions, such as stopping or accelerating of cars in traffic videos.

The main contributions of this paper are: **(1)** A novel method for modeling a trajectory as a 3D continuous dense flow field from sparse set of vector sequences using Gaus-
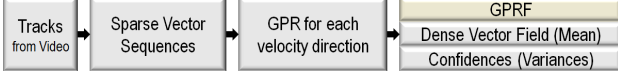
Figure 2. **Overview of our framework:** From tracks to GPRF.

sian Process Regression; this enables modeling complex patterns, especially as seen in videos of traffic. **(2)** A random sampling strategy for learning stable multiple classes of flow field; this generates a fair distribution of certainties and alleviates unwanted variation. **(3)** A set of effective metrics to compare an input trajectory (incomplete or complete) to learned flow fields. And, **(4)** a locally dominant ratio for effectively discriminating a correct pattern from incorrect ones, and calculating ambiguities used for detecting anomalous patterns.

1.1. Related Work

Analysis of motion trajectories has recently been applied to video surveillance with variety of approaches [6, 8, 9, 23]. Effective methods for motion trajectory matching are necessary for such analysis. Zhang [25] and Morris [11] compare different distance metrics used for motion trajectory matching and show that alignment based approaches such as Dynamic Time Warping (DTW) and Longest Common Subsequences (LCS) give the most stable results. However, these approaches compare only the spatial proximity of temporal vector sequences, and do not model second order movements (*i.e.*, acceleration and deceleration). In other efforts, Hu [6] successfully demonstrates a framework that clusters traffic motions and detects anomalous trajectories using statistical analysis of tracks. While our work focuses on a similar analysis of motions, their method does not support modeling complex scenarios like stop-and-go.

Recently, Gaussian Process (GP) has been widely used in the field of human motion analysis [21], tracking [18], surface modeling/completion [17], and data classification [2] for its compactness and powerful inference mechanism for missing data. Ellis et al. [5] models 2D displacement vectors with GPR, and the regressed vectors are used to update state space equations in tracking algorithms. In our work, we adopt the GP regression method to build a 3D spatio-temporal flow field and use that to generate a representation for matching.

Time series signal analysis of this form is also applicable to event/action retrieval. Lin [10] introduced a powerful framework measuring a distance between normalized signals by adding normally distributed weights for output values. This provides us with an important concept for normalizing flow fields.

2. Gaussian Process Regression for Flow

To model time-varying motions, we first extend the motion trajectories extracted from video sequences into the
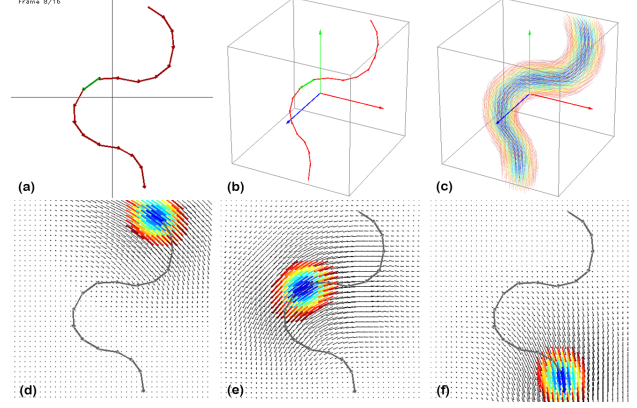

Figure 3. **Example of GPRF, and mean flows: (Upper row) (a)** Sample trajectory having a duration of 16 frames: each arrow starts from the location of $x(u, v, t)$, and has the velocity values **(b)** The trajectory in (u, v, t) space $\in \Re^3$ **(c)** GPRF generated from the trajectory **(Bottom row)** The projected mean flows in each of $u - v$ slice at $t = 0$ **(d)**, $t = 8$ **(e)**, and $t = 16$ **(f)** respectively. Mean flows with different levels of confidence exist in any grid points in the space. In each image, only mean flows having the variance of less than half the maximum variance among 95% confidence are shown as color values. The colors vary from the larger posterior variances (red) to smaller variances (blue).

spatio-temporal domain. We define a trajectory as a discrete vector sequence in $\Re^3$. Let $x \in \Re^3$ be the position (u, v, t), and the position sequence be $\mathbf{x} = \{x_1, \dots, x_n\}$. Moreover, let $y \in \Re^3$ be the velocity of each direction (y_u, y_v, y_t) [1], and the velocity sequence be $\mathbf{y} = \{(y_{1,u}, y_{1,v}, y_{1,t}), \dots, (y_{n,u}, y_{n,v}, y_{n,t})\}$. We also denote the sequence of each velocity component as: $\mathbf{y}_u = \{y_{1,u}, \dots, y_{n,u}\}$, $\mathbf{y}_v = \{y_{1,v}, \dots, y_{n,v}\}$, $\mathbf{y}_t = \{y_{1,t}, \dots, y_{n,t}\}$. The trajectory sequence of n vectors are defined as $\mathrm{T}_n = \{\mathbf{x}, \mathbf{y}\}$. We first briefly introduce Gaussian Process Regression (GPR) and explain how we use it for modeling the dense vector field from the set of sparse vector sequences (Fig. 2).

2.1. Constructing Posterior Density

We consider the regression model $y = f(x) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. We use a Gaussian process model on f with a mean function $m(x) = \mathbb{E}[f(x)] = 0$ and a covariance function $K(x, x'') = \mathbb{E}[(f(x) - m(x))(f(x'') - m(x''))] = \mathbb{E}[f(x)f(x'')]$. The observation vector $\mathbf{y} = \{y_1, \dots, y_n\}$ then follows a zero-mean multivariate Gaussian with a covariance matrix, $\mathbf{K}^* = \mathbf{K} + \sigma^2 \mathbf{I}$, where $[\mathbf{K}]_{ij} = K(x_i, x_j)$. The posterior density for a test point x^*, $p(y^*|x^*, \mathbf{x}, \mathbf{y})$, is a univariate normal distribution with the mean $\bar{y}^*$ and the variance $\mathrm{var}(y^*)$:

$$\bar{y}^* = k(x^*)^T (K^*)^{-1} y$$
$$\mathrm{var}(y^*) = K(x^*, x^*) - k(x^*)^T (K^*)^{-1} k(x^*), \quad (1)$$

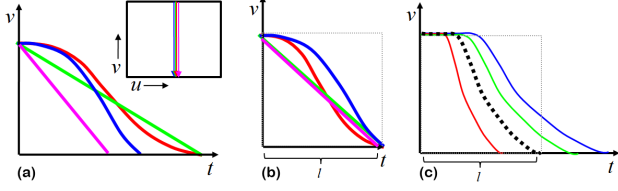[1] y_t is a frame difference, thus it is usually a constant. See Section 2.2

Figure 4. **Normalizing frames**: **(a)** Each red, blue, green and magenta line indicates the trajectories from tracks started from a same spatial position toward a same destination with different velocities, and accelerations (or in different frame rates). The image in the upper-right shows trajectories projected onto the spatial domain. **(b)** All of the trajectories are normalized in time (t) axis as l. **(c)** Another example showing that each colored line has the same trajectory with the same velocity and acceleration but diverges later. The dashed black line shows the normalized trajectory.

where $k(x^*) = [K(x^*, x_1), \ldots, K(x^*, x_n)]^T$. In our case, we train a separate GP regression for each of $\mathbf{y}_u$, $\mathbf{y}_v$, and $\mathbf{y}_t$ at a given point set $\mathbf{x}$.

We chose Gaussian Automatic Relevance Determination (ARD) kernel [4] as the covariance function. We select its hyper-parameters using the limited memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimizer [12] by maximizing the marginal log-likelihood.

2.2. Mean Flow and Confidence Band

Since we want to model a continuous flow field where the function represents velocities for each spatial direction and the uniform temporal grid (frame difference) at any point $x = (u, v, t)$, we calculate a GP regression model for each velocity component. Hence, each of the posterior densities $p(y_u^*|x^*, \mathrm{T}_n)$, $p(y_v^*|x^*, \mathrm{T}_n)$, and $p(y_t^*|x^*, \mathrm{T}_n)$ are computed separately. We can then express the mean flow (see Fig. 3) as a vector field:

$$\Phi(x) = \bar{y}_u^*(x)\mathbf{i} + \bar{y}_v^*(x)\mathbf{j} + \bar{y}_t^*(x)\mathbf{k} \in \Re^3, \qquad (2)$$

with a variance for each velocity component $\mathrm{var}(y_u^*(x))$, $\mathrm{var}(y_v^*(x))$, $\mathrm{var}(y_t^*(x))$ respectively. The sequence of mean flow vectors with minimum variances over each time u–v slice (see Fig. 3) defines the approximated version of input trajectories.

We denote the pair of the mean flow Φ and its variances as *Gaussian Process Regression Flow (GPRF)*, the size of output variance in posterior density having 95% of certainty ($1.96\sigma^2$) as a *confidence band (CB)*, and the sequence of mean flow having a minimum variance over each time grid as a *Approximation of Learned Trajectory (ALT)*.

3. Learning Stable GPRFs

In learning GPRFs, there are two issues we need to address: (1) How to model a GPRF from different trajectories, which may have different lengths, and (2) How to handle multiple GPRF models trained from different numbers of trajectories
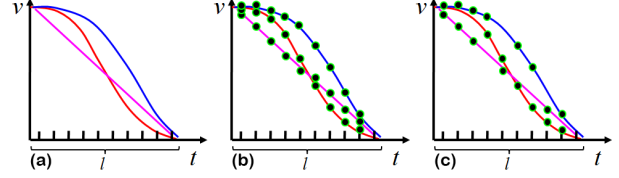

Figure 5. **Sampling points for training**: Suppose we are training a class using three trajectories. **(a)** All trajectories are normalized in l evenly distributed grid in time axis. **(b)** Adding all the samples per time grid for generating GPRF. In this case, training the same number of trajectories have to be guaranteed for all the other classes. **(c)** If we need only two samples per time grid, we randomly sample without replacement two points from three trajectories at each time slice.

with heterogeneous scales and frame rates. Before applying the Gaussian process regression framework, we need to normalize the length of the tracks used for constructing classes (Section 3.1).

3.1. Normalization

Unlike alignment-based approaches (i.e. DTW and LCS) [15, 20], our flow field uses a normalized *frame*, which is similar to the method introduced in [10]. Therefore, the time axis is discretized into l equal sized frames. In practice, when we construct multiple GPRFs from different training sets, the positions of each sample along the t axis and their velocity components of $\mathbf{y}_t$ should be normalized by $\frac{l}{n_i}$, where n_i is the number of sample per trajectory.

Consequently, as shown in Fig. 4, all the GPRFs or trajectories are distributed in the same scale, and we can compare motion trajectories with same *re-sampled* scale regardless of their actual lengths and frame rates. We consider two tracks with different starting positions to be in different classes, even if they have the same set of velocities. However, one should choose an adequate number of samples from each normalized trajectory for constructing a GPRF for each class. In the next subsection, we describe some essential steps for the selection.

3.2. Number of Samples and Variances

The computed variances for each velocity component in Eq. 2 (i.e. $\mathrm{var}(y_u^*(x))$, $\mathrm{var}(y_v^*(x))$, $\mathrm{var}(y_t^*(x))$) depend on (1) the number of nearby samples and (2) the hyper-parameters for the selected covariance function used in each GP model. Unfortunately, the marginal log-likelihood objective is non-concave [16] and we can hope to obtain only locally optimal hyper-parameters unless good initial starting sets of hyper-parameters are provided. Therefore, balancing the adequate number of samples for each training set is a key element for stable GPRF learning. For Gaussian process regression (and GPRF consequently), the confidence band is narrower in regions with more samples. Therefore, comparing an input trajectory to each of

the learned GPRFs, without the consideration of balancing, may be unfair, since it will favor the one with more samples. In addition, redundant observations and noise may affect the variances which would incorrectly quantify the level of *certainty*.

To tackle these problems, we first determine the number of samples per time step and allocate this same number of samples for each of the trained GPRFs. For each time step, we sample among the entire set of trajectories, which are chosen for a specific class, with random uniform probability and allocate samples among them. The random sampling method can alleviate both the unwanted reduction of variance due to the noise and the uneven distribution of variance levels in different classes. In our test, we use three samples per time step, as it gives us maximum confidence difference of 0.8 (worst case) in a resolution of our region of interest (the size of $u - v$ slice).

4. Similarity Measurement

We now have GPRFs computed for each class. Our next task is to evaluate the posterior probability density of a given online testing point x^o and its velocity vector y^o using Eq. 1 and 2 to measure similarity as a form of likelihood in the given training set (class). Let us first denote the online trajectory $T_n = \{\mathbf{x}^o, \mathbf{y}^o\}$ as a trajectory with n observations, and the N–learned classes represented by GPRFs as $\mathbf{X}_k$ $(k = 1 : N)$. Assume that the time component of the online point is already normalized in unit lengths of l.

4.1. Measuring the Local Likelihood of Testing Data

As discussed earlier, we can evaluate a GPRF on any point regardless of the scale of the point. This evaluation is analogous to a prediction using the learned GPRF. In other words, given a GPRF class, the mean velocity vector in the location of testing point x^o is computed as: $\bar{y}_u^*(x^o)\mathbf{i} + \bar{y}_v^*(x^o)\mathbf{j} + \bar{y}_t^*(x^o)\mathbf{k}$ with the variances for each direction.

If the velocity vector of x^o becomes different, the posterior probability density will decrease upon the variance (confidence) of the point (See Fig. 6 (a), and (b)). If the position of x^o is further away from ALT, the variances for the velocity components become larger and result in a smaller posterior probability density (Fig. 6 (c)).

Suppose that we evaluate our online test point using the k–th GPRF, $\mathbf{X}_k$. Given an independent GP prior on each velocity component, we define the local likelihood of the testing point, χ on the training set $\mathbf{X}_k$, as:

$$\chi(x^o, y^o)|_{\mathbf{X}_k} = p(y_u^o|x^o, \mathbf{X}_k)p(y_v^o|x^o, \mathbf{X}_k)p(y_t^o|x^o, \mathbf{X}_k) \quad (3)$$

Note that the magnitude of y^o is normalized, and set to $|\bar{y}^*(x^o)|$ before the calculation. This likelihood is a main local similarity measurement in our approach. The details
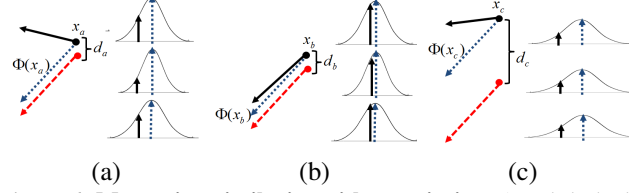

(a)   (b)   (c)

Figure 6. **Measuring similarity with certainties**: A red dashed arrow indicates the approximation of learned trajectory. A blue dotted arrow is a mean flow vector at each test point and a black arrow indicates its velocity vector. In the right side of each image, posterior densities for each u, v, t are shown. **(a)** The test point x_a is close to ALT, but the vectors are quite different. **(b)** Both the location of the input point x_b, and its velocity are close to mean flow and ALT respectively. **(c)** The example of the worst case: The test point x_c is further away from ALT (large variances of the posterior probability densities), and the velocity vectors are different (away from mean).

of how we use this likelihood in multiple classes of GPRF will be discussed in the following section.

5. Classification and Prediction of Trajectories

In this section, we first describe a method to compute the global similarity between the trajectory and learned GPRFs. Using the local likelihood criterion (Eq. 3), we then describe methods for classifying patterns of trajectories, which can be either complete or incomplete. A *complete trajectory* consists of an entire set of segments collected from the moment a car enters to the moment it exits the observation region. Otherwise, a trajectory is *incomplete*. Incomplete trajectories are hard to classify because we do not know (1) when a trajectory ends (i.e. the car exits from the observation region) and (2) how the trajectory varies within the observation region.

5.1. Similarity for Complete Trajectories

Suppose that we have N GPRFs constructed from training datasets $\mathbf{X}_k(k \in [1, N])$, and a complete test trajectory $T_n = \{x_i, y_i\}_{i=1}^n$. We define the global likelihood of an input trajectory to be the k-th GPRF:

$$\mathcal{L}_k(T_n) = \frac{1}{n}\sum_{i=1}^{n}\left[\chi(x_i, y_i)|_{\mathbf{X}_k}\right] \quad (4)$$

We also assign each part of the test trajectory $i = 1 : n$ to the class with the highest local likelihood:

$$l(i) = \arg\max_{k \in [1,N]}\left[\chi(x_i, y_i)|_{\mathbf{X}_k}\right] \quad (5)$$

We define c_k $(k = 1 : N)$ as the number of times $l(i)$ is equal to k for $i = 1 : n$. We can then compute the proportion of the input trajectory to be the k-th GPRF as $\rho_k(T_n) = \frac{c_k}{n}$. The proportion is an effective parameter for the trajectory matching since it reflects how *dominant* the

chosen class is. We refer to $\rho_k(\mathrm{T}_n)$ as a *proportion of dominance*. The final *global similarity* of the input trajectory to the specific GPRF k is then defined as:

$$\mathcal{S}_k(\mathrm{T}_n) = \rho_k(\mathrm{T}_n)\mathcal{L}_k(\mathrm{T}_n). \qquad (6)$$

The index of the closest GPRF is $\arg\max_{k\in[1,N]} \mathcal{S}_k(\mathrm{T}_n)$.

While the classification procedure based on the proportion of dominance is similar to the weighted nearest neighbor approach, we demonstrate that knowing the distribution of ρ_k over all the classes gives us the notion of ambiguity. We will discuss this in Section 6. Fig. 10 shows a simple validation test of the above procedure.

5.2. Prediction from Incomplete Trajectory

We now assume that we are given an online trajectory of unknown length. Therefore, we have to predict the adequate scale for the incomplete track at time τ.

Let us first denote the function $\phi(x, \alpha)$, which scales only the third component (t axis) of the input point x by scalar α. We then evaluate Eq. 4 as:

$$\mathcal{L}_k^*(\mathrm{T}_\tau) = \frac{1}{\tau}\sum_{i=1}^{\tau}\left[\chi(\phi(x_i, \tfrac{l}{\tau+j}), \phi(y_i, \tfrac{l}{\tau+j}))\Big|_{\mathbf{X}_k}\right] \quad (7)$$

by varying k and j, where the l is a normalized length used for constructing GPRFs. If we take the j^* and k^*, which maximize the $\mathcal{L}_k^*$, the selected j^* gives us the right scale for the time axis, and $k*$ is an index of the chosen GPRF. Then, the percentage of local selection for $i = 1 : \tau$ can be calculated by counting c_k^* from the following function:

$$l^*(i) = \arg\max_{k\in[1,N]}\left[\chi(\phi(x_i, \tfrac{l}{\tau+j^*}), \phi(y_i, \tfrac{l}{\tau+j^*}))\Big|_{\mathbf{X}_k}\right] \tag{8}$$

From Eq. 6, we can then measure the global similarity as: $\mathcal{S}_k^*(\mathrm{T}_\tau) = \rho_k^*(\mathrm{T}_\tau)\mathcal{L}_k^*(\mathrm{T}_\tau)$. An incomplete trajectory often contains few vectors. Its similarity measure is thus incorrectly computed from a small number of earlier observations. To avoid this, we weight each term in $\mathcal{L}_k^*$ and each $l^*(i)$ in computing c_k^* to give larger weights to the testing vectors close to the current position at τ.

6. Anomaly Detection

Inspired by previous work on anomaly detection [3, 14, 6], we first define the properties of anomalous events in traffic motion trajectories and deal with the events from an unsupervised perspective while the normal patterns are labeled [6]. The properties of anomalous events we assume in this work (and our criteria to detect anomalies under the constraints) are the following:

Unlikelihood against normal patterns : Obviously, an anomalous trajectory is not likely to be learned from normal patterns (i.e. wrong direction, undefined pattern etc.).
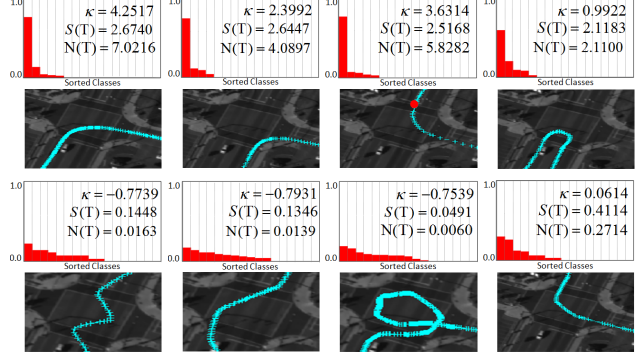

Figure 7. **Kurtosis and Normality examples**: Graphs in the first and third rows show the distribution of sorted $\rho_k(\mathrm{T}_n)$ of some examples. Images in the second and the fourth row are the corresponding trajectories. In each graph, y–axis denotes the proportion of dominance (0.0 to 1.0), and x–axis shows grids of 17 different classes (shown in Fig. 1). The leftmost bar in each graph shows a dominance level of the chosen class. Listed with each graph are the excess kurtosis κ, global similarity $\mathcal{S}_k(\mathrm{T}_n)$, and normality value $\mathrm{N}_k(\mathrm{T}_n)$. Our method classified the examples in the upper row as normal and lower ones as anomalous trajectories. Notice that the third example in the upper row models a car which *stopped* for a while at the red light.

Our solution is to measure the global similarity of tracks to learned normal classes (Eq. 6), and see how much the similarity of the chosen class is far from the similarity from normal trajectories. Let us first denote the selected class as k, and its maximum global similarity as $\mathcal{S}_k(\mathrm{T}_n)$.

Ambiguity : Subsequences of a pattern sometimes fluctuate between normal patterns (i.e. zigzag patterns) or change from one class to another class. Even though we are not certain whether the subsequence goes into the category of anomaly or not (since it was not labeled), the fluctuation and different selection of subsequences could increase the ambiguity for decision.

As discussed in section. 5.1, the proportion of dominance value $\rho_k(\mathrm{T}_n)$ effectively reflects this tendency. A distribution of $\rho_k(\mathrm{T}_n)$ with a broader spread represents a case of ambiguity in which there is no one dominant class. We interpret this as one of our criteria to detect anomalous events.

To verify this, we first sort the dominant proportion values $\rho_k(\mathrm{T}_n)$ ($k = 1 : N$) in a non-increasing order. We then measure the *excess kurtosis* of this distribution. The excess kurtosis is calculated from $\kappa = \frac{\mu_4}{\sigma^4} - 3$, where μ_4 is the fourth moment of the mean from sorted ρ_k, and σ is the standard deviation. In any case, $-2 < \kappa < \infty$. If $\kappa < 0$, the distribution of the sorted ρ becomes broader and the peak (dominance) becomes smaller. If $\kappa = 0$, the distribution is almost similar to normal distribution. We assume a trajectory with $\kappa \geq 0$ as more likely to be a normal.

Using the criteria above, we define the normality function N such that the online trajectory T_n to be a class k as:
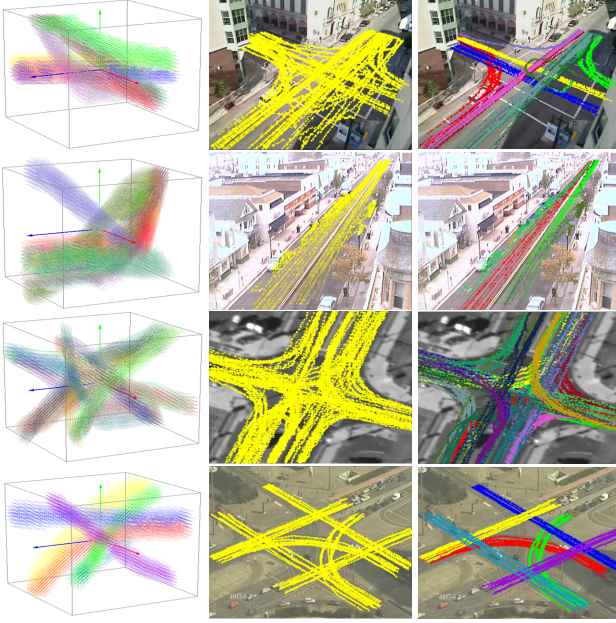
Figure 8. **Datasets and each of their GPRFs :** Example images from the videos used in this paper, and their learned GPRFs are shown. For each row, first column shows a normalized mean flows constructed from some of trajectories in each data set, and the second column shows testing trajectories, and the last column shows testing trajectories labeled in different colors for testing purpose.

Set	Data type	#GP	frame rate	ANO
Adam [22]	Intersection	8	10–30	Y
Ocean [13]	St.Road	11	5	Y
UCF [1]	Intersection	6	30	N
CLIF [19]	Intersection	17	varying	Y

Table 1. **Dataset description:** #GP refers to the number of trained classes (different types of straight, left-right turns, u-turns, stop-and-go, and parking with many of 2nd order movements) and ANO refers if there is anomalous sample exist in the data or not. For non-anomaly data set, we only evaluate the similarity.

$$N_k(T_n) = \frac{1}{2}\left[(\kappa + 2)\mathcal{S}_k(T_n)\right] \qquad (9)$$

Using the normality function, we first define m–trajectories, $T_{n(i)}^i$ ($i \in [1, m]$), used for training the class k, and the γ_k as the minimum normality value of evaluating all the m–trajectories to the class k. The decision criterion for the anomalous trajectory is then defined as $N_k(T_n) < 0.5\gamma_k$.

Fig. 7 shows examples of the normalcy test using some of our trajectory data using learned GPRF classes shown earlier in Fig. 1.

7. Results and Evaluation

We evaluate our method on four different video data sets taken from different locations (Fig. 8, and Table. 1). Since CLIF data does not have a sufficient number of trajectories, we added simulated trajectories generated from our traffic simulation software. Each data set has different frame rates

	GPRF		DDTW	
Set	Accuracy	Precision	Accuracy	Precision
Adam	0.9955	0.9843	0.8928	0.7296
Ocean	0.966	0.8782	0.9488	0.7424
UCF	1.0000	1.0000	1.0000	1.0000
CLIF	0.9929	0.9488	0.9881	0.9274

Table 2. **Trajectory recognition using complete trajectories :** Each value denotes the average of each class. Our proposed approach works better in most of datasets except UCF data, which has small number of very distinctive motion patterns. See Fig. 8

and durations. Since our approach generating GPRF is not affine invariant, we rectified all the video sequences and used their trajectories in a canonical view (top-view). Each data set has more than 100 trajectories, which are labeled for testing purposes (Fig. 8, third column), and all have anomalous trajectories, except the UCF data. The UCF data set has a relatively shorter duration compared to the other data sets, and does not contain anomalous events. Each data set is divided into a training set, from which GPRFs are trained, and a test set for evaluation.

Similarity measurement of complete trajectory: We empirically evaluate the effectiveness of our method in trajectory classification by comparing to a well-known trajectory matching algorithm. We consider the Derivative Direct Time Warping (DDTW) [7] method, because of its robustness in higher dimensional signals. For a fair comparison, we do not consider anomalous events. Table. 2 enumerates the average accuracy and precision of all classes using complete trajectories.

To evaluate the effectiveness in discriminating a specific pattern of trajectory from different patterns, we measure the global likelihood and the global similarity of randomly chosen trajectories from each trained class. Fig. 10 shows a comparison of the similarity measurements. Our global similarity measurement dramatically discriminates the input trajectories from ones belonging to incorrect classes and even from anomalous tracks.

Similarity measurement of incomplete trajectory : We evaluate our method on incomplete trajectories both quantitatively and qualitatively. The graph in Fig. 9 shows the quantitative result. Our approach works better in most of the data sets even with short duration of sequences from the input trajectory. Fig. 12 demonstrates how our approach works in incomplete trajectories from online video.

Notice that the examples in Fig. 12 A and B describe how our approach can model and recognize complex motions such as deceleration and stopping situation, which is not possible in spatial proximity based approaches.

Evaluation of anomaly detection Fig. 11 shows ROC curves for each data set. Except the results from the Ocean data, other results show that TPR is more than 90% with less than 20% of FPR. As shown in Table 1, the Ocean data
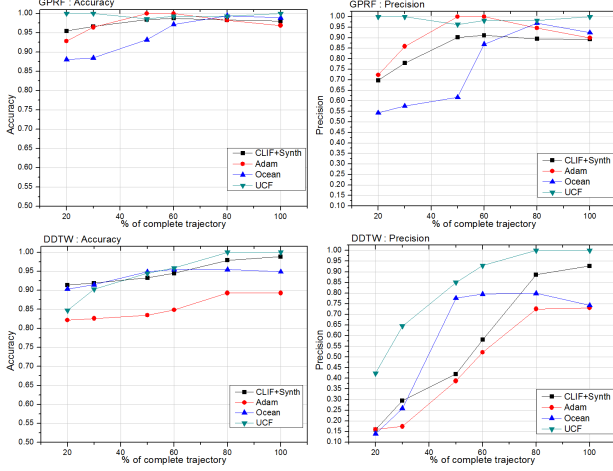
Figure 9. **Trajectory recognition using incomplete trajectories :** For each graph, x–axis indicates the percentage of sub-tracks used for evaluation (i.e. 50% means by the first 50% of total length of testing trajectory). (**Upper row**) Results from our approach, and (**Bottom row**) Results from DDTW
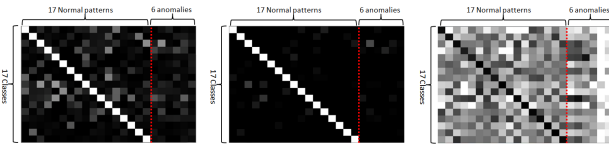

Figure 10. **Similarity validation test :** In each image, the row indicates learned 17 classes, and the first 17 columns are the trajectories which are randomly chosen from each of labeled sets (for the visualization purpose, we arranged each trajectory corresponding to the order of classes). The last 6 columns consist of randomly chosen anomalous trajectories from our anomalous sets. Note that all values are normalized from 0 to 255 by the maximum similarity of each classes. (Left) Using global likelihood (Eq. 4), (Middle) Using Global Similarity (Eq. 6). (Right) Minimum distances from DDTW : a darker color represents a better score.
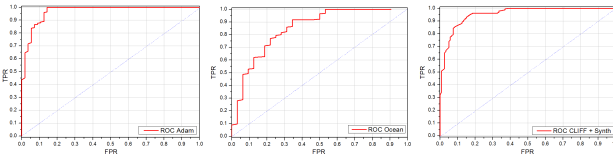

Figure 11. **ROC curves for anomaly detection :** x axis refers to False Positive Rate (FPR), and y axis for True Positive Rate (TPR), we evaluate each curve by varying a threshold on each normality function (Eq.9). (**Left**) : Adam, (**Middle**) : Ocean, (**Right**) : CLIF

has very low frame rate, thus, its actual online trajectory is too sparse. Therefore, the kurtosis measurements were too sensitive, since the distribution were constructed from too few samples. However, it still gives us 80% of TPR with 30% FPR. Fig. 13 demonstrates that our method effectively detects anomalous events in incomplete trajectories.

8. Conclusions, Limitations and Future work

In this paper, we introduce a new framework for modeling motion trajectories in the spatio-temporal domain using flow fields generated from a GPRF. Our experiments demonstrate that our approach can recognize motion patterns from both complete and incomplete tracks even with GPRFs trained from a minimal number of labeled samples.

There are some limitations and avenues of improvements. First, our approach assumes that the types of normal patterns are defined *a priori*. Secondly, if the trajectory is too sparse, the kurtosis measurements in earlier stages of the track can be unstable due to the sensitivity of kurtosis in a sparse distribution. Finally, our approach does not recognize traffic jams and such patterns would be detected as an anomaly. A traffic jam, however, can be represented as a *step* shape function in the normalized (u, v, t) coordinates, and we plan to use this method to model traffic jams.

References

[1] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *Proc' of ECCV*, pages 1–14, 2008. 6

[2] Y. Altun, T. Hofmann, and A. J. Smola. Gaussian process classification for segmenting and annotating sequences. In *Proc' of ICML*, pages 4–9, 2004. 2

[3] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, pages 1–15, 2009. 5

[4] W. Chu and Z. Ghahramani. Preference learning with gaussian processes. In *Proc' of the ICML*, 2005. 3

[5] D. Ellis, E. Sommerlade, and I. Reid. Modelling pedestrian trajectories with gaussian processes. In *International Workshop on Visual Surveillance*, pages 1229–1234, 2009. 2

[6] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:1450–1464, 2006. 1, 2, 5

[7] E. J. Keogh and M. J. Pazzani. Derivative Dynamic Time Warping. In *SIAM SDM'01*, 2001. 6

[8] S. Khalid and A. Naftel. Classifying spatiotemporal object trajectories using unsupervised learning of basis function coefficients. In *Proc' of the third ACM international workshop on VSSN*, pages 45–52, 2005. 2

[9] X. Li, W. Hu, and W. Hu. A coarse-to-fine strategy for vehicle motion trajectory clustering. In *Proc' of the ICPR '06*, pages 591–594, 2006. 2

[10] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proc' s of ACM SIGMOD workshop on data mining and knowledge discovery*, 2003. 1, 2, 3

[11] B. Morris and M. Trivedi. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proc'IEEE CVPR '09*, pages 1–8, 2009. 1, 2

[12] J. Nocedal and S. Wright. Numerical Optimization, Series in Operations Research and Financial Engineering, 2006. 3

[13] S. Oh and A. Hoogs. Unsupervised learning of activities in video using scene context. In *Proc' of ICPR*, pages 3579–3582, 2010. 6
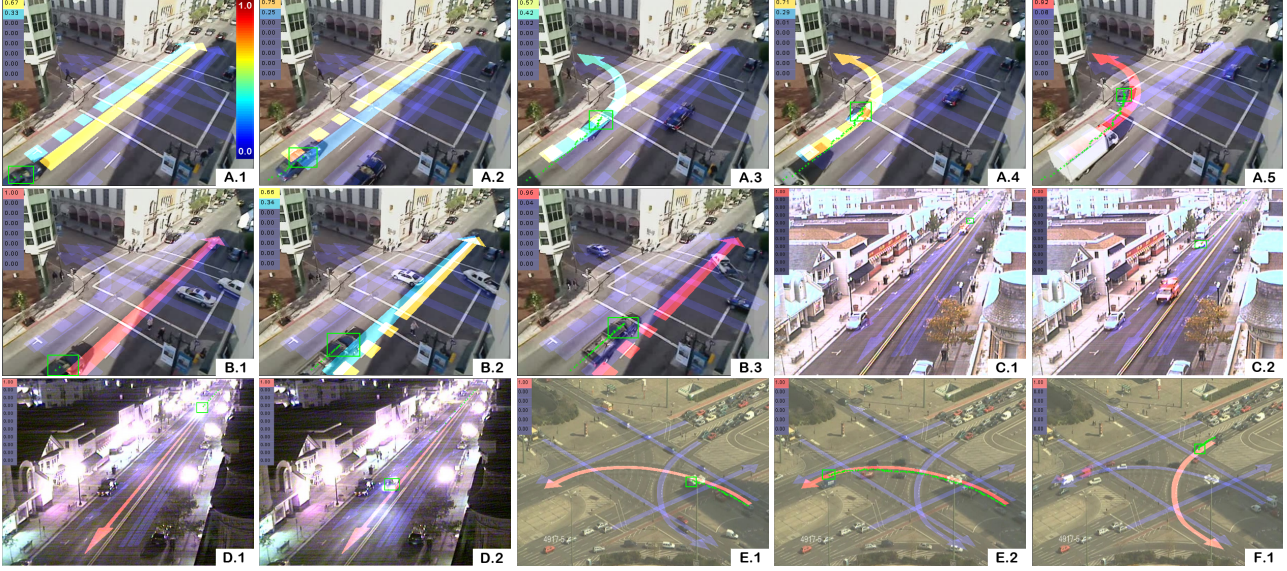
Figure 12. **Qualitative evaluation of trajectory prediction :** Each arrow indicates a possible pattern of motion on the road. The color of each arrow indicates the probability level to be classified as the given pattern (See the color table in A.1). Dotted lines indicate the "stop (includes deceleration) and move" **A.1**. A car moves straight ahead with a probability of 67% for continuing to move straight. **A.2** As the car decelerates, a pattern for stopping situation becomes dominant. **A.3** The car slightly changes its direction to left. Because it needs to decelerate before a left turn, the probability of taking the left turn increases. **A.4** The left turn becomes dominant. **A.5** Finally, even before the car enters the left road, the probability for a left turn increases to 92%. **B.1** A car originally moves ahead in the second lane with the probability of 100% for continuing that motion pattern. **B.2** The car is slowing down. **B.3** Finally the car stopped for a traffic signal. **C.1** A car moves ahead to bottom left, but decreases its speed. Therefore even the spatial pattern is similar to straight, the patterns for parking becomes dominant **C.2** Finally the car parks. **D.1–2** Though spatial movement is similar to the examples shown in C, due to its velocity and acceleration, it was proven to be straight pattern. **E.1–2** and **F.1** Examples of prediction in UCF data
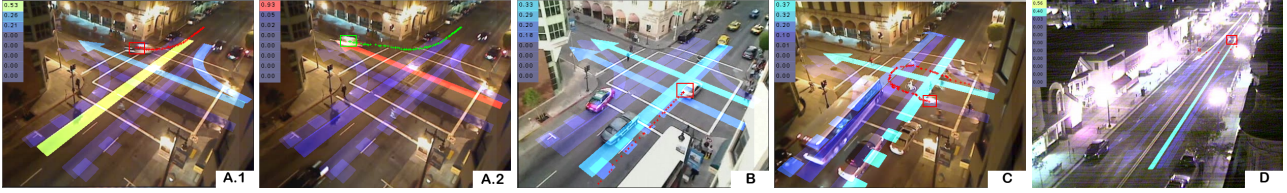

Figure 13. **Qualitative evaluation of incremental anomaly detection:** The color of the trajectory indicates the normalcy (green), and anomaly (red) **A.1** A police car goes in the wrong direction on a one-way road, and is classified as an anomalous pattern. **A.2** As the car merges to left direction, the trajectory becomes normal. **B** A car was suddenly stopped inside the intersection due to a pedestrian, and we classify it as an anomaly. **C** Bikes are moving inside the intersection. **D** A car takes a U-turn which is not allowed in the street.

[14] C. Piciarelli and G. L. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recogn. Lett.*, 27:1835–1842, 2006. 1, 5

[15] L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*. 1993. 3

[16] C. Rasmussen. Gaussian processes in machine learning. *Advanced Lectures on Machine Learning*. 3

[17] V. C. Raykar and R. Duraiswami. Fast large scale gaussian process regression using approximate matrix-vector products, 2007. 2

[18] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Proc' of CVPR*, pages 238–245, 2006. 2

[19] US AIR FORCE. CLIF https://www.sdms.afrl.af.mil/. 6

[20] M. Vlachos, D. Gunopoulos, and G. Kollios. Discovering similar multidimensional trajectories. In *Proc' ICDE*, pages 673–680, 2002. 1, 3

[21] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30:283–298, 2008. 2

[22] WWW. http://www.justin.tv/adamsblock. 6

[23] J. Xiao, C. Yang, F. Han, and H. Cheng. Multimodal technologies for perception of humans. In *Vehicle and Person Tracking in Aerial Videos*, pages 203–214, 2008. 2

[24] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan. Detecting time series motifs under uniform scaling. In *Proc' of ACM SIGKDD '07*, pages 844–853, 2007. 1

[25] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proc' of ICPR '06*, pages 1135–1138, 2006. 1, 2