Distribution Driven Extraction and Tracking of Features for Time-varying Data Analysis

Soumya Dutta and Han-Wei Shen

Abstract—Effective analysis of features in time-varying data is essential in numerous scientific applications. Feature extraction and tracking are two important tasks scientists rely upon to get insights about the dynamic nature of the large scale time-varying data. However, often the complexity of the scientific phenomena only allows scientists to vaguely define their feature of interest. Furthermore, such features can have varying motion patterns and dynamic evolution over time. As a result, automatic extraction and tracking of features becomes a non-trivial task. In this work, we investigate these issues and propose a distribution driven approach which allows us to construct novel algorithms for reliable feature extraction and tracking with high confidence in the absence of accurate feature definition. We exploit two key properties of an object, motion and similarity to the target feature, and fuse the information gained from them to generate a robust feature-aware classification field at every time step. Tracking of features is done using such classified fields which enhances the accuracy and robustness of the proposed algorithm. The efficacy of our method is demonstrated by successfully applying it on several scientific data sets containing a wide range of dynamic time-varying features.

Index Terms—Gaussian mixture model (GMM), Incremental learning, Feature extraction and tracking, Time-varying data analysis

1 Introduction

In the era of big data analytics, effective exploration of time-varying data poses a significant challenge to the data scientists. Since experts from diverse fields are interested in a wide range of phenomena, defined as features, efficient detection and tracking of such features is an essential task in temporal data understanding. A key component in such analysis is the ability to accurately classify the large scale data based on the expert's interest. A visual exploration with a focus on the relevant data allows domain scientists to quickly make crucial decisions about the important scientific problems.

However, owing to the ever increasing complexity of scientific phenomena, precise definition of a feature (i.e. the region of interest) is often unavailable. Features such as the eye of a storm system, circulating vortex cores in a flow field, rapidly propagating earthquake shock-waves are hard to be separated by specific threshold values [2]. Therefore, the tracking algorithms which rely upon predetermined feature descriptors, are not readily applicable in these scenarios. Scientists need visualization systems where they can directly interact with the data and locate the feature of interest based on the initial vague hypotheses. But, repeating this process manually for a large time-varying data is tedious and impractical.

Majority of the tracking algorithms proposed in the past [5, 16, 24, 29, 30, 34, 35, 43, 44] have a general assumption that the definition of the feature is predetermined and hence the feature extraction process is deterministic. Therefore, given only a fuzzy feature description, automatic detection and tracking of such regions requires novel algorithmic approaches. A key requirement of such algorithms to be considered as practical is to have the ability to quickly adapt to a refined/new feature description without going through the entire raw data again. Also the scientific data contains features which can undergo rapid changes over both space and time and usually do not maintain any specific structure. Therefore, tracking such a region requires robust techniques which can efficiently capture its dynamic nature and be able to detect it in consequent time steps.

In the absence of precise target definition, use of statistical theory based approaches have shown promising results in the recent past [4, 14]. Analysis using probability distributions has become an emerging trend and numerous visualization problems have benefited from such stochastic approaches [11, 17, 21]. In this work, we use probability distribution functions as a measure of feature definition given a user highlighted region in the data. Since features in scientific data sets demonstrate properties like deformation and non-rigidity, use of distributions to represent such features adds great flexibility in our tracking algorithm. We exploit both temporal and spatial coherency of data to build a novel distribution driven feature tracking algorithm. The key observation here is that a tracking algorithm needs to account for the two key types of information:

1. possibility of the presence of motion at a specific region which might indicate existence of a potential feature.
2. possibility of the existence of the feature at a specific region given a signature of the target feature.

Here, the term possibility reflects the *degree of belief* of certain event. Note that the motion information can be inferred by modeling the temporal dynamics of the data, while estimating the second possibility measure requires classification of data domain into spatially coherent regions that match the target definition. While none of these information mentioned above independently can give accurate results, a combination of them however yields an algorithm which works well for the extraction and tracking of features without a precise feature definition.

In order to efficiently capture the temporal dynamics, the proposed algorithm divides the data into blocks and employs an *incremental learning scheme* for modeling the continuous time-varying distributions of data at each block in the form of Gaussian Mixture Models (GMM) [31, 36]. We estimate both the feature's location and its motion using distributions and classify the data domain into a feature-aware space. To measure the existence of a moving object through a local region, we employ a foreground estimation algorithm which helps us to quantify the first possibility measure stated above. Given a target region of interest, we model it as a GMM and then estimate the possibility of each data block containing the target which allows us to compute the second possibility measure. Finally they are combined to generate a feature-aware classification field where high possibility valued regions are representative of the feature's location. Applying a threshold on the possibility values based on user's requirement, we are able to segment the classification field and focus on the feature. Tracking fuzzy features using the classification fields enhances the robustness of our algorithm since such fields inherently encapsulate the spatiotemporal data dynamics and allow us to analyze the feature probabilistically. Therefore our contributions in this work are threefold:

- *Soumya Dutta is with the GRAVITY group, The Ohio State University. E-mail: dutta.33@osu.edu.*
- *Han-Wei Shen is with the GRAVITY group, The Ohio State University. E-mail: hwshen@cse.ohio-state.edu*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication 20 Aug. 2015; date of current version 25 Oct. 2015. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org. Digital Object Identifier no. 10.1109/TVCG.2015.2467436

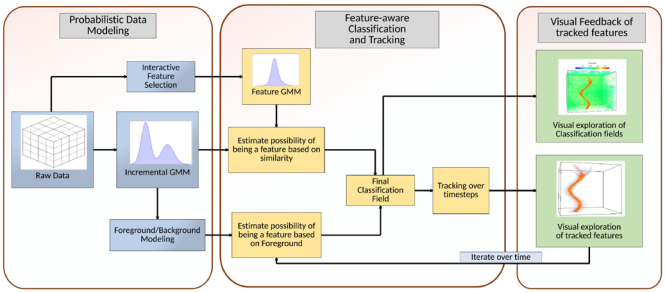


Figure 1: A schematic diagram of the proposed method.

1. We take advantage of the incremental learning scheme of GMMs for modeling efficient and compact temporal data distributions and employ a foreground modeling to detect the existence of motion in spatial domain.
2. We propose a new algorithm which models features as a GMM and reconstructs a feature-aware classification field where the regions with high possibility values highlight the target feature.
3. Finally, we present a tracking algorithm using the classification fields and visualize the evolution of time-varying features using volume visualization techniques.

This paper is organized as follows: In Section 2 we present a comparative discussion of the related research works to the topic of this paper. A brief overview of our system is provided in Section 3. Section 4 presents our data modeling scheme and in Section 5 and Section 6 the proposed algorithm is described in detail. We show the results obtained by our method in Section 7 followed by parameter study in Section 8 and a discussion in Section 9. We conclude our work in Section 10 by highlighting several possible future directions.

2 Related Works

Distributions in visualization. Use of probability distribution functions to deal with scientific problems in visualization has become a promising approach. For answering arbitrary range distributions from the data, Chaudhuri et al. [4] proposed an integral distribution based method. For analyzing local statistical properties of the data Lee et al. [19] used integral distributions with discrete wavelet transformation. To leverage the performance of query driven visualization, Gosink et al. [11] used distribution functions effectively. Jhonson and Huang [17] allowed querying on distributions for enhancing scientific data understanding. Local histograms were used for designing efficient transfer functions for scientific data sets [21]. For compact data representations and data classification mixture of Gaussians have gained popularity in recent years. A Gaussian mixture model based volume visualization was proposed in [20]. GMMs were also used for probabilistic transfer function designing [23]. A block-wise approach was taken by Gu and Wang for a hierarchical graph based analysis of time-varying data [12]. For coherent transfer function design for time-varying data, incremental Gaussian mixtures were also used in [42].

Feature extraction and tracking in visualization. Feature extraction and tracking is an important problem for scientific data visualization and has been explored in the past. For tracking volumetric features in scientific data Samataney et al. [29] proposed a correspondence based approach. By exploiting volume overlapping, Silver and wang [34] tracked volume features with high accuracy as well. Ji and Shen used earth mover's distance to design a globally optimum feature tracking algorithm [16]. In another work, Ji et al. used higher dimensional isosurfaces for tracking volume features [15]. For tracking features in distributed AMR data sets, Chen et al. used feature tree as a visualization representation of tracked features [5]. Tzeng and Ma [39] proposed a machine learning approach for automatically learning and tracking features in large scale simulation data. Ozer et al. recently proposed techniques for tracking a group dynamic features together as a collection, where the problem of tracking was modeled as activities in scientific data [24, 25]. Using a predictor-corrector method, Muelder and Ma introduced a new algorithm for efficient feature tracking [22]. To quantify goodness in feature correspondence, Reinder et al. introduced an attribute-based feature tracking algorithm for scientific data sets [28]. In a recent work, Sauer et al. utilized particle information for enhanced feature extraction and tracking in joint particle/volume data sets [30]. Their method allowed to track features in data sets when sufficiently dense temporal sampling is not available. A TAC based distance field was used effectively in the works of Lee and Shen for analyzing time-varying features [18]. Theisel and Seidel proposed a method for tracking features like saddle, source, and sinks in time-varying vector field directly using streamlines [37]. Garth et al. in another work presented techniques for tracking vector field singularities [10]. A survey of feature tracking algorithms also can be found in [27] by Post et al.

In this work, we extend the capability of feature tracking algorithms by proposing a new distribution driven technique which enhances the feature extraction part when precise feature definition can not be obtained. Below we present a short overview of our algorithm before going into the details.

3 Method Overview

Our high level goal in this work is to devise an efficient algorithm capable of tracking features in large scale data when precise feature definitions are not available. We use mixture of Gaussians (GMMs) to model the feature and employ a distribution driven technique for extraction and tracking of such features. We model the data block-wise and store distributions in form of GMMs for each block. Figure 1 presents a schematic diagram of the proposed system. Initially, given a region in the data as the feature of interest, we construct the feature GMM using the data from the selected region. We also construct distributions for all the blocks of data by incrementally learning the parameters of the GMMs and quantify the possible existence of a moving object in a block by adapting a foreground estimation algorithm. Next, we compute the chance of a block being part of the feature by comparing the distribution of the block with the feature GMM directly by exploiting the spatial coherency of the data. To measure the final possibility of the block as a part of the feature, we combine the two types of estimated information to construct a *feature-aware classification field* where high valued regions highlight user interested features. Finally, we demonstrate an automatic tracking technique using classification fields and explore the evolution of tracked features by interactive volume visualization techniques.

4 Stochastic Data Modeling and Incremental Estimation of Motion Information

In this section, we introduce the details regarding the initial feature selection technique and modeling of temporal data distributions. Since the size of time-varying data can be very large, we aim at a compact representation of distributions with a fast computation time and small memory footprint. This allows us to accelerate the tracking while users refine or specify new features since we only need to access the already computed summarized data without touching the entire raw data. Popular distribution estimator histogram computes the distributions quickly, but its storage requirement makes it unsuitable for our work. Another non-parametric estimator Kernel Density Estimation also requires high storage and this method is computationally expensive. So, to meet the requirements in our work, mixture of Gaussians (GMM) presents a suitable choice for modeling distributions. Use of GMMs [1] is well known for data classification [20, 23, 42]. Since multiple Gaussians are used to model the data, no assumptions about the underlying data distribution are made [38]. Furthermore, based on the Gaussian properties, GMMs allow efficient computation by directly using the mixture components [38]. Below we first describe how the experts select their region of interest and then formally present the proposed distribution driven model in detail.

4.1 Interactive Feature Selection

Since scientists may not always have a precise definition of the feature, we allow them to pick their region of interest directly in the volume space from an initial time step. Scientists can explore the data and based on their experience and knowledge, they highlight a region interactively, where they hypothesize the feature of interest exists. In this way, they are not required to define a hard threshold for the feature which otherwise is a difficult task for complex scientific features. In a previous work, selection of relevant data directly from volume space has shown to be effective over selection in histogram domain [23]. A selected region provides us the sample points which allow us to model the region as a GMM. Representing the feature using a mixture of Gaussians provides us a basis for quantifying the vagueness in feature definition to a statistically meaningful representation. Modeling features as a distribution is not new and scientists have used this approach in the past [7, 9, 21]. In fact, since the features in the scientific data is usually non-rigid objects, often without any definite shape, modeling them using distribution presents a suitable choice [9].

4.2 Incremental Gaussian Mixture Model for Distribution Estimation of Time-varying Data

In this work, we aim to exploit the inherent spatiotemporal coherency of time-varying data to achieve increased accuracy and robustness in our algorithm. Keeping the large size of time-varying data in mind, a block-wise partition of data for analysis is adapted. The whole data space is partitioned into smaller non-overlapping blocks. Such a block-wise approach is widely employed in computer vision and video processing applications for exploiting the spatial coherency in data at a reduced computational complexity. Also it was previously shown that, for scientific data sets, a block-wise approach is more suitable than a voxel-wise approach when the data size becomes large [41]. Furthermore, for capturing the temporal coherency in data modeling, we advocate for an incremental scheme for estimating temporal data distributions. Formally, the probability density $p(X)$ of a GMM for a random variable X is expressed as:

$$p(X) = \sum_{i=1}^{K} \omega_i * \mathcal{N}(X|\mu_i, \sigma_i) \tag{1}$$

where K is the number of Gaussian components. ω_i, μ_i and, σ_i are the weight, mean, and standard deviation for the i^{th} Gaussian component respectively. It is to be noted that the sum of weights in the mixture, $\sum_{i=1}^{K} \omega_i$ is always equal to 1. Computation of parameters for the GMMs is typically done by *Expectation Maximization* (EM) which uses an iterative approach to maximize a likelihood function [1]. However, since we want to model the temporal dynamics of the data, an *incremental learning scheme* is preferred, which leads to stable distribution estimation for time-varying data [31, 36]. Use of an incremental modeling not only makes our computation faster, but also permits us to adapt a foreground estimation model which is built using the incremental GMM. Next, we present the details of the incremental update scheme for GMMs.

Incremental update scheme for GMMs. We employ an incremental update method presented in [36] for estimating the parameters of the GMMs for each block of data over time. Since scientific data generally presents temporal coherency, modeling data distribution using an incremental algorithm yields promising results as was reported in [42]. Such a modeling in turn increases the efficacy of the tracking algorithm because it is able to preserve the temporal coherence in the estimated GMMs between consecutive time steps. For the initial estimation of the parameters, we apply the off-line EM algorithm per block for the first time step only. Then, from the next time step onwards, we update the parameters of the GMMs incrementally for each block as we observe new data.

For each block, every new data point is checked against the existing K Gaussians. A positive match is found if a data point lies within the 2.5 standard deviation of a Gaussian. If multiple matches are found, then the best matched Gaussian is selected, which is the Gaussian with the minimum matched value. If none of the K Gaussians match the current data value, then the least probable distribution in the model is replaced with a new Gaussian with the current data value as the mean, an initial high standard deviation, and a low weight. The weights at time t for the i^{th} mixture are adjusted as:

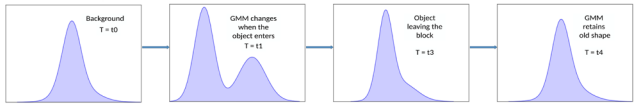


Figure 2: Evolution of the GMM of a block while an object moves through it.

$$\omega_{i,t} = (1-\beta)\omega_{i,t-1} + \beta(I_{i,t}), \;\; i \in \{1,2...K\} \tag{2}$$

β is called the learning rate and the value of $I_{i,t}$ is 1 for the distribution with the best match and 0 otherwise. After the adjustment, all the weights are normalized again for maintaining consistency. The μ and σ parameters for the unmatched distributions remain the same, however for the matched distribution they are updated as:

$$\mu_{i,t} = (1-\beta)\mu_{i,t-1} + \beta\mu_{i,t} \tag{3}$$

$$\sigma^2_{i,t} = (1-\beta)\sigma^2_{i,t-1} + \beta(\mu_{i,t} - x_{i,t})^2 \tag{4}$$

Once we have observed all the points for a block in the current time step, the GMM will give us the updated distribution for the current time step. It is evident that the model adapts to the new data since it adds or removes Gaussians from the existing model as required. While learning the distributions for each block, we also estimate the possibility of each block having a moving object in it. For measuring such possibility, we manipulate the distributions learned by the incremental GMM and employ a foreground estimate to extract the desired information as described below.

4.3 Estimation of Moving Features Using Foreground Detection

In a time-varying data, if a feature has a motion, then by exploiting such motion information, the location of the feature can be identified efficiently. If the moving feature enters a block which was not present there in the previous time step, the block will encounter new data points which will result in a creation of new GMMs during the distribution estimation. If the new data points change the block's distribution significantly compared to its previous state, then such blocks can be characterized as containing the moving feature in the current time step. Identifying those blocks will give us the feature's possible location in space.

Data blocks containing such moving feature are often interpreted as the foreground region in the time-varying data, which demonstrate distinguishable properties compared to the relatively static background region. The possibility of a block being part of the foreground can be estimated by the amount of new data points the block has encountered in the current time step. If majority of the points are new, then the block should be classified as a foreground with high confidence. Since, we model the temporal distribution of the data using an incremental update scheme, at any given instant, the GMMs at each block represent a temporal distribution of the block created using the data observed in earlier time steps. The Gaussians with higher weights in the block GMM are the representative of the portion of the data which the block has encountered consistently in the past and the high weights reflect that. Therefore, when a new data point comes, it will not find a match with any of such Gaussians and will add a new Gaussian in the model. We aim to identify those new data points and by doing so we can quantify the possibility of the block being part of the foreground. While observing new data points during the distribution estimation, we keep track of all the data points that: (1) do not match any existing Gaussians with weight higher than a threshold T, and (2) matches with a newly created Gaussian. All such points represent the new data points that the block has observed in the current time step. As the

number of such points increase, the chance of that block of being a foreground also increases. So, the possibility that a block containing a foreground object is quantified by the fraction of the new data points to the total number of observations in the block:

$$POS_{foreground,t}(b_{i,t}) = q_{i,t}/n_{i,t} \quad (5)$$

where $q_{i,t}$ is the number of observations that satisfies either the clause (1) or (2) stated above, and $n_{i,t}$ is the total number of observations for the i^{th} block at time t. The value of $POS_{foreground,t}(b_{i,t})$ is always between 0 and 1, where $POS_{foreground,t}(b_{i,t}) = 1$ signifies no data points in current time step matched any existing Gaussians and thus chance of the block being a foreground is maximized. As we iterate over all the time steps, we measure this possibility value for each block per time step and keep this information. Later we will use this information and combine it with another possibility measure for the final classification of each block as being part of a feature of interest.

In Figure 2 we show the conceptual evolution of a GMM of a block over a sequence of time steps as an object moves through it. At time $t0$ the block is considered as a part of the background. However, as the object enters the block at time $t1$, the distribution changes and the possibility value of this block being a foreground increases. From $t1$ - $t3$ the block shows evidence of being a part of a foreground object and finally when the object exits the block, the GMM returns back to its old shape, as can be seen at time $t4$.

For each time step, we store the estimated parameters of the GMM and also a possibility value for each block measured by Equation 5. Observe that the estimation of the possibility value presented in this section is oblivious to the feature. But this provides us a way to measure the chance of a block being part of a moving object which can be a potential feature. In the next section, we introduce another possibility measure for a block being part of a feature when we observe the feature distribution. Finally, we show how the two possibility values are combined to construct a feature-aware classification field where the regions with high possibility values indicate the existence of features.

5 Feature-Aware Classification Fields

Previous section introduces a measure for each data block which gives the information regarding the chance of existence of a moving object in a block. For any robust feature extraction and tracking system, detection of motion component is essential for improving the tracking results [26]. However, since this information does not consider the target feature definition, the extracted regions may require further refinement based on user's need. Also, if the feature does not have a strong motion component then we can not make any definitive conclusion about the feature from only foreground information. To remedy this, we introduce another measure which estimates the possibility of a block being part of a feature by observing the feature distribution and helps us to finally classify the blocks.

5.1 Classification Based on Feature Similarity

Our goal is to measure the possibility of each block being a part of the target feature. This possibility measure exploits the spatial coherency and extracts the regions which contain similar distributions as the target GMM. We measure the similarity between the GMM of each block and the feature GMM. There are several techniques available for measuring the similarity between two GMMs such as the Kullback-Liebler divergence (SKL) and, Bhattacharyya-based distance measures [33, 45]. However, SKL does not have a closed form solution and needs Monte-Carlo approximation which makes it computationally expensive. Also it was reported that the Bhattacharyya-based similarity measure is generally fast and leads to good results [33]. So, for measuring the similarity between GMMs, we have used the Bhattacharyya-based distance measure which can be expressed as:

$$\Psi(p,p') = \sum_{i=1}^{n}\sum_{j=1}^{m} \omega_i \omega'_j \mathscr{B}(p_i, p'_j) \quad (6)$$

where p and p' are the GMMs and n and m are the number of mixture components of GMM p and p' respectively. $\mathscr{B}$ is the Bhattacharyya distance between two Gaussian kernels and is defined as:

$$\mathscr{B}(p,p') = \frac{1}{8}(\mu-\mu')^T \left(\frac{\Sigma+\Sigma'}{2}\right)^{-1} (\mu-\mu')^T + \frac{1}{2} ln \left[\frac{|\frac{\Sigma+\Sigma'}{2}|}{\sqrt{|\Sigma||\Sigma'|}}\right] \quad (7)$$

here μ, μ' and Σ, Σ' are the mean and covariance of the Gaussian kernels p, p' respectively. After computing the values of $\Psi(\cdot)$ for all the blocks, the values are normalized. Given the feature GMM f_t at time t, the possibility of i^{th} block $b_{i,t}$ being part of the feature at time t is computed as:

$$POS_{similarity,t}(b_{i,t}) = 1 - \Psi_{norm}(b_{i,t}, f_t) \quad (8)$$

Note that the value of $POS_{similarity,t}(b_{i,t})$ is always between 0 and 1 and is maximum for the block which matched best with the feature GMM f_t and as the degree of match reduces i.e. the similarity between feature GMM and block GMM decreases, the value of $POS_{similarity,t}(b_{i,t})$ also drops.

So far, we have described two types of possibility values for each block and each of them tries to classify a block of being part of a feature. In the following, we demonstrate how these two information are combined effectively to obtain a more accurate classification of all the blocks instead of using them individually.

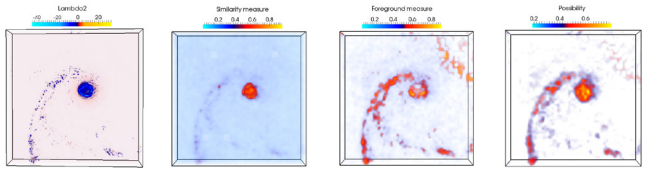
5.2 Construction of Feature-Aware Classification Fields by Combining Multiple Possibility Values

In this section, we present the method for combining the possibility values discussed in earlier sections to achieve a final robust classification of all the data blocks. Such a classification will assign higher values to the blocks which are more probable of being part of the target feature. Note that the possibility values defined earlier, tries to analyze the block from different perspectives. For the first method, high value of $POS_{foreground,t}(b_{i,t})$ for a block signifies that there is a high chance of existence of a feature in that block. However, since it does not directly consider the target feature definition, we can not come to a certain conclusion by just using this measure. To complement this deficiency, we have incorporated another possibility measure $POS_{similarity,t}(b_{i,t})$ which calculates the possibility by taking into account the similarity between a block GMM and the user interested feature GMM. However, when the feature distribution is not clearly separable from the background or the feature size is sufficiently small, performance of this approach deteriorates. So, we can not always completely rely on the similarity based measure for the classification. Therefore, we seek a consensus between the two measures to classify all the data blocks with high confidence.

In statistical theory, there exists several techniques for combining multiple hypotheses for inference. In our case, we have two hypotheses (possibilities of being a feature) and they can be combined either linearly or non-linearly. A popular and effective technique for linear combination of hypotheses is presented in [6], called the *linear opinion pool*. This technique is fast to compute and suitable for interactive algorithms such as ours. Hence, following this strategy, the two possibility values are combined as:

$$POS_{feature}(b_i) = \gamma * POS_{similarity}(b_i) + (1-\gamma) * POS_{foreground}(b_i) \quad (9)$$

Here γ acts as the mixing parameter which plays an important role. The value of γ always chosen between 0 and 1 which determines how much contribution each of the possibility measures will have in the final classification. Based on the knowledge of experts, this parameter can be selected carefully to enhance the robustness of classification. If a data set contains a moving feature and scientists are interested in tracking such feature then the value of γ is chosen accordingly such that the contribution of $POS_{foreground}(b_i)$ is more in the classification and similarly if the target feature does not show a strong movement over space, we can set a high γ values to increase contribution of $POS_{similarity}(b_i)$. In the absence of specific knowledge about the feature dynamics, we can set $\gamma = 0.5$, which accounts for the equal contribution of both the measures in final classification. Once all the



(a) Lambda2 field with target feature. (b) Similarity measure of vortex feature. (c) Foregorund possibility of the feature. (d) Classified field.

Figure 3: Feature estimation exploiting spatial and temporal coherency using hurricane Isabel data at T=34.

Algorithm 1 Construct Feature-Aware Classification Field

1: Input: $GMM(feature)$, $timestep$
2: **for all** block b_i **do**
3: Compute $POS_{similarity}(b_i)$. (Equation 8)
4: Compute $POS_{feature}(b_i)$. (Equation 9)
5: Use $POS_{feature}(b_i)$ for construction of classification field.
6: **end for**

data blocks are classified and a possibility value is assigned to them, a scalar field can be constructed using the possibility values for all the points in the block. Such a field is called the *feature-aware classification field* and Algorithm 1 presents the pseudo code for constructing such field. Direct visualization of such a field can convey the information regarding the likelihood of the feature's existence at current time step. Note that this field is generated by combining the two possibility measures which are derived directly by exploiting the spatial and temporal coherency of the time-varying data. In the absence of a precise feature definition, such a classification field allows scientists to observe the evolution of the features in a time-varying data.

Figure 3 demonstrates the usefulness of having two key possibility measures, used in this work and shows why just a single measure is not sufficient. In Figure 3a the λ_2 field of hurricane Isabel data is shown for time step 34 where the vortex region and its spread is highlighted. Initially, the target feature is specified as shown in Figure 10 in time step 1. Figure 3b depicts the $POS_{similarity}(b_i)$ field where it is clearly visible that the vortex core is identified only and the smaller band of vortices are mostly missing, or identified with low confidence. However, in Figure 3c we see the $POS_{foreground}(b_i)$ field which captures the smaller bands of vortices with higher accuracy, but the detected core region is not as accurate as in Figure 3b. Finally, Figure 3d presents the combined feature-aware classification field which is able to preserve both the core and the small vortex bands with high accuracy. This means that tracking using classification fields will yield robust results since it is able to capture the target feature in detail.

For any tracking algorithm, the accuracy of extraction of features is an important step. If the extracted features are not reliable then the tracking may not give a meaningful result to the scientists. Since we are dealing with an uncertain feature definition, the proposed technique solves an important problem in automatically detecting the feature evolution over time. In the next section we present a technique for tracking features using the feature-aware classification fields.

6 Tracking Using Feature-Aware Classification Fields

Feature tracking in visualization is an important task and researchers have looked into this problem in the past [16, 24, 29, 30, 34, 35, 43]. Even though above techniques achieve stable tracking results, the feature extraction part of those methods rely on the precise feature description. In this work, we extend the capability of the feature tracking techniques by introducing a new distribution driven method, which is able to track volume features that are selected directly from raw data interactively, therefore without any precise description. We have used GMM of the selected region to model the target feature. To make the feature extraction more accurate and robust, we first transform the data into a feature-aware classification field as described in the earlier section. Such a field allows us to classify the data by their relevance to the user interested feature. In the classification field, regions with high possibility values represent the existence of the feature of interest and they can be easily visualized and explored. We perform tracking in this feature-aware space because the classification field allows to easily extract the feature by applying a suitable user specified threshold on the possibility values. Below we discuss the method in detail.

Algorithm 2 Tracking In Feature-Aware Classification Field

1: Input: $GMM(b_i)$, $GMM(feature) : \forall i \in 1,2,..n$
2: Initialize $f_{target} := GMM(feature)$
3: **for all** t in T **do**
4: Generate Feature-Aware Classification Field($GMM(feature)$, t) (Algorithm 1)
5: Thresholding ($>= poss_{th}$) on the Classification field.
6: Apply connected component algorithm on the thresholded results.
7: Compute distance between centers of target feature and all the detected regions from the current time step.
8: Find the best match l with the minimum distance to the target feature f_{target}.
9: Set $f_{target} := l$
10: **end for**

A visual inspection of the classification field using interactive volume visualization techniques allows scientists to easily locate their feature of interest by focusing on the high valued regions in the field. Our method allows the users to inspect the classified field of an initial time step and provide a suitable threshold possibility value ($poss_{th}$), which we apply to the classification fields of later time steps for automatic extraction of the target feature. After the threshold ($poss_{th}$) is applied to the classification fields, a connected component based region growing algorithm is employed to the result of the thresholding to extract all the connected features. Each such detected region is treated as a separate feature. A match with the given target feature is found by using a distance based method as was described earlier in [29], where the Euclidean distances between the centers of the target feature f_{target} at time t with all the other detected regions at time $t+1$ are computed and the region with the minimum distance is tagged as the target feature f_{target} in time $t+1$. This process is repeated for consecutive time steps to continue the tracking process. Algorithm 2 sums up the steps of our tracking algorithm which implicitly calls Algorithm 1 for generating the classification fields for each time step and tracks the feature of interest using it. In Algorithm 2, T represents the final time step. As can be seen, at the end of calculation of every time step, the f_{target} is updated with the best matched feature l from the current time step which is used in the next time step as the reference feature.

In some complex scenarios, apart from changing the position and size, the target feature may undergo several evolutionary events such as birth, split, merge, and dissipation etc. Unlike traditional automatic feature tracking systems where all the existing features are extracted based on a predefined feature definition and tracked over time, we are more concerned with tracking a specific region of interest which has been identified vaguely from a region directly specified from the raw data. Therefore, we do not require to focus on a feature birth process explicitly. To detect the dissipation of a feature, we set an upper limit to the matched minimum distance value. The motivation is that given sufficient temporal resolution, the evolutionary change of a time-varying feature happens gradually and if a sudden anomaly is detected during the tracking in terms of the matched minimum distance, the event needs further attention of experts. Therefore, during tracking, we compare the value of the matched distance at each time step with the predefined upper limit and if the value is greater than the limit, we finish tracking and report the time step back to the user for further investigation. Events like feature split or merge can be detected in our system by keeping track of the feature mass. In our tracking algorithm, we measure the mass of the identified feature as was described in [29] at each time step and compare it with the previous time step. A sudden and large change of mass indicates a potential feature split or merge event, where increase in mass indicates feature merge and decrease in mass signifies feature split. Even though a big drop in the mass may also reflect a shrinking/disappearing event, however, by keeping track of such event, our system is able to detect those time steps and they are reported back to the users for further investigation. The proposed method allows users to set a predefined threshold value based on their

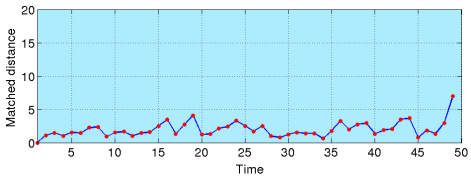
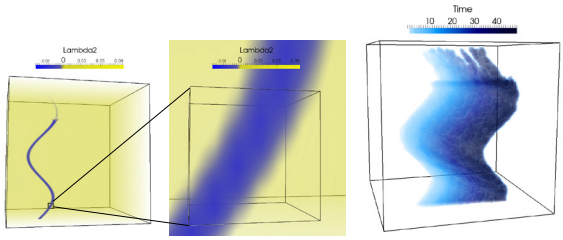


Figure 4: Matched distance values over time for Tornado data set.



(a) Selected feature in the Tornado data set and a zoomed in view of the selected region. (b) Temporal trace volume of the tracked feature.

Figure 5: Feature tracking in Tornado data set.

domain knowledge for the change of mass between two consecutive time steps and if a change of more than the threshold is detected, the time step is marked and is reported back for further exploration. In Figure 4, we show one example plot of the minimum matched distance values for all the time steps for the Tornado data set. The upper limit to the matched distance was set to 15 for this experiment. As we can see that, for all the time steps the proposed method was able to extract and track the target feature with high consistency which is reflected by the low matched distance values throughout all the time steps.

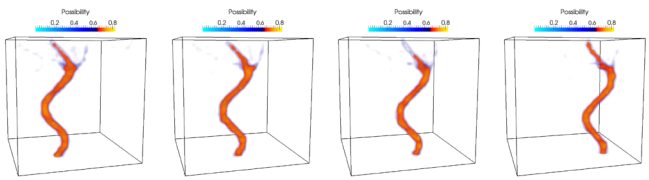
7 Results

In this section we demonstrate the effectiveness of the proposed method in extracting and tracking features with fuzzy definition, using several scientific data sets. All the experiments were done on a Linux machine with an Intel core i7-2600 CPU, 16 GB of RAM and an NVIDIA Geforce GTX 660 GPU with 2GB texture memory. In all the case studies, using a maximum of 3 Gaussians produce stable results and so the maximum number of Gaussians per GMM is set to 3 for the experimentation.

7.1 Case Study 1: Tornado Data Set

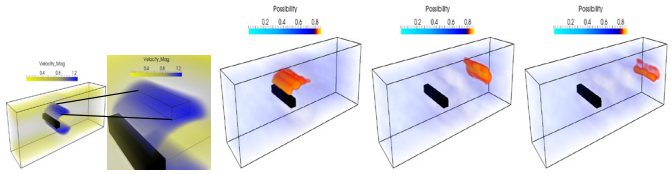
The first experiment is to study a Tornado data set of dimension $128 \times 128 \times 128$, containing velocity vectors at each grid point, generated by an analytical function [8]. The data set has 50 time steps and simulates a tornado like vortex structure. For this case study, we have modified the analytical equation so that the center of the tornado changes position with time. The block size of $4X4X4$ is used for the experimentation. The goal is to track the vortex core of the tornado.

Figure 5a presents the selection of the fuzzy vortex region from the first time step of the data. We have computed the lambda2 (λ_2) vortex criterion using the velocity field for measuring *vortexness* at each spatial point. Even though theoretically negative values of λ_2 criterion



(a) T=10. (b) T=20. (c) T=30. (d) T=49.

Figure 6: Extraction and tracking in Tornado data set. The vortex core is tracked over time and the results of 4 selected time steps are shown.



(a) Selected feature. (b) T=25. (c) T=35. (d) T=40.

Figure 7: Extraction and tracking of the selected feature in 3D Flow around a cylinder data set. High velocity feature at 3 selected time steps have been shown.

represent vortex region, deciding a precise threshold using a λ_2 value is difficult and often needs manual tuning. Nevertheless, visualizing the lambda2 field allows experts to mark the region in the data where the vortex exists. From the highlighted region, we extract all the points and fit a GMM on those data points to represent the region (*feature of interest*) using its distribution. After that, the proposed extraction and tracking method is applied on all the other time steps of the data for automatically extracting and tracking the vortex region.

Figure 6a, 6b, 6c, and 6d depict the tracked feature of interest at 4 time steps. Even though we have estimated the feature distribution by only using the sample points from the initial highlighted region as shown in Figure 5a, our extraction algorithm is able to recover the complete connected vortex core region from the data accurately. For the construction of feature-aware classification fields, the value of γ is set to 0.5 which means the final classification will have 50% contribution from the foreground component (the motion component) and rest of the 50% contribution comes from the similarity based measure. In this work, we perform tracking in the classified fields, and focus on the high valued regions. For extracting the target region which strongly represents the feature of interest, regions with a possibility value higher than 0.65 are considered in this study. The results reflect that our method is able to extract and track automatically the feature over time. A short demo video of the tracking of the tornado feature is provided which demonstrates that our technique is able to track the feature consistently over time with high accuracy.

To provide a comprehensive view of the tracked feature, we also create a temporal trace of the feature by constructing a scalar volume using time steps as the scalar value at each grid point. At the end of tracking, the temporal trace volume is obtained which shows the dynamic transition of the tracked feature over space. In Figure 5b we show such a feature trace volume of the Tornado data set. The movement of the tracked feature from left to right is evident from the gradual transition of color. When the feature have a continuous motion, then the trace volume allows the experts to visualize the overall temporal evolution of the feature efficiently.

7.2 Case Study 2: 3D Flow around a cylinder data Set

This case study demonstrates feature extraction and tracking in the 3D Flow around a cylinder data Set. This is a 3D time-dependent incompressible flow data with a Reynolds number of 200 and a square cylinder has been positioned symmetrically between the two parallel walls. The data set consists of velocity vectors and simulates a complex periodic vortex shedding phenomena which is well known as the *von Kármán vortex street*. This is a direct numerical Navier Stokes simulation by Simone Camarri and Maria-Vittoria Salvetti, Marcelo Buffoni, and Angelo Iollo [3] which is made publicly available [13]. We have used a uniformly re-sampled version which has been provided by Tino Weinkauf and used in von Funck et al. [40]. The data is represented by a grid of $192 \times 64 \times 48$ and there are total 102 time steps. For experimentation, $4X4X4$ block size is used.

In order to explore the periodic flow pattern and the vortex shedding which are produced by the von Kármán vortex street, study of the velocity field is useful. The high velocity waves show the periodic patterns exist in the data and help scientists to understand the phenomena in greater detail. For tracking the rapidly moving high velocity vortex street, we have used velocity magnitude field in this case study.

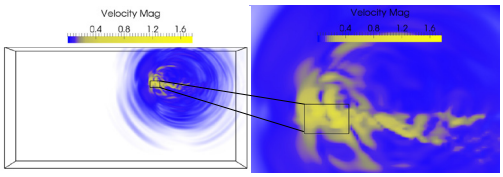
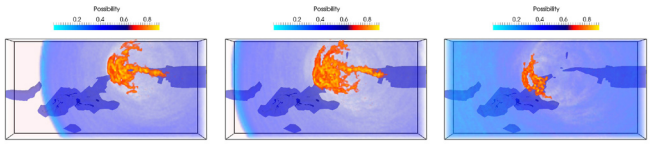


Figure 8: Selected feature in Earthquake data set and a zoomed in view of the selected region.

In Figure 7a, the region with high velocity magnitude is selected as the region of interest which moves periodically through the simulation grid. Tracking of such region is non-trivial as the feature is hard to define by a hard threshold value. Furthermore, isolation of such feature consistently over the time range poses significant challenges. We have applied our extraction and tracking algorithm in this data set and Figure 7 demonstrates the results we have obtained. In Figure 7b, 7c, and 7d we show the tracked feature which moves forward over time. For creating the classification fields, the value of γ is set to 0.7 which means that the foreground measure contributes more than the similarity measure in this experiment and for isolating the feature, we have used possibility value larger than 0.8. From the results depicted in Figure 7, it is evident that the proposed method is able to isolate and track the selected feature over time effectively.

7.3 Case Study 3: Earthquake Data Set

Our next case study uses an Earthquake data set which is a time-varying data consisting of wave velocity vectors. The dimensions of the 3D volume data is $750 \times 375 \times 100$ and we have used 100 time steps to perform the experiment. The data set describes a simulation of earthquake of magnitude 7.7 on the Southern San Andreas Fault and was generated using TeraShake 2.1. The TeraShake 2.1 simulation was performed by scientists at the Southern California Earthquake Center (SCEC) and researchers at San Diego Supercomputer Center (SDSC). It records the velocity vectors of the earthquake waves spreading over time. For this case study, we have considered the mag-



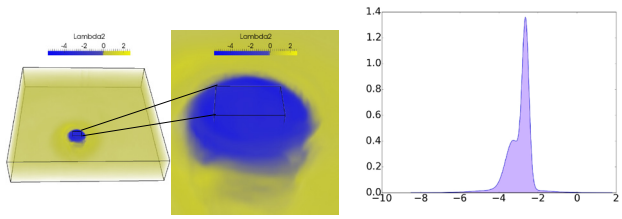
(a) T=10. (b) T=20. (c) T=30.

Figure 9: Extraction and tracking of the propagation of high velocity shock waves in Earthquake data set. Results of 3 selected time steps are presented.

nitude of the velocity field since studying the high velocity waves, the direction and intensity of the earthquake can be understood in detail. The data is divided into blocks of $5X5X10$ for experimentation. In Figure 8, the region with high velocity magnitude is selected as the region of interest. Figure 9 depicts the result of tracking such high velocity region over the selected time range. In Figure 9a, 9b, and 9c we present the tracked target region at 3 selected time steps 10, 20, and 30 respectively. The images show how the high velocity waves propagate over time. We also visualize the land and the basin regions with the feature to reflect the areas effected by the strong wave. While creating the feature-aware classification fields, value of γ is set to 0.3 and in tracking phase, we have considered possibility values higher than 0.72 for isolating the feature of interest at each time step. Results presented in Figure 9 show that the proposed method is able to detect and track the strong wave front feature.

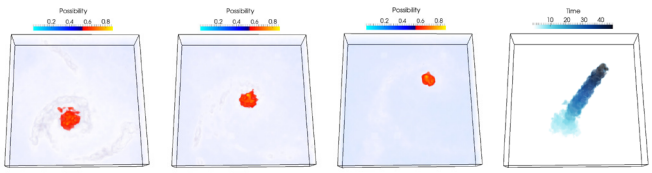
7.4 Case Study 4: Hurricane Isabel Data Set

Next, we present the fourth case study using Hurricane Isabel data which is a time-varying data set containing a vector field of wind velocity. The data set is a courtesy of NCAR and the U.S. National



(a) Feature selection in Hurricane Isabel data. (b) Estimated GMM of the feature.

Figure 10: Selected feature in Hurricane Isabel data set, a zoomed in view and the GMM of the selected region.



(a) T=11. (b) T=28. (c) T=41. (d) Temporal trace of the feature.

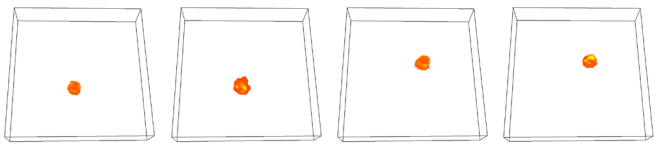
Figure 11: Extraction and tracking of the vortex at Hurricane eye in Isabel data set. Results of tracking of 3 selected time steps are shown.

Science Foundation (NSF), and was created using the Weather Research and Forecast (WRF) model. The data set corresponds to an actual physical space of 2139km (east-west) x 2004km (north-south) x 19.8km (vertical), which is represented by a grid of $250 \times 250 \times 50$ and there are total 48 time steps. For experimentation, $5X5X5$ block size is used.

An important task in this data set is to extract and track the temporal evolution of the low pressure eye (core) of the storm system where a strong vortical flow exists. As discussed earlier in [2], accurate tracking of the location and spread of the eye is critical in understanding the strength of the storm. We have computed the lambda2 (λ_2) field using the velocity field for the initial selection of the vortex region. Note that, the use of a hard thresholding on the λ_2 value is not always robust and often requires user intervention. Also, the dynamic nature of the feature makes the task of tracking challenging.

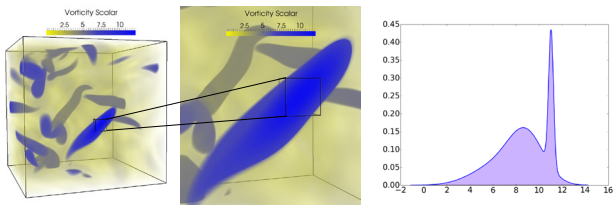
In Figure 10a, the feature selection is demonstrated. It is evident that the feature boundary is fuzzy and separating it using a hard threshold is cumbersome. Figure 10b displays the GMM which is estimated from the selected region and it is treated as the feature definition in the proposed tracking algorithm. The tracking algorithm, described in Algorithm 2 is applied on the entire data set over all the time steps using the estimated GMM as the feature of interest. At every step of the Algorithm 2, it internally calls the Algorithm 1 for the construction of the feature-aware classification field. Since the feature of interest has a motion in the space, we use $\gamma = 0.3$ for capturing such motion information while computing the classification fields. Final tracking is done on the classification fields and the feature is extracted at each time step and visualized using volume visualization techniques. In Figure 11, the detected vortex region (the Hurricane eye) is presented for 3 selected time steps to show effectiveness of our method. For isolating the feature in the final classified possibility fields, possibility value of greater than 0.55 is considered. From Figures 11a, 11b, and 11c it is evident that the proposed method is able to detect and track the eye of the storm with high accuracy. Also in Figure 11d we show the temporal trace volume of the feature to present the overall evolution of the target feature.

Figure 12 shows a comparison between our method and the correspondence based volume tracking method [29]. We have implemented the volume tracking algorithm for this comparative study. Since the volume tracking method requires a predefined precise feature description for tracking, we have used $\lambda_2 < -0.001$ as our feature definition in this study. Figure 12a and 12c show the results obtained from the



(a) Feature extracted using volume tracking method at T=15. (b) Feature extracted by the proposed method at T=15. (c) Feature extracted using volume tracking method at T=35. (d) Feature extracted by the proposed method at T=35.

Figure 12: A comparison between the volume tracking method and the proposed algorithm. The proposed method is able to produce comparable results with a fuzzy feature descriptor.



(a) Feature selection in Vortex data set with a zoomed in view. (b) Estimated GMM of the user interested feature.

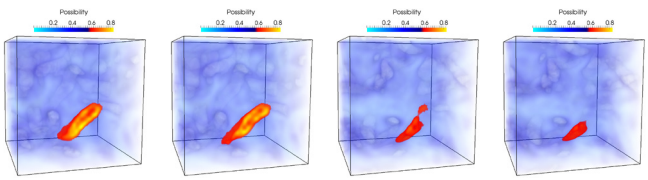
Figure 13: Selected feature in Vortex data set, a zoomed in view and the GMM of the selected region.

volume tracking method for time steps 15 and 35 respectively, and Figure 12b and 12d show the extracted feature for the same time steps with the fuzzy feature description. It can be seen that the results obtained by the proposed method are very similar to that of the volume tracking method with only minor differences. It shows that the proposed method is capable of extraction and tracking of features which are only vaguely defined. Therefore, this method enhances the capability of the existing feature tracking algorithms by providing a novel way of dealing with fuzzy volume features. Furthermore, the temporal trace volume depicted in Figure 11d also confirms that our method is robust and can track the time-varying features with high accuracy consistently.

7.5 Case Study 5: Vortex Data Set

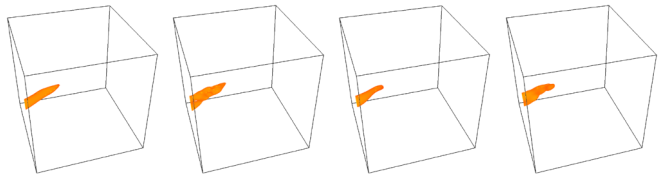
Our final case study shows the result on a Vortex data set which is a pseudo-spectral simulation of coherence vortex structures. The dimension of this data set is $128 \times 128 \times 128$ and is divided into blocks of $4X4X4$. The scalar variable in the data is vorticity magnitude. We used 30 time steps of the data set to demonstrate the effectiveness of our algorithm on this data set. The data set contains several tubular vortex cores which undergo rapid shape changes and complex events such as split, merge, creation, and dissipation.

Figure 13a shows a specific region which is selected for this case study from the first time step. Figure 13b depicts the GMM estimated from the selection as the feature to be tracked. From Figure 13a it is visible that there are several vortex regions in the data set and therefore, explicit correspondence is important in this case for accurate tracking of the selected feature. We have applied our tracking algorithm, presented earlier, for tracking the selected feature. In Figure 14



(a) T=3. (b) T=6. (c) T=14. (d) T=15.

Figure 14: Extraction and tracking using Vortex data set. Tracked feature for 4 selected time steps are displayed.



(a) Feature extracted using volume tracking method at T=3. (b) Feature extracted using proposed method at T=3. (c) Feature extracted using volume tracking method at T=8. (d) Feature extracted using proposed method at T=8.

Figure 15: A comparison between the volume tracking method and the proposed algorithm using Vortex data set.

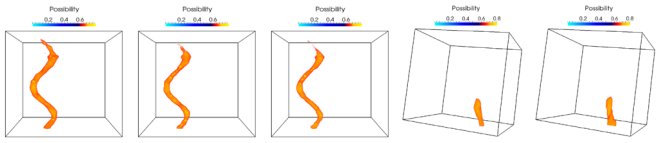
we demonstrate the results of extraction and tracking by showing the tracked feature at 4 selected time steps. Since, the features in this data set change their shape rapidly and the motion is not a dominant component, the value of γ is set to 0.15, so that we take larger contribution (85%) from the similarity based possibility measure to achieve higher accuracy in tracking. After the feature-aware classification fields are constructed, we use possibility value 0.58 as a threshold for identifying all the connected regions in the data set. Then by applying the Algorithm 2 we detect and track the target feature.

From the Figures 14a - 14d, it is evident that the key feature gradually dissipates as time increases. Finally, the feature dissipates at time step 22 which is detected in our system by the predefined upper limit set for the distance value between the matched feature and tracked feature from previous time step. A detailed visual exploration shows that our method is able to show the feature split phenomena clearly. From the tracked results presented in Figure 14, time steps $14-16$ are significant because the target feature undergoes a split in this time range. In Figure 14c we can see that the feature is about to split into two segments and the split happens in time step 15. As the split happens, we continue to track the closest component of the feature and report the time step where the split has happened. In our current system, time step 15 is detected to have a potential split since in this time step mass of the feature decreases 30.9% compared to its previous time step. A short demo video of the tracking of the selected feature is provided as a supplementary material which demonstrates the temporal evolution of the feature for all the tracked time steps. Another observation here is that, as the feature gradually shrinks in size, the possibility values also drop. This trend is visible from the sub-figures in Figure 14 where the higher time steps show less yellow regions on the feature and more red regions, indicative of low possibility values.

In Figure 15 we present a comparison between the proposed method and the volume tracking method. This data set has multiple complex features (vortex cores) which are identified as isolated segmented regions where the segmentation criterion used is region with scalar value $>= 7.0$. After the segmentation is done, all the isolated regions are treated as separate features and a segmented region is selected as the target feature and applied the volume tracking method to track it over time. Also for applying the proposed method on the same feature, a small region is selected from the target feature as a representative sample. Figure 15a and 15c show the extracted feature obtained by the volume tracking method at time steps 3 and 8 and Figure 15b and 15d show the results produced by the proposed method without the background context. By observing the results depicted in Figure 15, it is evident that the proposed method generates very similar results compared to the volume tracking algorithm and can robustly track the feature. So, in the absence of a predefined feature definition, the proposed method presents a tracking framework which allows users to highlight their target region of interest directly in the raw data and automatically track it over time efficiently.

8 Parameter Choice and Performance Analysis

From Equations 2 and 4, we observe that β controls the contributions of the new time step and the previous time step while estimating the GMM parameters. So, the value of β determines how quickly the parameters of the GMMs change with time. Since the transition in the time-varying process is usually smoother, changing the value of β does



(a) Tornado feature with $\beta = 0.2$. (b) Tornado feature with $\beta = 0.4$. (c) Tornado feature with $\beta = 0.7$. (d) Vortex feature with block size 4X4X4. (e) Vortex feature with block size 8X8X8.

Figure 16: Parameter choices for the proposed method.

not impact the results significantly. However, increasing the value of β may cause some loss of accuracy in the estimated feature as can be seen from Figures 16a-16c. In all the experimentation, we have found that $\beta = 0.2$ works well and gives stable results. So, we have used $\beta = 0.2$ for all the case studies. Since we advocate for modeling the local properties of the data for accurate feature classification, the proposed method works well for smaller block sizes. Smaller blocks allow us to better preserve the accuracy of the feature in the classification field. In Figure 16d and 16e we show the results obtained from the vortex data set with block sizes $4X4X4$ and $8X8X8$ respectively. It is observed that in both the cases the proposed method is able to segment the feature, however, the one with the smaller block size produces a more refined estimated feature.

Table 1: Data set descriptions and average CPU Time performance per time step for different computation components.

Data Sets	Block size	Classification Field Creation (secs.)	Tracking (secs.)
Tornado	$4X4X4$	3.580	1.432
Hurricane Isabel	$5X5X5$	5.041	0.737
Vortex	$4X4X4$	3.601	0.492
Earthquake	$5X5X10$	51.133	15.83
Flow Around a cylinder	$4X4X4$	1.1724	0.3378

In Table 1 the timings are reported for the test cases. The classification field generation time includes I/O time and can be done separately before the tracking process. The incremental estimation of the GMMs and the foreground estimation algorithm requires only a linear scan of the raw data. The incremental algorithm used here is significantly less expensive in estimating the GMM parameters compared to the off-line EM algorithm and also suitable for streaming data/in-situ frameworks. The estimated GMMs are used for computing the feature similarity measure and finally the previously measured foreground information is combined with the similarity measure to generate the classification fields. The advantage of the method is that even when the feature is changed, the algorithm does not require the access to the raw data and can generate the classification fields using the previously computed GMMs. It only needs raw data access for the first time step to re-estimate the feature GMM. Since we use mixture of Gaussians to model the data, the storage complexity is significantly low as we have to store only the parameters of the GMM and a possibility value for each block for future use. For creating the final possibility field a segmentation based region growing algorithm is used. Table 1 shows the tracking time separately for all the case studies. Also, since the computation is done block-wise, therefore for significantly large data sets, the algorithm can be parallelized by distributing data over multiple nodes and processing each block in parallel.

9 Discussion

For any feature tracking algorithm, a robust extraction method is a key component, because if the extracted features are not reliable, then tracking them can lead to misleading outcomes. Almost all of the previous tracking works have assumed a predetermined feature definition for the extraction stage. However, little attention is paid when the precise feature definition can not be obtained. In this work, we extend the capability of feature extraction and tracking algorithms by proposing a new distribution driven approach which is able to deal with the uncertainties inherent in the given fuzzy feature definition and allow reliable feature extraction and tracking.

For tracking *predefined* volumetric features, researchers have proposed comprehensive techniques [15, 35]. However, one potential disadvantage of those techniques is that, if the feature definition is changed, the algorithm would require going through the raw data again. In another work, a texture based feature tracking algorithm was proposed in [2] where high-dimensional textural attribute vectors are used for feature representation. The technique obtained accurate results even with low temporal sampling. But the technique required to find an appropriate neighborhood window for searching the feature. Also, the drifting problem is recognized as a limitation of this work. A recent trajectory based feature tracking algorithm [30] has demonstrated promising results, but it is limited to only data sets containing additional particle data and its accuracy is dependent on the particle density. A more general flow pattern extraction technique for 2D flow fields has been introduced earlier in the works of Schlemmer et al. [32] based on moment invariants. The method is able to detect critical points in flow fields and also find user defined (in circular domain) complex flow patterns from the data efficiently. The work presented in [32] and the proposed method, both achieve feature estimation by utilizing a pattern matching approach where we use distributions as a statistical pattern for the target feature. The goal of our algorithm in this work is to efficiently track vaguely defined volume features in 3D time-varying scalar fields.

Our method efficiently exploits both spatial and temporal coherency present in the data and utilizes them to compute the two key information: (1) motion and (2) similarity with target feature distribution. Since none of these information alone is sufficient for achieving a robust feature extraction, we combine them to construct a feature-aware classification which helps us to extract and track key features. So, in the absence of precise feature definition, proposed method allows tracking of fuzzy features robustly. Another advantage of the proposed method is the use of the incremental framework for data modeling. Since the model does not require all the data beforehand and can work as new data stream in, the method is suitable for a in-situ feature tracking framework. Also the parametric distribution representation keeps the storage requirements tractable as the data size scales up. However, with increased block size, the feature extraction accuracy gets affected since smaller features inside a block can not be captured with sufficient details. Also, if multiple features exist inside a block then, the proposed method will detect the block as part of the feature but separation between them is not be possible.

10 Conclusion and Future Works

In the absence of a precise feature definition, the proposed method models the data space and the specified region of interest using mixtures of Gaussians and transforms the data space into a feature-aware classified field where high valued regions reflect a higher possibility of the existence of the feature. Such a distribution driven classification allows us to construct a robust tracking algorithm where the tracking is performed in the classification field. In the future, we wish to integrate our system with an in-situ streaming data framework to perform real time feature extraction and tracking. Besides this, we would like to adapt our method for feature tracking in time-varying ensemble data sets and also to multivariate data sets. Furthermore, we also want to study the effectiveness of our method on data sets with sparsely sampled time steps.

Acknowledgments

This work was supported in part by NSF grants IIS- 1250752, IIS-1065025, and US Department of Energy grants DE- SC0007444, DE-DC0012495, program manager Lucy Nowell.

References

[1] J. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, 1998.

[2] J. Caban, A. Joshi, and P. Rheingans. Texture-based feature tracking for effective time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1472–1479, Nov. 2007.

[3] S. Camarri, M.-V. Salvetti, M. Buffoni, and A. Iollo. Simulation of the three-dimensional flow around a square cylinder between parallel walls at moderate Reynolds numbers. In *XVII Congresso di Meccanica Teorica ed Applicata*, 2005.

[4] A. Chaudhuri, T. H. Wei, T. Y. Lee, H. W. Shen, and T. Peterka. Efficient range distribution query for visualizing scientific data. In *Pacific Visualization Symposium (PacificVis), 2014 IEEE*, pages 201–208, March 2014.

[5] J. Chen, D. Silver, and L. Jiang. The feature tree: visualizing feature tracking in distributed amr datasets. In *Parallel and Large-Data Visualization and Graphics, 2003. PVG 2003. IEEE Symposium on*, pages 103–110, Oct 2003.

[6] R. T. Clemen and R. L. Winkler. Combining probability distributions from experts in risk analysis. *Risk Analysis*, 19(2):187–203, 1999.

[7] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149 vol.2, 2000.

[8] R. Crawfis and N. Max. Texture splats for 3d scalar and vector field visualization. In *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on*, pages 261–266, Oct 1993.

[9] A. Elgammal, R. Duraiswami, and L. Davis. Probabilistic tracking in joint feature-spatial spaces. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–781–I–788 vol.1, June 2003.

[10] C. Garth, X. Tricoche, and G. Scheuermann. Tracking of vector field singularities in unstructured 3d time-dependent datasets. In *Visualization, 2004. IEEE*, pages 329–336, Oct 2004.

[11] L. Gosink, C. Garth, J. Anderson, E. Bethel, and K. Joy. An application of multivariate statistical analysis for query-driven visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(3):264–275, March 2011.

[12] Y. Gu and C. Wang. Transgraph: Hierarchical exploration of transition relationships in time-varying volumetric data. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2015–2024, Dec 2011.

[13] International CFD Database, http://cfd.cineca.it/.

[14] H. Jänicke, A. Wiebel, G. Scheuermann, and W. Kollmann. Multifield visualization using local statistical complexity. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1384–1391, 2007.

[15] G. Ji, H.-W. Shen, and R. Wenger. Volume tracking using higher dimensional isosurfacing. In *Visualization, 2003. VIS 2003. IEEE*, pages 209–216, Oct 2003.

[16] G. Ji and H. wei Shen. Feature tracking using earth movers distance and global optimization, pacific graphics 2006.

[17] C. Johnson and J. Huang. Distribution-driven visualization of volume data. *Visualization and Computer Graphics, IEEE Transactions on*, 15(5):734–746, Sept 2009.

[18] T.-Y. Lee and H.-W. Shen. Visualizing time-varying features with tac-based distance fields. In *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific*, pages 1–8, April 2009.

[19] T.-Y. Lee and H.-W. Shen. Efficient local statistical analysis via integral histograms with discrete wavelet transform. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2693–2702, Dec 2013.

[20] S. Liu, J. Levine, P. Bremer, and V. Pascucci. Gaussian mixture model based volume visualization. In *Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on*, pages 73–77, Oct 2012.

[21] C. Lundstrom, P. Ljung, and A. Ynnerman. Local histograms for design of transfer functions in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1570–1579, Nov. 2006.

[22] C. Muelder and K.-L. Ma. Interactive feature extraction and tracking by utilizing region coherency. In *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific*, pages 17–24, April 2009.

[23] H. Obermaier and K. I. Joy. Local data models for probabilistic transfer function design. In *Eurographics Conference on Visualization (EuroVis 2013) Short Papers*, pages 43–47, 2013.

[24] S. Ozer, D. Silver, K. Bemis, and P. Martin. Activity detection in scientific visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 20(3):377–390, March 2014.

[25] S. Ozer, J. Wei, D. Silver, K.-L. Ma, and P. Martin. Group dynamics in scientific visualization. In *Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on*, pages 97–104, Oct 2012.

[26] E. Polat and M. Ozden. A nonparametric adaptive tracking algorithm based on multiple feature distributions. *Multimedia, IEEE Transactions on*, 8(6):1156–1163, Dec 2006.

[27] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Comput. Graph. Forum*, 22(4):775–792, 2003.

[28] F. Reinders, F. H. Post, and H. J. W. Spoelder. Attribute-based feature tracking. In *Data Visualization 99*, pages 63–72. Springer Verlag, 1999.

[29] R. Samtaney, D. Silver, N. Zabusky, and J. Cao. Visualizing features and tracking their evolution. *Computer*, 27:20–27, 1994.

[30] F. Sauer, H. Yu, and K.-L. Ma. Trajectory-based flow feature tracking in joint particle/volume datasets. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints):1, 2014.

[31] K. Schindler and H. Wang. Smooth foreground-background segmentation for video processing. In *Proceedings of the 7th Asian Conference on Computer Vision - Volume Part II*, ACCV'06, pages 581–590, Berlin, Heidelberg, 2006. Springer-Verlag.

[32] M. Schlemmer, M. Heringer, F. Morr, I. Hotz, M.-H. Bertram, C. Garth, W. Kollmann, B. Hamann, and H. Hagen. Moment invariants for the analysis of 2d flow fields. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1743–1750, Nov 2007.

[33] G. Sfikas, C. Constantinopoulos, A. Likas, and N. P. Galatsanos. An analytic distance metric for gaussian mixture models with application in image retrieval. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*, ICANN'05, pages 835–840, Berlin, Heidelberg, 2005. Springer-Verlag.

[34] D. Silver and X. Wang. Volume tracking. In *In Proceedings of the Visualization 96 Conference*, pages 157–164. Computer Society Press, 1996.

[35] D. Silver and X. Wang. Tracking scalar features in unstructured data sets. *Proceedings Visualization '98 (Cat. No.98CB36276)*, 98, 1998.

[36] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –252 Vol. 2, 1999.

[37] H. Theisel and H.-P. Seidel. Feature flow fields. In *Proceedings of the Symposium on Data Visualisation 2003*, VISSYM '03, pages 141–148, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[38] T. T. Tran, L. Peng, B. Li, Y. Diao, and A. Liu. Pods: A new model and processing algorithms for uncertain data streams. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 159–170, New York, NY, USA, 2010. ACM.

[39] F.-Y. Tzeng and K.-L. Ma. Intelligent feature extraction and tracking for visualizing large-scale 4d flow simulations. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pages 6–6, Nov 2005.

[40] W. von Funck, T. Weinkauf, H. Theisel, and H.-P. Seidel. Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization 2008)*, 14(6):1396–1403, November - December 2008.

[41] C. Wang, H. Yu, and K.-L. Ma. Importance-driven time-varying data visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1547–1554, Nov 2008.

[42] Y. Wang, W. Chen, J. Zhang, T. Dong, G. Shan, and X. Chi. Efficient volume exploration using the gaussian mixture model. *Visualization and Computer Graphics, IEEE Transactions on*, 17(11):1560–1573, Nov 2011.

[43] Y. Wang, H. Yu, and K. Ma. Scalable Parallel Feature Extraction and Tracking for Large Time-varying 3D Volume Data. *Eurographics Symposium on Parallel Graphics and Visualization*, D:55–62, 2013.

[44] G. H. Weber, P.-T. Bremer, M. S. Day, J. B. Bell, and V. Pascucci. Feature tracking using reeb graphs. In V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny, editors, *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*, pages 241–253. Springer Verlag, 2011. LBNL-4226E.

[45] C. H. You, K. A. Lee, and H. Li. Gmm-svm kernel with a bhattacharyya-based distance for speaker recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(6):1300–1312, Aug 2010.