Texture-based feature tracking for effective time-varying data visualization

Jesus J. Caban, *IEEE Student Member*, Alark Joshi, *IEEE Student Member*, and Penny Rheingans *IEEE Member*

Abstract— Analyzing, visualizing, and illustrating changes within time-varying volumetric data is challenging due to the dynamic changes occurring between timesteps. The changes and variations in computational fluid dynamic volumes and atmospheric 3D datasets do not follow any particular transformation. Features within the data move at different speeds and directions making the tracking and visualization of these features a difficult task. We introduce a texture-based feature tracking technique to overcome some of the current limitations found in the illustration and visualization of dynamic changes within time-varying volumetric data. Our texture-based technique tracks various features individually and then uses the tracked objects to better visualize structural changes. We show the effectiveness of our texture-based tracking technique with both synthetic and real world time-varying data. Furthermore, we highlight the specific visualization, annotation, registration, and feature isolation benefits of our technique. For instance, we show how our texture-based tracking can lead to insightful visualizations of time-varying data. Such visualizations, more than traditional visualization techniques, can assist domain scientists to explore and understand dynamic changes.

Index Terms—Feature tracking, texture-based analysis, flow visualization, time-varying data, visualization

1 Introduction

Effective visualization of time-varying data is extremely difficult to achieve due to the complexity of illustrating multiple timesteps and clearly showing changes and variations over time. In time-varying data, each timestep has specific features which domain scientists are interested in visualizing, analyzing, and comparing. Commonly, key features move independently and at different speeds throughout the volume space without following any particular pattern or linear transformation. Such random transformations create a challenging problem when visualizing relative motion, tracking specific features, and illustration volumetric changes over time.

Domain experts and scientists need to analyze, track, and visualize specific features within time-varying data to better understand the underlying dynamics and build predictive models. Recently, it has been demonstrated that analysis and quantification of changes occurring within the inner eye-wall of hurricanes will permit forecasters to predict sudden strengthening or weakening[18]. Such research underscores the necessity of developing new techniques that through accurate tracking and extraction of hurricane features will provide more information-rich visualizations of changes occurring within the time-varying data.

Currently, forecasters and atmospheric scientists primarily visually inspect animations and simulations to analyze hurricane intensity, detect weather changes, and make predictions. However, such visual examinations are limited in terms of analyzing the hurricane changes due to the motion stabilization that occurs in the viewer's eyes. The human eye inherently focuses on broad patterns such as the general rotation of the hurricane while ignoring minor patterns including specific feature paths and changes. This is problematic since minor patterns are extremely important in the analysis and prediction process. An accurate tracking, such as the eye is not capable of accomplishing, of hurricane attributes–such as vortex tubes, clouds, and wind–would greatly enhance current visualization and prediction techniques.

- *Jesus J. Caban is with the University of Maryland, Baltimore County, E-mail: caban1@cs.umbc.edu.*
- *Alark Joshi is with the University of Maryland, Baltimore County, E-mail: alark1@cs.umbc.edu.*
- *Penny Rheingans is with the University of Maryland, Baltimore County, E-mail: rheingan@cs.umbc.edu.*

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007. Published 14 September 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

A widely used method to examine changes between different timesteps is that of applying a voxel-wise difference operation after aligning the data. Among volumes where features move independently and in different directions, however, such difference operations are impractical given the complexity of volume registration. Establishing correspondence between volumes is an important and key problem that must be solved before feature changes can be accurately analyzed. Without accurate volumetric registration, the differences between consecutive timesteps are imprecise and highly dominated by false positive results. Since most of the key features within dynamic time-varying datasets move independently and are mixed with noise or less important features, we cannot assume any particular rigid transformation between consecutive timesteps. Therefore, existing linear registration techniques are not capable of accurately aligning dynamic time-varying data. An alignment technique that offers a possible solution for registering time-varying data is a free-form deformable registration approach[1]. However, since features within dynamic time-varying data change in shape, intensity, and orientation, deformable registration techniques usually do not converge or the resulting registration is highly dominated by errors.

Within time-varying data, an accurate tracking of key features is a challenging task due to the overall restrictions and the few assumptions that should be made. First, features of interest are not easily segmented or thresholded from less important data. Second, a specific local feature may not have a corresponding key feature in the consecutive timestep. Third, a particular feature may not have overlap in the next timestep.

In this paper, we introduce a texture-based feature tracking technique capable of tracking multiple features over time by analyzing local textural properties and finding correspondent properties in subsequent volumes. Tracked objects are used to illustrate changes over time and to suggest volume segments that can be used as landmarks for a deformable volumetric registration. Our texture-based tracking method has been tested with both synthetic and real world time-varying volumetric data. Results with both synthetic and actual hurricane data show the strength of texture-based techniques and demonstrate how tracked features can be used to better visualize, illustrate, and annotate volumetric data for highlighting specific paths, variations, and changes over time.

2 Related Work

Research in the field of visualizing time-varying datasets has been focused mainly on the data to be visualized. Computational fluid dynamic (CFD) simulations figured heavily in early research of visualization of time-varying data. Samtaney et al. [12] were the first to

1077-2626/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

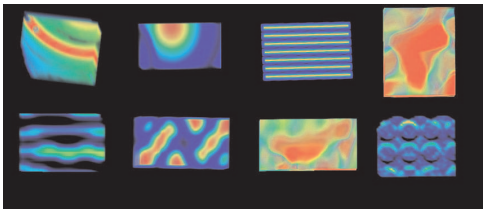
Texture Features			Texture Patterns
Histogram Statistics	**Co-occurrence Matrix**	**Run-Length Matrix**	
Standard mean	Co-occurrence energy	Long run emphasis	
Standard skewness	Co-occurrence Inertia Difference	Run length non-uniformity	
Standard Absolute Deviation	Co-occurrence Correlation	Low gray-level run emphasis	
Standard variance	Co-occurrence Average Difference	Short run low gray-level emphasis	
Standard Kurtosis	Co-occurrence Entropy Difference	Long run low gray-level emphasis	
Standard Deviation	Co-occurrence Inertia	Short run high gray-level emphasis	
	Co-occurrence Entropy	Long run high gray-level emphasis	
	Co-occurrence Average Sum	Short run emphasis	
	Co-occurrence Entropy Sum	Run gray-level non-uniformity	
		Run percentage	
		High gray-level run emphasis	

Table 1. Run-length-based, 1^{st} and 2^{nd} order statistical texture features used to accurately characterize specific textural patterns. The right image shows a set of subvolumes with characteristic textural patterns.

identify computer vision feature extraction techniques and to extend them from the 2D image domain to the 3D volume domain.

Since then a number of researchers have proposed other techniques for tracking features within volumetric time-varying data. Silver and Wang [13, 14] presented a feature tracking technique that extracts features, organizes them into an octree structure, and tracks the thresholded, connected components in subsequent timesteps by assuming that all the features between adjacent timesteps overlap. Reinders et al. [10] proposed a tracking technique that uses feature attributes, such as position, mass, and size, to solve the feature correspondence problem between frames. Ji et al. [7] introduced a method to track local features from time-varying data by using higher-dimensional isosurfacing. Tzeng and Ma [16] proposed a technique to extract and track features in time-varying data by using a machine learning module capable of learning from the transfer function which specific features the user wishes to track and visualize and then applying that knowledge to the visualization pipeline in subsequent volumes.

There are, however, several limitations to existing feature tracking techniques for time-varying volumetric data. First, previous approaches assume a high and sufficient temporal sampling in which features overlap in subsequent timesteps. Second, most existing techniques assume that important features can be easily segmented. That is, the 3D image can be thresholded to find and highlight the specific features to track. Third, noisy volumetric data has not been considered.

Texture analysis and pattern classification techniques have been widely used for various tasks in the computer vision, computer graphics, and visualization fields. Belongie et al. [2] used color- and texture-based image segmentation together with the expectation-maximization (EM) algorithm for retrieving similar images from a large image collection. Recently, Guan et al. [4] used texture-based techniques for categorizing traditional Chinese painting images.

In the medical imaging field, texture analysis has recently received much attention. Wang et al. [17] have used it for identifying anisotropic features to improve mammogram image registration. Christodoulou et al. [3] used texture-based analysis to extract features and accurately classify plaque images. Xu et al. [19] proposed a 3D texture feature approach to classify and differentiate lung CT images.

In designing our texture-based tracking technique, we sought to overcome specific limitations found in current feature tracking techniques and to construct a method capable of registering and delivering more effective visualization, illustration, annotation, and isolation of important features over time.

3 Approach

Our feature tracking technique is based on our observation that time-varying data presents specific pattern and textural features that are characteristic to particular volume segments. These characteristic patterns represent actual dynamic data movement and thus allow tracking of changes in time-varying data to be better illustrated and visualized.

Initially our system employs a feature selection process to determine which specific features of interest bear tracking. This advanced texture-based tracking, analysis, and correlation approach assigns each feature of interest a characteristic multi-dimensional vector that defines and contains the specific properties of the target features. Those characteristic multi-dimensional vectors that best match our target features are then located in subsequent timesteps, permitting multiple features to be tracked simultaneously over time. These tracked features allow time-varying volumetric data to be effectively visualized, illustrated, and annotated and thus better capture and illustrate dynamic changes occurring over time.

3.1 Texture Analysis

In computer vision and image processing, textures are fundamental to identifying, characterizing, and comparing objects and regions with similar properties. Image differentiation can be achieved by characterizing similar properties and regular frequencies that occur in the repeated patterns of specific images.

Through the detection and tracking of textural changes, especially entering, salient, and special patterns, our system can achieve better illustration and visualization of the variations occurring within time-varying data. Our technique uses several textural metrics to identify and track features within dynamic time-varying data.

We use a combination of first-order, second-order, and run-length matrices to analyze, characterize, and compare textures. The first-order statistics measure the likelihood of observing a gray value at a random location in the image. Second-order statistics are defined as the likelihood of observing a pair of voxels v_1 and v_2 separated by a distance vector $\overrightarrow{d_{xyz}}$. Run-length matrices capture the coarseness of the texture.

First-order statistics contribute six different textural elements to our multi-dimensional feature vector. A frequency distribution or histogram is generated for each feature of interest, and then texture attributes, such as mean, variance, and standard deviation, are computed for the particular subvolume under consideration. Table 1 shows the six first-order textural features and operations we compute for each feature being tracked.

To compute the second-order statistics, we use co-occurrence matrices. Second-order statistics measure the probability of a pair of voxels v_1 and v_2 with intensities i and j occurring at some distance $\overrightarrow{d_{xyz}}$ apart. The co-occurrence matrices measure the frequency that a grayscale value appears in relation to another grayscale value on the image. Co-occurrence matrices are defined by

$$C(i,j) = \sum_{p=1}^{n}\sum_{q=1}^{n}\sum_{r=1}^{n} f(x) = \begin{cases} 1, & \text{if } I(p,q,r) = i \\ & \text{and} \\ & I(p+\Delta x, q+\Delta y, r+\Delta z) = j \\ 0, & \text{Otherwise} \end{cases} \quad (1)$$

where I is the n^3 subvolume under consideration and $(\Delta x, \Delta y, \Delta z)$ is the distance vector between corresponding voxels. Our system averages four co-occurrence matrices in the direction of 0, 45, 90, and 135 degrees to guarantee rotational invariance.

A particular advantage of co-occurrence matrices is that several textural features can be easily computed for free. Haralick et al. [5] pre-

sented 14 texture features, including contrast, correlation, variance, and entropy, that are automatically obtainable from a co-occurrence matrix. Table 1 lists nine second-order textural features that our technique uses to accurately characterize and compare textures.

The third texture analysis technique our system uses is run-length analysis, which captures the coarseness of a texture in a given direction. The general idea is to find strings of consecutive pixels that have the same gray level intensity along a specific linear orientation. For a 3D image I, a run-length matrix $p(i,r)$ is defined as the number of pixels of gray level i and run length r along the $\overrightarrow{d_{xyz}}$ direction. Again, to guarantee rotationally invariant attributes, we average four different matrices in the direction of 0, 45, 90, and 135 degrees. Two main features we extract from run-length matrices are long- and short-run emphasis (LRE, SRE) defined by

$$SRE = \frac{\sum_{i=1}^{N_g}\sum_{r=1}^{N_r}\frac{p(i,r)}{r^2}}{\sum_{i=1}^{N_g}\sum_{r=1}^{N_r}p(i,r)} \tag{2}$$

$$LRE = \frac{\sum_{i=1}^{N_g}\sum_{r=1}^{N_r}r^2 p(i,r)}{\sum_{i=1}^{N_g}\sum_{r=1}^{N_r}p(i,r)} \tag{3}$$

where N_g is the number of different gray levels in the image, N_r is the number of different run lengths that occur, and $p(i,r)$ the run-length matrix.

Within 3D time-varying data, run-length comparisons are important since the dynamic changes and random motions of voxels frequently leave behind long runs characteristic of their paths. Table 1 lists 11 attributes we compute from run-length matrices. The run-length matrices our system incorporates are primarily those identified by Tang [15] as they proved to be excellent for preserving texture information and improving image classification.

3.2 Feature Selection

Our system's feature selection process first determines what specific features of interest to tracked. Features can be selected through either an automatic or a user-based technique. Given a timestep $T_{initial}$, the automatic selection process picks all the features within a specific threshold value (e.g. size, brightness) as those for tracking. However, in the interest of flexibility, we also incorporated the option of manually choosing specific features within the volumetric data as those of interest to be tracked. This option is the only feature of our system that requires user input.

Our system then assigns a characteristic multi-dimensional vector to each feature of interest within the selected feature set F_{set}. For each feature $F_i \in F_{set}$, a subvolume S_i is computed. Then, a multi-dimensional vector that defines and contains the specific textural properties is computed for each subvolume S_i and becomes a unique identifier for the feature under consideration. Table 1 shows the 26 different texture attributes our system computes for each subvolume of interest. The table 1 image shows a set of 3D texture blocks obtained from different volumetric data. An important feature of our texture-based tracking technique is that each 3D subvolume receives an individual vector $v \in R^{26}$ that defines and characterizes its specific texture properties.

Finally, we addressed the ultimate goal of our multi-dimensional feature vector - the determination of an accurate descriptor for each feature under consideration. Very little can be assumed about features within time-varying volumetric data. Thus, we could not presume that all 26 textural features had the same relevance or that features of interest had unique properties. For instance, there could be situations where specific textural properties, such as intensity, would be shared between the feature under consideration and neighboring voxels that are not a part of the feature of interest. The necessity of weighting our textural feature vectors according to their overall relevance arose from such problems and uncertainties. To maximize the importance of specific features and minimize spurious metrics not truly indicative of differences between features, rather than using the R^{26} feature vector employing a uniform weighting value, our system utilizes mutual information to define a different cost for each element of our feature vector.

Specifically, minimal-redundancy-maximum-relevance (mRMR) is used to assign a weight to each element of the feature vector [9]. For a given feature of interest, mutual information shared between voxels within the feature under consideration and its neighbors is used to compute the costs per metric, which allows development of a set of textural features that better differentiates one set of voxels from the other. This is accomplished by computing, from each subvolume of interest S_i, a new subvolume $\bar{S}_i = S'_i \cap S_i$ where S'_i is a larger subvolume that includes S_i in addition to a predefined offset of voxels per axes. Then a weighting value for each of the 26 textural features is derived by computing a texture-based multi-dimensional feature vector for $\bar{S}_i$ and finding in linear time the weighted permutation that increases vector differences between S_i and $\bar{S}_i$ while also maximizing feature relevance and minimizing redundancy. For consistency purposes, our system always includes the 26 textual metrics in the final descriptor for each feature of interest; however, a different cost is assigned to each metric.

After completing the feature selection and weighting function processes, our system begins tracking multiple features simultaneously over time by locating in subsequent timesteps the characteristic multi-dimensional vectors that best match our target features.

3.3 Feature Tracking

Our feature tracking technique incorporates similarity measurements to find the best texture match in subsequent timesteps. To find the best corresponding subvolume, the system first estimates a tracking window and then applies a distance measurement function that compares different subvolumes to find the feature with the closest textural properties.

The size and location of the tracking window are chosen based on the dynamics observed between different timesteps within the time-varying data. In time-varying volumetric data with high temporal sampling, that is where features of successive timesteps can be found within a given offset of the original feature's location, which need not be overlapping, a simple tracking window shift and resize allow estimation of the tracking space. An accurate tracking window may also be estimated by using the gradient descent of the image to approximate the local affine motion of the voxels and then estimating the direction and offset that a search window should be shifted between two given timesteps. Just such a technique has been used to track features in video sequences and to estimate inverse motion parameters [1]. Finally, in time-varying volumetric data where nothing is known about its dynamic movements and the intensity variation causes the gradient descent to return erroneous estimations, a significantly large subvolume–or even the entire volume–can be used as the tracking window. However, this last situation should be avoided as both the overall tracking errors and the time required by the algorithm to converge will significantly increase.

After a tracking window has been estimated, a distance measurement function is applied to find the "best match". The "best match" function BM is defined as:

$$BM(S^{t_i}_{xyz}) = min\left\{\sum_{p=min_x}^{max_x}\sum_{q=min_y}^{max_y}\sum_{r=min_z}^{max_z} ED(S^{t_i}_{xyz}, S^{ti+1}_{pqr})\right\} \tag{4}$$

where $S^{t_i}_{xyz}$ is the subvolume under consideration at timestep i, (min_x, min_y, min_z) and (max_x, max_y, max_z) represent the estimated tracking window in which the feature could lie and $ED(S^{t_i}_{xyz}, S^{ti+1}_{pqr})$ is the Euclidean distance function between the feature under consideration and the current location within the tracking window.

3.4 Effective Visualization

An in-depth analysis of changes and variations within time-varying data is difficult to achieve with standard visualization techniques. Traditional visualization techniques present several limitations, especially

Fig. 1. Five different timesteps within the synthetic data.

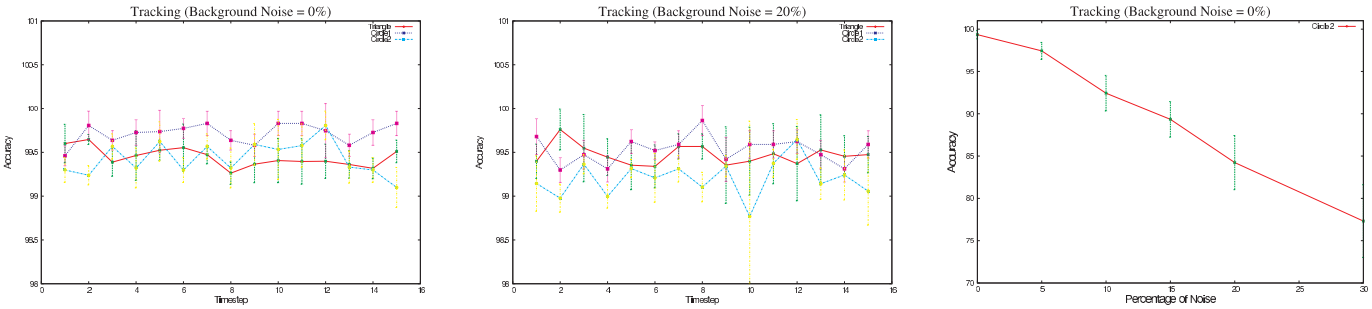


Fig. 2. Plots showing the accuracy of our tracking technique. (a) Accuracy in volumes without background noise. (b) Accuracy in volumes with 20% background noise. (c) Accuracy decrease as we increase textural noise.

in emphasizing time-varying features that move independently and at different speeds throughout the entire volume space.

Current visualization techniques that have been enhanced with feature tracking results make it possible for domain experts to better understand, visualize, and analyze the transformations that specific features undergo over time. We present three different ways for a user to visualize specific features of interest.

The simplest visualization and illustration technique is to create a path that can be annotated by combining the location of each independent feature of interest over time. This is a convenient method for adding additional information and cues to traditional visualizations.

The second visualization alternative is to view the feature of interest in isolation as it undergoes transformations. To accomplish this, our system visualizes each feature as an individual cutaway for which it can provide snapshots or animation of that feature's transformation over time.

The third visualization alternative is volumetric illustration, which has proven to be extremely effective. In particular, two-level volume rendering allows the user to visualize a feature within an entire volume [6] in two specific ways. First, specific segments of the volume can be highlighted or rendered in a way that draws the user's attention to particular features or regions of interest. Additionally, structures and features that surround features of interest can be illustrated for context.

4 Results on Synthetic Data

The strengths and limitations of our texture-based feature tracking technique have been tested with synthetic data. We have generated a time-varying dataset with 15 timesteps and a set of objects that independently move within the volume space without following any specific pattern. Each feature within the volume space has a particular pattern or texture used to characterize and identify it. Our synthetic data testing validated that our method worked even when overlap between features in subsequent timesteps did not exist and the shape and size of features did not remain constant. To test our system and its performance in the worst possible cases, random noise was added to the volumetric data, nothing was assumed about the possible motion of voxels, and the entire volume was used as the tracking window.

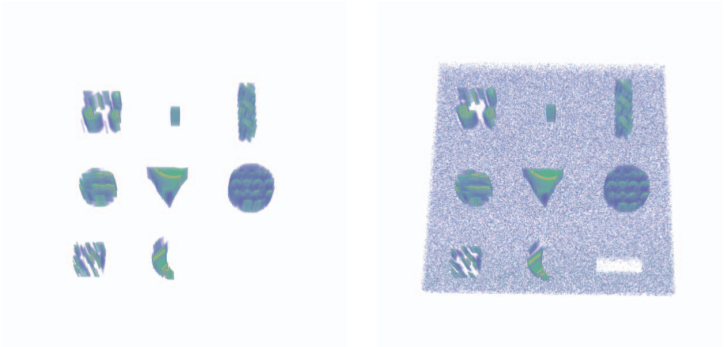
Figure 1 shows a screenshot of five different timesteps within our synthetic data. Observe how the two cylinders/circles move, the center triangle rotates, and the upper cube/square changes in size without any specific transformation occurring.

Our first experiment with the synthetic data was to automatically track three different objects over time. The three objects under consideration moved randomly and in most of the timesteps any of the features overlapped itself in consecutive timesteps. Knowing the center and location of each object of the synthetic data, we were able to measure the accuracy of our tracking results by computing the distance between the resulting location and the "real" position. This experiment demonstrated a 99.43% accuracy on average, thus confirming the strength of our technique. Figure 2a shows tracking accuracy in different timesteps and the errors that occurred over the 15 time steps through which we tracked each feature.

4.1 Background Noise

To test the accuracy of our technique with noisy data, we added random background noise to the synthetic volume. Figure 3 shows the synthetic volume before and after a 20% addition of random background noise. After running the texture-based tracking technique five separate times on three individual objects, we demonstrated that background noise has minimal effect on the accuracy of our technique. On average, this experiment demonstrated a 99.20% accuracy in finding the location and centroid of the feature under consideration. The performance and accuracy of our tracking technique augmented with 20% background noise is illustrated in Figure 2b. In this background noise experiment, we also found that actual noise minimally affected features with the textural properties of each remaining almost constant.

By tracking a set of individual features in synthetic volume space over time, we demonstrated that it is possible to annotate data and illustrate volume to obtain more accuracy regarding specific motions, changes, and movements. Figure 4 depicts the paths followed by three



Noise = 0% Noise = 20%

Fig. 3. (a) Original volume (b) Resulting volume after a 20% background noise was added.

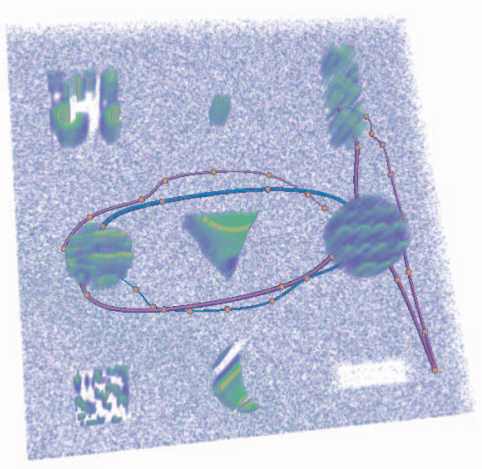


Fig. 4. From the tracking results it is possible to illustrate the path followed by any independent features and to show the actual direction by growing the line's width over time.

individual features over the 15 timesteps of the time-varying dataset used. The changes seen over time in line width allow accurate illustration of the specific path each feature follows over time. A primary advantage of our system is its flexibility in relation to annotation of tracks, paths, and changes within volumetric data. Such flexible annotation provides cues and insights about specific changes that visualization techniques and 2D/3D animations currently in use do not provide.

4.2 Texture Noise and Variation

We further tested the efficacy of our technique with textural noise and variation by adding random noise to the moving features. By inserting random noise, we found that within the texture properties and patterns of individual features variations and changes did occur. Figure 5 shows the patterns that resulted after 20% of textural noise was added to three features. By running our texture-based tracking technique with percentages of textural noise that ranged from 0% to 30%, we found that our technique has a 25% tolerance. That is, we found that by limiting the maximum tracking error to only one object where the resulting centroid was at most half of the object size away from the "real" location, acceptable limits exist when textural noise and variation is less or equal than 25%. Figure 2c illustrates how the accuracy of our tracking results decreases when textural noise and variation are increased.

The computed tolerance of 25% should not be taken, however, as definitively binding to our system. By analyzing and filtering input data and by weighting the multi-dimensional feature vector, it is possible to increase the accuracy of our tracking technique even when it

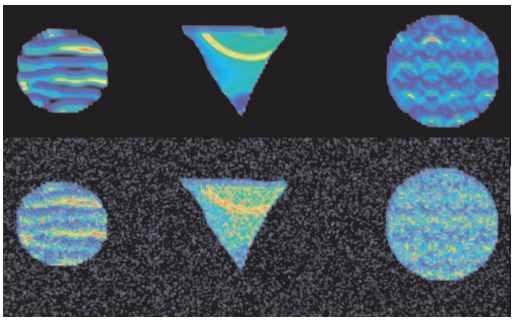


Fig. 5. Patterns before and after a 20% of textural noise and 5% background noise was added.

Feature Size (Subvolume)	Tracking Window Size	Tracking Time (sec)
10^3	10^3	0.000173
10^3	20^3	0.186
10^3	32^3	1.613
10^3	48^3	6.657
10^3	64^3	14.800
20^3	20^3	0.00107
20^3	32^3	2.248
20^3	48^3	22.048
20^3	64^3	66.351

Table 2. Performance of our texture-based feature tracking technique.

operates within noisy data. Of note, though, is that the specific filtering and weighting functions necessary to secure the overall accuracy of our system with noisy volumes are data and domain specific.

4.3 Time Analysis

Even with the worst possible scenario of background noise, textural noise, and complete volume (256x256x36) as the tracking window, on average our system required 20.43 seconds (+-1.21sec) to find the three objects in consecutive volumes. It is important to mention that in this worst case scenario experiment, the same dataset and same object were tracked in a fraction of a second by using the underlying dynamics of the volumetric data as explained in Section 3.3.

To better understand, analyze, and test the time required by our texture-based tracking technique, we extended the synthetic data so that we could measure the time required to find a specific feature within different track windows. Table 2 shows the time required to track a specific feature within different tracking windows. Note that the tracking of features within small tracking windows is efficient and can be done in a fraction of a second. However, when relatively large tracking windows are used, the time required to converge and return the best match significantly increases.

5 Application Domain: Hurricane Data

Visualization is a crucial step in the process of understanding and analyzing actual hurricane data. The development of our technique has been much enhanced by the knowledge of a domain expert experienced in the study of dynamic changes and variations found in the time-varying data of hurricanes. This expert's ability to compare relative differences of subsequent hurricane timesteps proved extremely important for analysis of changes within the hurricane.

The primary challenges in visualizing hurricane data are that the features are moving at varying speeds and in different directions. Elements, such as clouds, vortex towers, and rain bands, move throughout the volume space without following any particular patterns. Accurately tracking hurricane features within time varying-data is critical to annotating and illustrating hurricane changes.

Our results indicate that our texture-based tracking technique provides a better understanding of the internal dynamic changes happening over time within hurricanes. In this section, we discuss the applicability and efficacy of our technique to track, visualize, and analyze hurricane data.

5.1 Feature Selection

Generally, time-varying hurricane data are represented by dense arrangements of voxels in which each particle moves independently and at different speeds. We have extended our feature selection technique presented in Section 3.2 to automatically determine features of interest within hurricane data and to estimate a tracking window. Our feature selection approach consists of three steps.

During the first step, a given hurricane volume V_{t1} composed of Z images is divided into K groups. For each group or subvolume $S_i \in \frac{Z}{K}$, the maximum intensity projection image I_i is computed. During the second step, each image I_i is thresholded and a new image I_{ti} with a different window level is generated. Finally, the derivative (I_{di}) of each thresholded image I_{ti} is computed and used to determine features of interest.

The general concept behind our feature selection process is that each thresholded image I_{ti} is comprised of a summary of the features found within each particular segment of the subvolume under consideration. This is, finding specific features of interest within 3D subvolumes is a difficult task due to the amount of information in each subvolume. Similarly, finding features of interest within a single slice per subvolume does not provide enough information to decide which are important features. By computing the maximum intensity projection (MIP) image of the subvolume, our system produces a 2D image that more accurately summarizes and shows the underlying important features within the subvolume. Furthermore, changing the window level and thresholding the image allows us to highlight the specific features that contain the most information. Then, by translating back into 3D space, we have a 3D feature of interest.

The important features found within the thresholded MIP image are also used to better guide movement of the tracking window. In the particular situation of hurricanes, our technique increases the accuracy with which each feature is detected because of its ability to analyze all three images (I_i, I_{ti}, and I_{di}) before translating the tracking window. Figure 6 shows the pipeline we designed to automatically locate and select features of interest and to translate the tracking window in hurricane time-varying data.

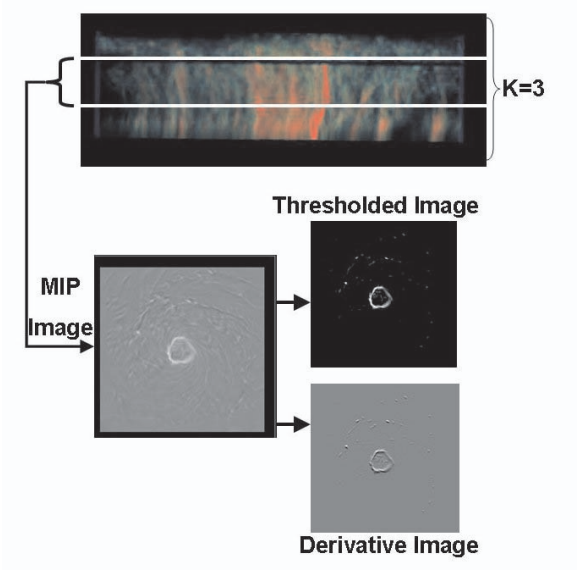


Fig. 6. A thresholded image of a maximum intensity projection image of a subvolume under consideration is used to find features of interest.

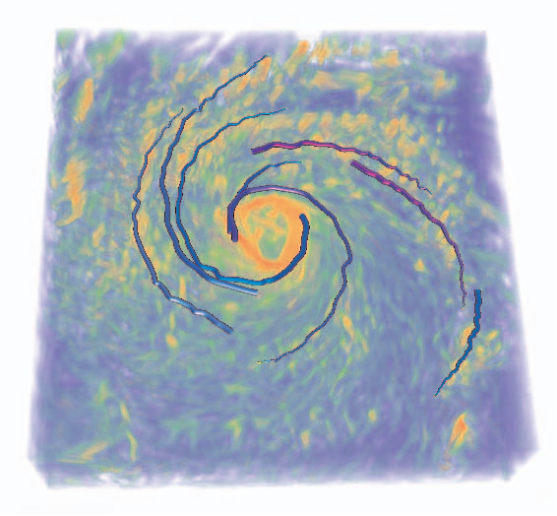
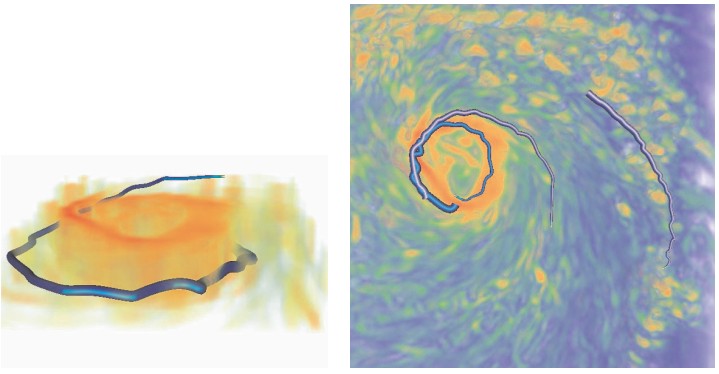


Fig. 7. Resulting path after tracking a number of independent features over time.



(a) (b)

Fig. 8. (a)Results of tracking a hurricane feature within 30 timesteps. (b) Path followed by three independent features over the same period of time.

5.2 Effective Visualization

We have tested our feature tracking technique with hurricane data from simulations of two dramatic atmospheric events: Hurricane Bonnie and Hurricane Katrina. This data was generated by researchers at NASA and shared with our collaborators in the UMBC Atmospheric Physics Department. To effectively illustrate feature changes and variations over time, we first computed a set of features of interest following the approach presented in section 5.1. The software, then, automatically tracked each feature independently. During the automatic tracking process, the characteristic properties of each feature were updated to compensate for structural variation and changes occurring over time. Since hurricane features can disappear or merge with other neighboring features, a maximum possible difference threshold value was defined between timesteps. If that maximum tolerance was met, the specific feature under consideration was not longer tracked.

Figure 7 shows results obtained through tracking eight independent features. From these results it is possible to see that most of the features under consideration followed a clear pattern such as moving counter-clockwise and inwards. Furthermore, when the tracking is limited to a particular number of timesteps, it is possible to see that features closer to the hurricane's center moved faster than features farther away from the hurricane's eye. Both of these findings are extremely important in terms of the analysis and interpretation of hurricane data.

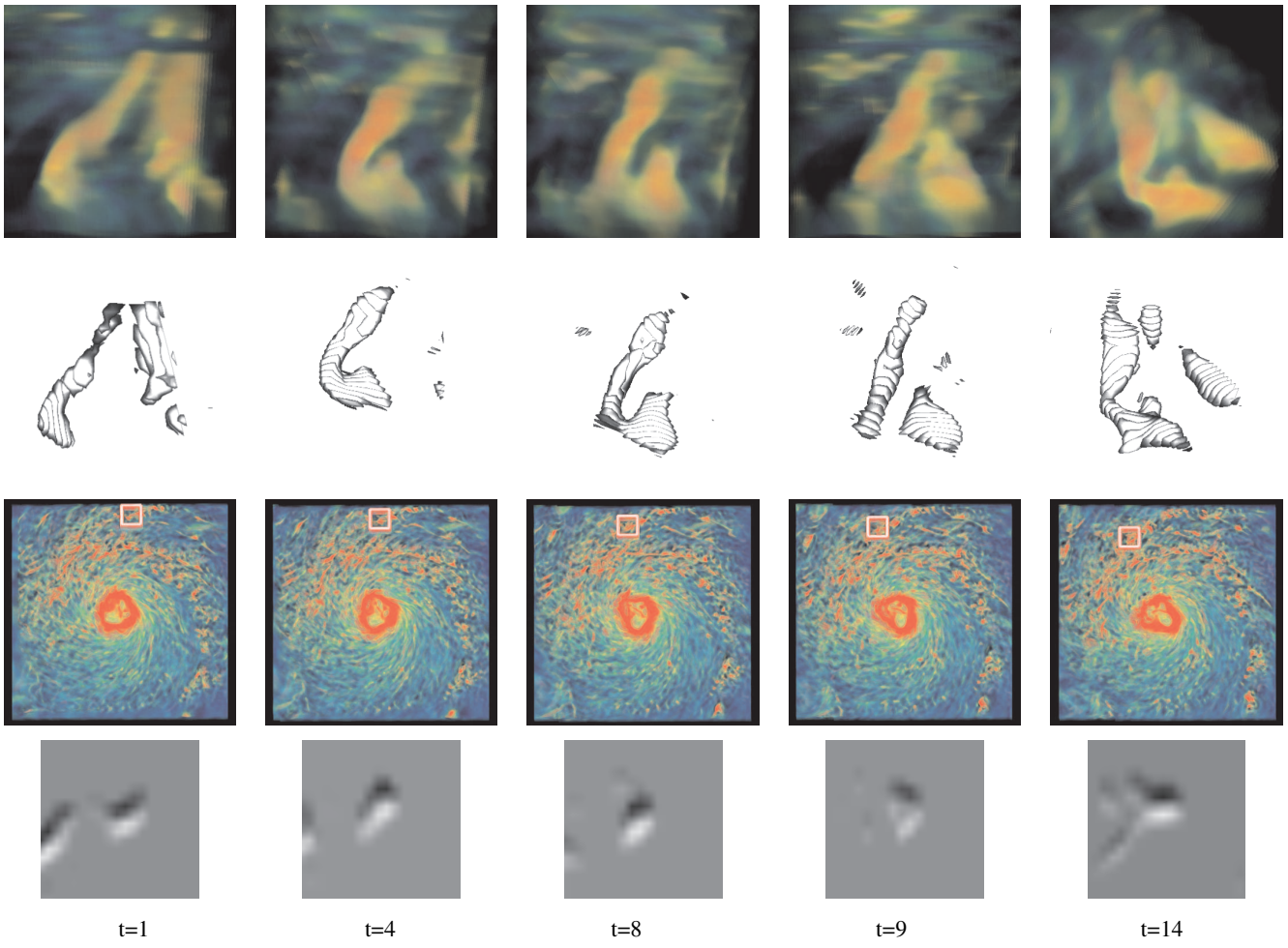


Fig. 9. Visualization and illustration of a specific hurricane feature over time. Our technique provides the flexibility to track, isolate, and illustrate specific feature changes and variations over time. The set of 2D images show the resulting thresholded MIP images used to better estimate the tracking window.

Figure 8b shows the path followed by three independent features over the same period of time. After tracking those three specific features within the first 26 timesteps in hurricane Bonnie, we found that features closer to the hurricane's eye rotated 360 degrees while features farther from the center rotated less than 90 degrees during the same time period of 78 minutes. Such results make it is possible to determine the rotational speed differences around the hurricane and to compute a feature's movement and speed towards the interior of the hurricane.

To better understand the intensification process of hurricanes, our collaborators propose focusing on the change undergone by the hurricane from one timestep to another. In general, the change undergone by a hurricane is has been calculated by taking a voxel-wise difference operation and then visualizing difference. Due to the fact that a hurricane rotates even as the center of its eye moves over time, such difference operations applications are limited in their use. Before a meaningful difference between two consecutive timesteps can be computed, the hurricanes have to be registered. We tested a number of registration techniques, several of which were capable of applying global transformations. We also found that rigid registration techniques lack accuracy due to the dynamic changes and variations that take place within hurricane features and are reflected in data. Also, we found that deformable registration techniques were unable to deal with registering different timesteps due to the uncertainty found within the data.

Based on our tracking results we determined that instead of registering the entire volume, we should isolate and segment out specific hurricane features and visualize them over time. We consider each target feature as an independent cutaway feature and thus provide the viewer with snapshots, animations, and illustrations of the transformation of these cutaways over time. The dynamic nature of hurricane data makes hurricane features extremely hard to segment and separate. However, from our accumulated tracking information, we can create a subvolume around a specific feature and then segment out each subvolume. Computing that subvolume over the track timesteps allows us to cut away a specific feature and visualize changes and variations over time.

Illustrative visualization has proven to be extremely effective in a wide range of medical and scientific domains. Our research indicates the specific, particular structural changes and variations occurring over time within each target feature are less clear when visualized than when illustrated. Figure 9 shows a hurricane feature being tracked over time. After cutting away the specific feature, we can visualize its changes over the first 15 timesteps of the data. We believe that such specific feature segmentation will provide crucial information to domain experts seeking to better understand the dynamics happening between timesteps.

Additionally, we have applied the two-level rendering approach as a visualization technique in regards to hurricanes. By illustrating the hurricane and rendering the specific feature under consideration, it is possible to draw the viewer's attention to a specific region of interest while also providing the context of the hurricane. Figure 10 shows an illustration of the eye of a hurricane. Applying boundary and silhouette enhancement techniques [11] highlights the overall structure of the hurricanes. Furthermore, when we combined illustration techniques

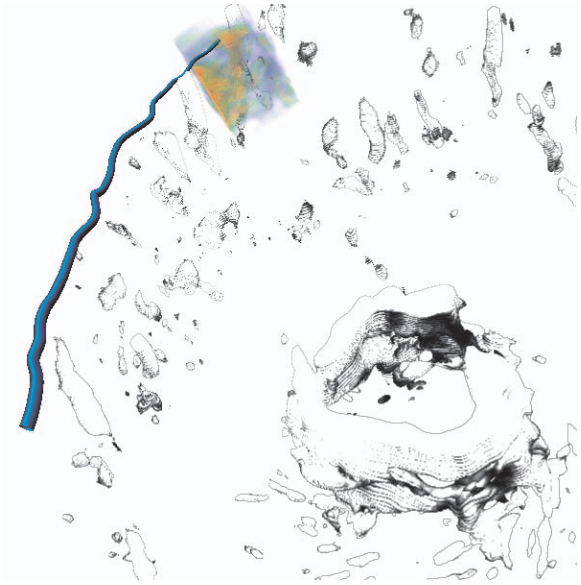


Fig. 10. Direct volume rendering of the hurricane feature under consideration within the illustration of the overall structure of the hurricane.

with direct volume rendering, it was possible to show the volumetric data of a target feature while illustrating the rest of the hurricane. In particular this method permits, the feature of interest to be visualized with full detail while the rest of the hurricane is visualized in an illustrative manner that provides context around the targeted feature.

6 Discussion

Accurate tracking, visualization, and illustration of specific features within time-varying data still remains an open research topic. Over the last several years a number of researchers have proposed different techniques for tracking features within volumetric time-varying data. However, assumptions made in previous work limited the specific type of data and applications addressed. For instance, the three key observation and assumptions made by Silver and Wang[13] about overlapping features and the basic assumption made by Ji et al. [7] about creating and following an isosurface between timesteps can only be applied to data with high temporal resolution and sampling. Our technique is not based on those assumptions and is capable of working with both low-temporal and high-temporal resolution in time-varying volumes.

A significant advantage of our technique is that it is an extensible approach that can be used across multiple domains and with different data formats. The multi-dimensional feature vector that characterizes each feature of interest has the potential to be extended for even better tracking of particular objects. For instance, we propose that if specific features can be distinguished by a their particular shape, a 3D shape descriptor added to the multi-dimensional feature vector would permit more accurate differentiation and tracking of features over time.

A limitation of our technique is the "drifting problem" which exists when small errors are introduced to our texture-based multi-dimensional feature vector over time. As part of our future work, we would like to explore how a statistical update approach similar to the one proposed by Matthews et al. [8] can be used to reduce such cumulative errors.

7 Conclusion

In this paper we have introduced a texture-based feature tracking technique that overcomes some current limitations that hamper the illustration and visualization of dynamic changes within time-varying volumetric data. Experiments and results with synthetic and real-world hurricane data demonstrate the usefulness and effectiveness of our texture-based tracking technique. Furthermore our results show that our texture-based tracking can lead to insightful visualization, illustration, annotations, and segmentation of specific features over time.

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments and suggestions. We would like to thank our collaborators, Lynn Sparling and Sam Trahan of the UMBC Physics Department, for a fruitful partnership, Scott Braun from NASA for providing hurricane data, and Rosemary Klein for her help and support in writing this paper. This work was supported in part by NSF grant 0121288.

References

[1] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, 2001.

[2] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 675–682, 1998.

[3] C. I. Christodoulou, C. S. Pattichis, M. Pantziaris, and A. Nicolaides. Texture-based classification of atherosclerotic carotid plaques. volume 22, pages 902–912, July 2003.

[4] X. Guan, G. Pan, and Z. Wu. Automatic categorization of traditional chinese painting images with statistical gabor feature and color feature. In *Lectures in Computer Science*, pages 743–750. Springer Berlin/Heidelberg, 2005.

[5] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 3, pages 610–621, 1973.

[6] H. Hauser, L. Mroz, G. I. Bischi, and M. E. Gröller. Two-level volume rendering. volume 7, pages 242–252, 2001.

[7] G. Ji, H.-W. Shen, and R. Wenger. Volume tracking using higher dimensional isosurfacing. In *IEEE Visualization '03*, pages 209–216, 2003.

[8] L. Matthews, T. Ishikawa, and S. Baker. The template update problem. In *IEEE Pattern Analysis and Machine Intelligence*, volume 26, pages 810–815, June 2004.

[9] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 27, pages 1226–1238, 2005.

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

[11] P. Rheingans and D. Ebert. Volume illustration: Non-photorealistic rendering of volume models. In *IEEE Transactions on Visualization and Computer Graphics*, volume 7(3), pages 253–264, 2001.

[12] R. Samtaney, D. Silver, N. Zabusky, and J. Cao. Visualizing features and tracking their evolution. volume 27, pages 20–27. IEEE Computer Society Press, 1994.

[13] D. Silver and X. Wang. Volume tracking. In R. Yagel and G. M. Nielson, editors, *IEEE Visualization '96*, pages 157–164, 1996.

[14] D. Silver and X. Wang. Tracking scalar features in unstructured datasets. In *IEEE Visualization '98*, pages 79–86, 1998.

[15] X. Tang. Texture information in run-length matrices. In *IEEE Transactions of Image Processing*, volume 7, pages 1602–1609, 1998.

[16] F.-Y. Tzeng and K.-L. Ma. Intelligent feature extraction and tracking for large-scale 4d flow simulations. In *IEEE Supercomputing Conference*, pages 8–15, 2005.

[17] K. Wang, H. Qin, P. R. Fisher, and W. Zhao. Automatic registration of mammograms using texture-based anisotropic features. In *Proceedings of 2006 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 864–867, 2006.

[18] H. E. Willoughby. Atmosphere: Forecasting hurricane intensity and impacts. In *Science*, volume 315, page 1232, 2007.

[19] Y. Xu, M. Sonka, G. McLennan, J. Guo, and E. A. Hoffman. MDCT-based 3-D texture classification of emphysema and early smoking related lung pathologies. volume 25, pages 464–475, 2006.