DIFFERENTIATING AND MODELING MULTIPLE MOVING OBJECTS IN MOTION IMAGERY DATASETS

Peggy Agouris*, Panos Partsinevelos, Anthony Stefanidis

Department of Spatial Information Science & Engineering, National Center for Geographic Information & Analysis
University of Maine, 348 Boardman Hall, Orono, ME 04469-5711, USA, {peggy, panos, tony}@spatial.maine.edu

Commission V, WG V/5

KEY WORDS: Video Analysis, Self-Organizing Maps, Trajectories, Spatiotemporal Modeling

ABSTRACT:

Motion imagery datasets capture evolving phenomena like the movement of a car or the progress of a natural disaster at video or quasi-video rates. The identification of individual spatiotemporal trajectories from such datasets is farm from trivial when these trajectories intersect in space, time, or attributes. In this paper we present our approach to this problem and relevant algorithms. A key component of our work is the attribute classification (AtC) strategy, a novel approach to classify individual trajectories using a sequence of image processing and neural network tools. Geometry, k-means clustering, backpropagation and self-organizing maps are the tools applied towards the classification of such datasets. Other key components of our approach include the novel g-SOM approach to generalize spatiotemporal datasets, and the concept of spatiotemporal helixes, used to model the behavior of individual objects. In this paper we present these key components of our approach and some experimental results.

1. INTRODUCTION

Motion imagery (MI) datasets capture evolving phenomena like the movement of a car or the progress of a natural disaster. Depending on the nature of the observed phenomenon MI datasets may be video feeds, or sequences of still imagery at distinct intervals. Furthermore, their content may be quite dense, with numerous objects moving in the monitored area. The identification and modelling of individual trajectories in MI datasets represents a substantial challenge for the photogrammetric and computer vision communities. Interesting work in this field has been performed in both video processing and spatiotemporal analysis domains. In the trajectory domain, there is work for spatio-temporal synthetic dataset generation to simulate movement trajectories (Pfoser & Theodoridis, 2000). In (Bradshaw et al., 1999) Kalman filters are used to describe the motion and real time trajectory acquisition. In (Bremond & Medioni, 1998) a system is used to extract and recognize moving objects, as well as to classify the motion by modeling scenarios. Sorting data according to their spatial occupancy through tree structures is a regularly proposed data manipulation scheme (Sellis et al., 1987). Interesting work on indexing animated objects is reported in (Kollios et al., 1999; Vazirgiannis & Wolfson, 2001).

This paper addresses the classification and modeling of moving object trajectories from an input video dataset. By the term classification we imply the identification of all instances of the same object and their linking into individual trajectories. This is often treated as a clustering problem, and common approaches include the use of k-means clustering and its variations (Hartigan, 1975), self-organizing maps (Kohonen, 1977), and neural networks (Sweeny et al. 1994). In recent years clustering is application oriented. Thus, research focuses on forming variations of clustering techniques according to specific applications and dataset formations (Ng & Han, 1994).

This paper is organized as follows. In Section 2 we present an overview of our approach to video analysis and trajectory modeling, followed by a discussion of the involved datasets in Section 3. Section 4 presents our attribute classification strategy for the identification of individual trajectories from a complex input video signal. Section 5 offers an overview of our g-SOM algorithm for the generalization of spatiotemporal trajectories, while Section 6 presents the concept of spatiotemporal helixes to model this information. We conclude with some experiments (Section 7) and future plans (Section 8).

2. APPROACH OVERVIEW

Motion and video imagery are emerging as major sources for intelligence information, especially in rapidly evolving operations. Novel deployment techniques, e.g. video sensors onboard unmanned aerial vehicles (UAVs), and distributed sensor networks are supporting the collection of timely, geospatially registered information, enabling the precise monitoring of mobile objects. This transition from static to motion imagery is introducing substantial challenges related to the large amounts of data involved, and the corresponding processing requirements.

Our approach to motion imagery analysis and modeling is a three-stage process (Fig. 1):

- The first stage of our approach uses as input the video feed and identifies in it a number of individual trajectories. A combination of accumulative frame differencing (AFD), morphological filtering, and attribute classification (AtC) solutions performs the identification of individual object

* Corresponding author.

trajectories using as input a video signal depicting multiple moving objects.

- During the second stage of our approach each trajectory is generalized through an extension of self-organizing maps (SOM), identifying critical nodes on them.
- Finally, the innovative concept of the *spatiotemporal helix* serves as a motion indexing mechanism to describe the motion patterns of individual objects, and to classify and differentiate the trajectories of different types of objects.

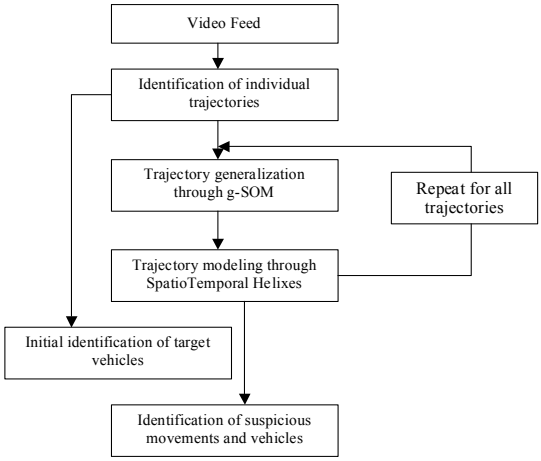

Figure 1. Video analysis framework.

In this paper we present an overview of our activities in these three stages, with more emphasis put on the first stage, and specifically our AtC strategy.

3. FROM VIDEO TO TRAJECTORIES

The goal of the first stage of our approach is to detect motion in selected movie scenes. To detect motion we begin with a standard accumulative frame differencing (AFD) algorithm. The parameters for this task include the gray value difference threshold, and the number of frames used for our analysis, Both parameters can be initialized at default values (based on anticipated velocity and radiometric conditions) that can be subsequently adapted based on the analysis of the results. The output of AFD is a binary movie of motion. An example frame of such a binary motion movie can be seen in Figure 2 (top). One can clearly distinguish the leading and trailing edges of a person moving through the scene. This is obviously due to the use of absolute value differences in our AFD approach. In order to link these two motion components we proceed with a morphological operation. More specifically, a combination of dilation-and-erosion (D&E) produces a solid cluster representing the movement of an object during this sequence of frames (Fig. 2, bottom). A filtering algorithm allows us to eliminate useless information by using pre-specified ranges of size (and/or other parameters like eccentricity) values. In this manner we can also distinguish for example motion data for people and cars.

Spatiotemporal trajectories can be best visualized by making use of the 3-dimensional *(x,y,t)* spatiotemporal domain of a scene, comprising two *(x, y)* spatial dimensions representing the horizontal plane, and one *(t)* temporal dimension. The complex trajectory of an object over time is described as the union of its locations over time. While Fig. 2 shows the results for two clearly distinct moving objects (the second one is on the left hand side of the field), in general we can have numerous objects moving within the sensor field of view. These moving objects can produce numerous, often entangled trajectories over the extent of a brief video segment.

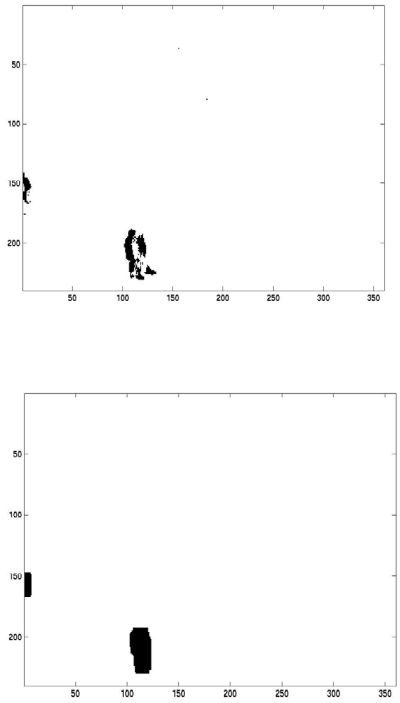

Figure 2: AFD result (top), followed by dilation-and-erosion (bottom).

We assume as input dataset for our AtC solution information that has the form of *(x,y,t,c,s)*, where x,y,t are the S-T coordinates of the trajectory, c is radiometry ranging from 0-255 (for common grayscale images), and s is the size of the object, defined by the number of patch pixels describing the object. All attributes associated with an object may be somewhat imprecise, due to common problems like occlusions, noise etc. Thus, patches corresponding to the same object may still vary in size and/or color.

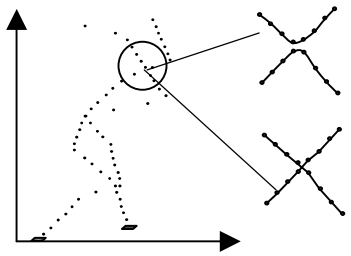

Figure 3. Potential misclassification of trajectories.

Let us consider a pair of trajectories. If they are distinct in space, time or attributes, classification could be easily completed through proximity analysis. On the other hand, if they are entangled as shown in figure 3, the classification of each point to its corresponding trajectory is not a trivial task. The proximity of locations and/or attributes causes pure geometric analysis using neighboring distance metrics to be often inadequate. For the two segments on the crossing of the

trajectories of figure 1 we cannot be sure whether the trajectories follow the almost straight path or the curved one.

At this point since the focus is on the classification of trajectories we consider few moving objects in our scene, and also consider common non-deformable moving objects (e.g. cars).

4. ATTRIBUTE CLASSIFICATION STRATEGY

The analysis introduced in this section focuses on the differentiation of trajectories, based on both spatio-temporal and attribute coordinates. Our attribute classification (AtC) strategy comprises the three sub-processes outlined in Fig. 4.

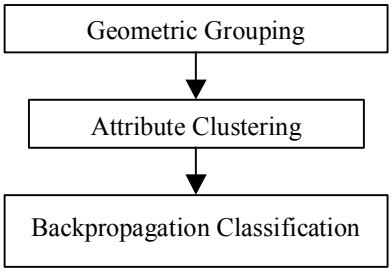

Figure 4. Attribute classification (AtC) Strategy.

4.1 Geometric Grouping

During geometric grouping we separate groups of points that are spatio-temporally distinct from any other group. This is accomplished by imposing a distance threshold to each pair of points. Points form groups as long as they are farther than the threshold distance from corresponding points of the same temporal instance, and are close to each other in the temporal direction.

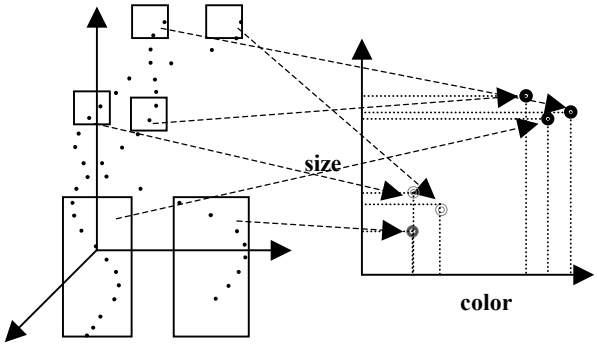

Figure 5. Branch formation and attribute mapping.

A temporal proximity is also required to describe the connection between the processed points. The groups of points are considered the base units over which the first step of classification takes place and they are termed as 'branches', shown in figure 5, left (boxes). There is strong confidence that the points included in each branch, belong to the same trajectory even though we do not yet know to which one.

4.2 Attribute Clustering through k-means

During *attribute clustering* we identify correspondences among the groups selected in the previous step. Our aim is to link groups that belong to the trajectory of the same object. We accomplish this goal by making use of the attribute space. Size and radiometry (color or gray values) are the main attributes we consider, but additional attributes may also be used (e.g. eccentricity). Fig. 5 (right) shows the transformation of the box contents onto a color-size attribute domain. In this attribute domain all instances of the same object will form compact clusters. Thus, group linking in the spatiotemporal domain becomes a clustering problem in the attribute domain. As long as the attributes of different objects are at least partially different, they can be differentiated in the attribute space.

The next step is to identify clusters in the attribute space data. The number of clusters corresponds to the number of distinct objects captured in our input video. This task is accomplished by utilizing a simple SOM or k-means algorithm that takes into account not only the separability of the data but also the neighboring of each point to the others. The result is shown in figure 6. Two nodes initialize the algorithm and after the iterative process they converge to the center of the two formed clusters each representing three attribute pairs. The branches are now classified and related to the trajectory they belong to (marked by white and black dots).

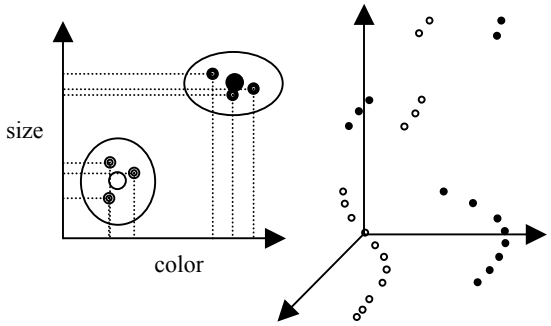

Figure 6. Attribute clustering and branch classification.

The remaining points, namely the ones near the intersections between the trajectories are still unclassified. Their classification is accomplished by utilizing a backpropagation neural network.

4.3 Backpropagation Classification

During *backpropagation classification* we use the information provided by the labeling of the distinct groups (accomplished through geometric grouping and attribute clustering) to assign a single trajectory label to each and every cluster of pixels in our input spatiotemporal dataset. We accomplish this task using a backpropagation neural network (BNN).

The backpropagation neural network (NN) (Haykin, 1999) is a broadly used neural network that is applied to numerous diverse applications. Its basic concept is that an input space gets connected with an output space through a series of synaptic neurons that form sets of hidden layers.

In our case the goal is to classify a multidimensional dataset into separable classes. The input space is five-dimensional as previously described (corresponding to the five coordinates identified in Section 3) and the output dimension equals the number of objects in the scene, as it was determined during attribute clustering. There is a need for training of the network in order to learn the specific classification task. Therefore, a set of correct classified input-output relations is required. This training phase would adjust the values of weights

and biases in the network and thus it is important to include as many correct training data as possible to accomplish the best possible performance. Our training space is formed by the already classified branch points of the trajectories, as described in the previous section. In many cases and according to the application at hand these already classified points comprise a large percentage of the original dataset and the performed classification is highly effective. The final classification step includes the separation of each classified point population into a different dataset, which most likely would be incomplete and would include some outliers-misclassified points. Many of the outliers can be easily removed according to simple proximity tests. Some additional post-processing procedures further enhance the classification confidence.

Experiments show that a two hidden layer network comprising of five and three nodes accordingly is capable to perform an adequate classification as shown in figure 4. A 5-dimensional input space results in a three possible output layer for a dataset comprising of three trajectories.

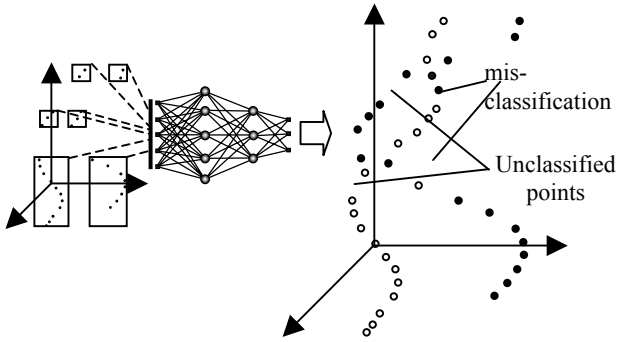

Figure 7. Backpropagation classification outcome.

5. TRAJECTORY GENERALIZATION THROUGH G-SOM

The process outlined in the previous section uses as input a video feed and produces a set of spatiotemporal trajectories corresponding to the objects moving in this feed. As video imagery is by its nature highly redundant, we need to reduce these trajectories to their essential information. This is a generalization task, and we approach it using a novel variation of Self-Organizing Maps (SOM).

The SOM algorithm (Kohonen, 1982) is a nonlinear and nonparametric regression solution to a class of vector quantization problems. It belongs to a distinct class of artificial neural networks (ANN) characterized by unsupervised and competitive learning. In this iterative clustering technique, cluster centers-nodes are spatially ordered in the network space $\Re_N$ in order to represent the input space $\Re_I$. The objective of the SOM is to define a mapping from $\Re_I^m$ onto $\Re_N^d$ where $m \geq d$. Applied to spatiotemporal trajectory data, it uses as input a large number of sequential points in the spatiotemporal (ST) domain, and distributes representative nodes to this input space so as to provide an abstract representation of it. Thus, it produces a generalized representation of the input space.

While standard SOM solutions provide adequate representations of relatively smooth lines (e.g. road networks (Doucette et al., 2000)), they are less successful when applied to complex spatiotemporal trajectories. As objects accelerate/decelerate, turn, or even stop, their spatiotemporal trajectories become fairly complex. This information is very important for intelligence analysis, as it signifies specific mobility patterns. Fig. 8 shows how a standard SOM solution fails at instances to capture the convoluted geometry of the input space. The generalized line (as it is delineated by the green nodes) is a polygonic approximation of the input space that misses some of the turns and corners of the actual trajectory.

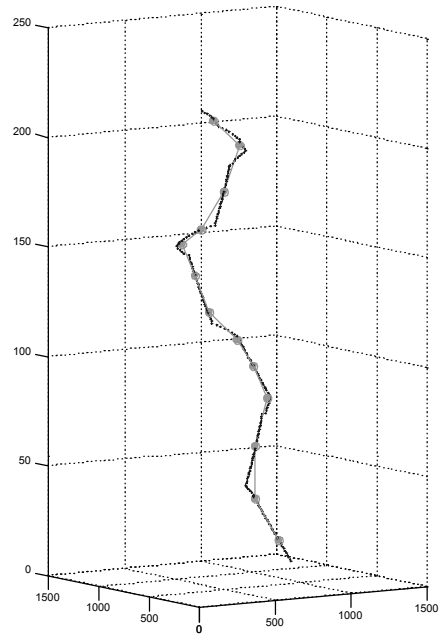

Figure 8. Standard SOM generalization of a trajectory.

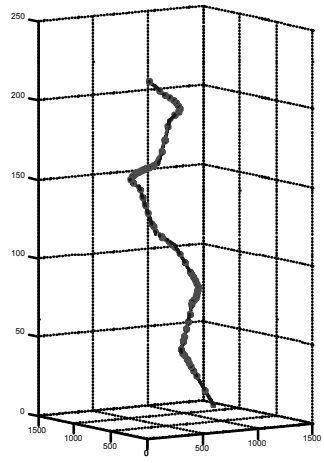

Figure 9. Application of g-SOM to generalize a trajectory.

In order to improve the use of SOM for spatiotemporal generalization we have to enable the SOM nodes to be distributed in a manner that captures the complexity of the analyzed trajectory. Complexity is defined in our context as the spatiotemporal variation of the moving object's behavior. In order to detect and quantify this variation we have introduced a novel variation of SOM that proceeds by:

- analyzing local variations of the input data to *identify* geometrically complex spots, and

densifying the number of local nodes used to represent such spots by adding a highly localized mini-SOM solution.

A detailed description of our geometrically-enhanced SOM (g-SOM) technique may be found in (Partsinevelos et al. 2001). Early experiments with g-SOM are confirming its superiority compared to the standard solution. Fig. 9 shows how the g-SOM solution uses more nodes to capture the complex spatiotemporal lifeline of Fig. 8.

6. TRAJECTORY MODELING THROUGH SPATIOTEMPORAL HELIXES

As mentioned above, the complex trajectory of an object over time is described as the union of its locations over time in a three-dimensional spatiotemporal domain (*x,y,t*). It can be visualized by piling the object's recorded positions on top of each other at the corresponding time instances. As an example, a circular object that remains stable will describe a cylinder in the spatiotemporal domain, while a rectangular object that is shrinking at a constant pace until it disappears will produce a pyramidal trace. The trajectories captured through the processes outlined in the previous two sections have to be indexed to support further analysis (e.g. comparison of trajectories to identify movement patterns common in a class of objects). Towards this goal we have introduced the innovative concept of the *spatiotemporal helix*.

We can identify two important types of geospatial information that describe the spatiotemporal behavior of an object: *movement* and *deformation*. First, the object *moves* changing its location with respect to an external reference frame. This information is represented by a trajectory describing the movement of the object's center of mass, as described in the previous two sections. The second type of spatiotemporal change refers to the object's internal reference frame and describes the variations over time of the object's shape. This change is represented through a set of vectors that pinpoint the placement, direction, and magnitude of the object's shape change.

We introduce the *spatiotemporal helix* (*STH*) as a compact description of an object's spatiotemporal variations. It comprises a central spine and annotated prongs. More specifically:

- The central *spine* models the spatiotemporal trajectory described by the center of the object as it moves during a temporal interval.
- The protruding *prongs* express deformation (expansion or collapse) of the object's outline at a specific time instance.

Fig. 10 is a visualization of the concept of the spatiotemporal helix. The spine is the vertical line that connects the nodes (marked as white circles), and the prongs are shown as arrows protruding from the spine, pointing away from or towards it. The gray blob at the base of the spine is the initial outline of the monitored object. The helix describes a movement of the object whereby the object's center follows the spine, and the outline is modified by the amounts indicated by the prongs at the corresponding temporal instances.

As a spatiotemporal trajectory, a *spine* is a sequence of (x,y,t) coordinates. It can be expressed in a concise manner as a sequence of spatiotemporal nodes $S(n^1,...n^n)$. These nodes correspond to breakpoints along this trajectory, namely points where the object accelerated/decelerated and/or changed its orientation. Accordingly, each node n^i is modeled as $n^i(x,y,t,q)$, where:

- *(x,y,t)* are the spatiotemporal coordinates of the node, and
- *q* is a qualifier classifying nodes as *acceleration* (q^a), *deceleration* (q^d), or *rotation* (q^r) ones.

The qualifier information *q* is derived by the local values of spine gradients. High values of the vertical gradient indicate acceleration or deceleration, while high values of the horizontal gradient indicate rotation. While this information is derivative of the other three values, it is considered semantically important for describing an object's behavior, and this is the reason we store it separately. These nodes of the STH spine are the points captured through the g-SOM algorithm (described in previous sections).

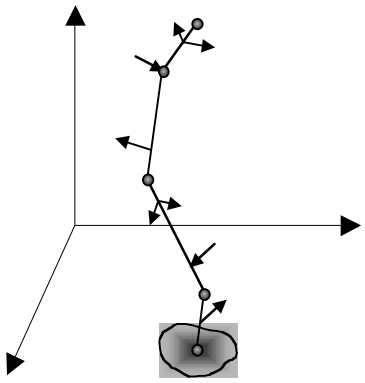

Figure 10. Visualization of the SpatioTemporal helix

7. EXPERIMENTS

In order to investigate the application of our attribute classification strategy we used two quite similar trajectories, and investigated the ability of our approach to distinguish them. In order to communicate the similarity between these two datasets we use a pair of attribute similarity percentages. The first percentage is the radiometric similarity between two trajectories, and the second percentage portrays the similarity in sizes of objects as they are captured throughout the MI dataset. For both percentages 0% corresponds to perfect dissimilarity and 100% corresponds to perfect similarity. Additionally, we used two error metrics to evaluate the results. First, the percentage of wrongly classified points shows how many of the classified points are assigned to the wrong class and is referenced to as PM. Second, the percentage of unclassified points shows how many points could not be assigned to any class (due to inherent ambiguities), and is referred to as PU.

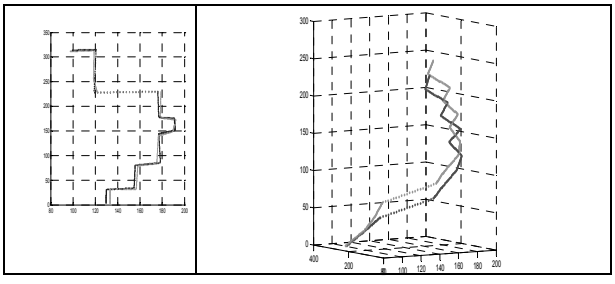

Figure 11. Input dataset for very close moving objects.

To compare our method with a common clustering classification we use as reference the c-means algorithm for both the 5-dimensional input space and the 2-dimensional attribute space. We use two different datasets (SET1 and SET2). In Fig. 6 we show the formed branches and their mapping on the attribute space for SET1. It represents an average situation, with two entangled trajectories. The resulting classification after the BP algorithm is shown in center, while the right side shows the results of a c-means algorithm. In Fig. 11 we show SET2. It represents an extreme situation, with two trajectories traveling in the same road, trailing each other by few instances. This is a situation that is quite difficult even for a human operator to discern, yet our algorithm shows excellent results.

For both datasets we introduced an increasing level of similarity in the two additional attributes (size and radiometry), ranging from 20 to 100%. In Table 1, we tabulate the results for these tests.

Set 1	After BP **(1-PU)**	After BP **(1-PM)**	Pure c-means **(1-PM)**
20,20	93	98	91
40,40	85	97	78
60,60	83	96	65
80,80	70	95	-
100,100	68	100	-
Set 2	After BP **(1-PU)**	After BP **(1-PM)**	Pure c-means **(1-PM)**
20,20	86	93	92
40,40	75	99	82
60,60	83	93	63
80,80	75	93	-
100,100	70	100	-

Table 1: Comparison between clustering algorithms.

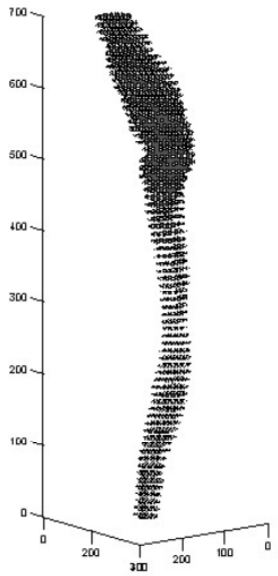

Figure 12: An evolving object

The first column shows the percentage of similarity in size and shape. The second column shows the percentage of classified points (1-PU) after our approach, the third column shows the percentage of correctly classified points (1-PM) after our approach, and the last column shows the same percentage (1-PM) for the c-means solution. Obviously, c-means by design will classify all input data, thus there is no meaning behind the 1-PU metric for this case.

The results tabulated in Table 1 show that our approach outperforms the c-means solution for both types of datasets. This is especially the case as the overlap in the two attributes increases. The c-means solution eventually collapses as overlap reaches very high values (80% or above), while our solution remains robust even under such unfavorable conditions.

It should also be noted that as shown in the Table the percentage of misclassifications in our approach remains controlled and low. Thus the produced results are of high accuracy.

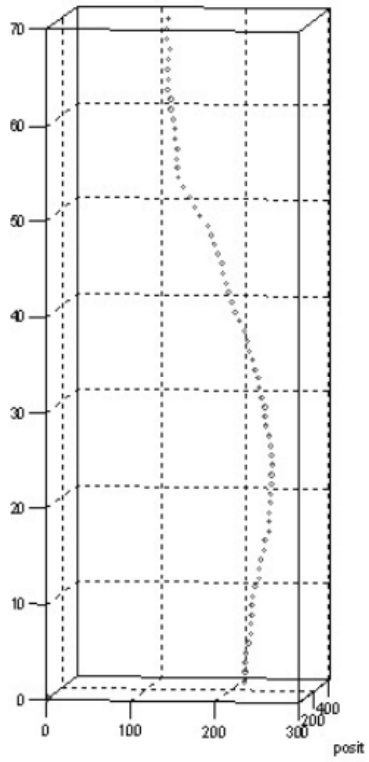

Figure 13: The spine of the S-T helix corresponding to the phenomenon of Fig. 12.

In Fig. 12 we show an evolving object extracted from a MI dataset, and in Fig. 13 the spine of its helix representation. A metric to describe how well a g-SOM extracted spine fits the original dataset is provided by the RMS distance between SOM nodes and actual data. Experiments with spatiotemporal trajectories indicate that the use of g-SOM for the generalization of spatiotemporal trajectories results in a reduction of this RMS metric by approximately 75% (Partsinevelos et al., 2001).

8. CONCLUSIONS

In this paper we presented an overview of our approach to motion imagery analysis for the identification and modelling of object trajectories. Our three-stage approach comprises innovative solutions to differentiate individual trajectories from

a complex video feed, generalize these trajectories, ad model them. We are currently working on developing similarity metrics to compare different spatiotemporal helixes. This will allow us to compare events as they develop, thus providing metric-quality potential for spatiotemporal analysis.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation through grants DGI-9983445, EPS-9983432, and IIS-0121269, and by the National Imagery and Mapping Agency under NURI Award NMA 401-02-1-2008.

REFERENCES

Bradshaw K., I. Reid and D. Murray, 1999. The Active Recovery of 3D Motion Trajectories and Their Use in Prediction, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3), pp. 219-234.

Bremond F. and G. Medioni, 1998. Scenario Recognition in Airborne Video Imagery, *IEEE Workshop on the Interpretation of Visual Motion*, Santa Barbara, CA.

Doucette P., P. Agouris, A. Stefanidis & M. Musavi, 2000. Self-Organized Clustering for Road Extraction in Classified Imagery, *ISPRS Journal of Photogrammetry and Remote Sensing*, Elsevier, 55(5-6), pp. 347-358.

Hartigan J., 1975. *Clustering Algorithms*, Wiley.

Haykin S., 1999. *Neural Networks*, Prentice Hall.

Kohonen T., 1977. *Self-Organizing Maps*, Springer-Verlag.

Kohonen, T., 1982. Self-Organized Formation of Topologically Correct Feature Maps, *Biological Cybernetics*, pp. 59-69.

Kollios G., D. Gunopulos and V.J. Tsotras, 1999. On Indexing Mobile Objects, *Proc. 18th Symposiium on Principles of Database Systems*, Philadelphia, PA, pp. 261-272.

Ng R.T., and J. Han, 1994. Efficient and Effective Clustering Methods for Spatial Data Mining, *20th Int. Conference on VLDB*, Santiago, pp. 12-15.

Partsinevelos P., A. Stefanidis & P. Agouris, 2001. Automated Spatiotemporal Scaling for Video Generalization, *IEEE-ICIP01 (International Conference on Image Processing),* Thessaloniki, Vol. 1, pp.177-180.

Pfoser D. and Y. Theodoridis, 2000. Generating Semantics-Based Trajectories of Moving Objects, *Int. Workshop on Emerging Technologies for Geo-Based Applications*, Ascona, Switzerland, pp. 59-76.

Sellis T., N. Roussopoulos and C. Faloutsos, 1987. The R+-tree: A Dynamic Index for Multi-Dimensional Objects, *Proc. 13th Int. Conference on VLDB*, Brighton, UK, pp. 507-518.

Sweeney W.P., M.T. Musavi, and J.N. Guidi, 1994. Classification of Chromosomes Using a Probabilistic Neural Network, *Cytometry*, 15(5).

Vazirgiannis M. and O. Wolfson, 2001. A Spatiotemporal Model and Language for Moving Objects on Road Networks, *Proc. 7th Int. Symposium Spatial & Temporal Databases*, Redondo Beach, CA, pp. 20-35.