SPATIOTEMPORAL ANALYSIS WITH ST HELIXES

Anthony Stefanidis Kristin Eickhorst Peggy Agouris

Dept. of Spatial Information Science and Engineering
348 Boardman Hall, University of Maine, USA
{tony, snoox, peggy}@spatial.maine.edu

Commission IV, WG IV/1

KEY WORDS: GIS, Modelling, Management, Database, Query, Multitemporal

ABSTRACT:

Efficient modelling of spatiotemporal change as it is depicted in multitemporal imagery is an important step towards the efficient analysis and management of large motion imagery (MI) datasets. Furthermore, the development of concise representation schemes of MI content is essential for the search, retrieval, interchange, query, and visualization of the information included in MI datasets. Towards this goal this paper deals with the concise modelling of spatiotemporal change as it is captured in collections of MI data, and the development of spatiotemporal similarity metrics to compare the evolution of different objects. Helixes represent both movement and deformation in a single concise model, and are therefore highly suitable to communicate the evolution of phenomena as they are captured e.g. in sequences of imagery. This integration of movement and deformation information in a single model is an extension of existing solutions, and is highly suitable for the summarization of motion imagery datasets, especially within the context of geospatial applications. In this paper we present the spatiotemporal helix model, its use to support spatiotemporal queries, and spatiotemporal similarity metrics for the comparison of helixes. These metrics allow us to compare the behavior of different objects over time, and express the degree of their similarity. To support these comparisons we have developed a set of mobility state transition (MST) cost metrics that express dissimilarity as a function of differences in state. In the full paper we present these models in detail, and proceed with experimental results to demonstrate their use in spatiotemporal analysis.

1. INTRODUCTION

The image processing community has been dealing with issues of object representation for many years. Of particular interest are techniques that model the changes that an object undergoes over time. Motion Imagery (MI) analysis makes use of video feeds or multitemporal sequences of static images, and thus is typically addressing object tracking over time. Storing such spatiotemporal information imposes obvious challenges related to the involved amount of data, and the complexity of spatiotemporal variations. Lifelines (Plaisant, Milash et al. 1996; Hornsby and Egenhofer 2002) and video summarization programs (Pope, Kumar et al. 1998; Zhou, Ong et al. 2000) represent some approaches developed in the GIS database and image processing communities to model spatiotemporal information.

In order to accommodate the particularities of motion imagery databases we have recently introduced the spatiotemporal (ST) helix as a modelling and generalization tool for spatiotemporal information handling (Stefanidis, Agouris et al. 2002; Agouris and Stefanidis 2003). Helixes differ from other approaches in that they not only capture the movement of an object's center of mass, but also incorporate information about changes in its outline. Furthermore, they allow us to identify critical instances in an object's history, and support metric analysis. Thus they are more than just a visualization mechanism, and incorporate databases and data storage techniques that allow the user to query for particular object behaviours. In this paper we provide an overview of our spatiotemporal helix model, and the ST comparison metrics we developed.

The paper is organized as follows. Section 2 of this paper details the spatiotemporal helix itself, and section 3 presents metrics for comparing multiple helixes. Helix aggregation is discussed in section 4. In Section 5 we demonstrate the implementation of the algorithms discussed in previous sections, and present experiments on noise removal, boundary reconstruction, and computational time for similarity queries. We conclude with our future plans.

2. THE SPATIOTEMPORAL HELIX

Spatiotemporal (ST) helixes are models of spatiotemporal variations. A spatiotemporal helix comprises a central spine and annotated prongs (Figure 1). More specifically:

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

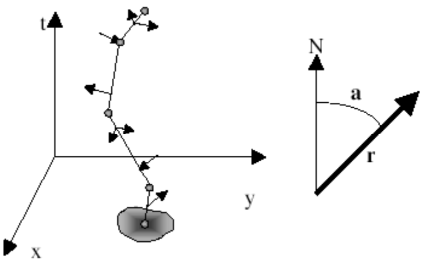

Figure 1: A spatiotemporal helix (left) and a detail showing the azimuth of a prong (right).

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

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

Each prong models the local expansion or collapse of the outline at the specific temporal instance when this event is detected, and is a horizontal arrow pointing away from or towards the spine. It is modelled as $p^i(t,r,a_1,a_2)$ where:

- *t* is the corresponding temporal instance (intersection of the prong and the spine in Fig. 2 left),
- *r* is the magnitude of this outline modification, expressed as a percentage of the distance between the center of the object and the outline, with positive numbers expressing expansion (arrows pointing *away from* the spine) and negative numbers indicating collapse (arrows pointing *towards* the spine),
- a_1, a_2 is the range of azimuths where this modification occurs; with each azimuth measured as a left-handle angle from the North (y) axis (Figure 1 right).

3. SIMILARITY METRICS

While a single helix conveys valuable information on the behavior of an object, it is the comparison of object behaviors that typically leads to knowledge discovery in typical geospatial applications. For example, comparing how two phenomena evolve may lead to the establishment of causality relationships. In order to support such applications we have developed metrics that support the comparison of ST helixes, to support the discovery of similarities or differences in object behaviours (Stefanidis et al., 2003).

We have developed abstract and qualitative metrics for helix comparisons. In such comparisons, one helix serves as reference and the second is a matching candidate (Figure 2). When examining a node on the reference helix, we do not simply look for a match at the same time instance on the candidate helix, but expand our time window to account for variations that may have occurred while obtaining the dataset. Thus we are not looking solely at time t2 for a match, but rather in an interval (t1,t3) (Fig. 2).

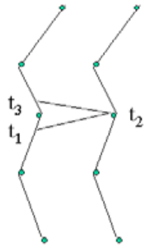

Figure 2: Comparison of reference helix (right) to candidate (left)

3.1 Abstract Comparisons

In abstract comparisons we are only considering the presence or absence of specific node and prong qualifiers. Two helixes are compared at each node in order to evaluate whether they exhibit similar behaviors. We proceed by assigning cost values to each nodal comparison. If both the reference and candidate helixes accelerate or decelerate at a given instant, then a cost value of 0 is given to that pair. If one is accelerating and the other is decelerating, a value of 2 is assigned (Table 1). Thus the comparison of highly dissimilar helixes produces high comparison results, while the comparison of two perfectly similar helixes produces a result of 0.

Helix 1 ↓ Helix 2 →	Accel.	Cons.	Decel.
Acceleration	0	1	2
Constant	1	0	1
Deceleration	2	1	0

Table 1. MST Cost metrics for comparing qualifier attributes of acceleration and deceleration

Similarly we can produce cost metrics for rotation attributes. Of interest in this abstract comparison is whether rotations are clockwise or counterclockwise. The corresponding mobility states in this context are clockwise rotation, counterclockwise rotation, and no rotation. The latter is indicated by the lack of a rotation node over a search interval. The corresponding metrics are shown in table 2, and follow a rationale similar to the one used in the cost metrics of table 1.

Helix 1 ↓ Helix 2 →	Clockw.	No	Counter**.**
Clockwise rotation	0	1	2
No rotation	1	0	1
Counterclockwise rotat.	2	1	0

Table 2. MST Cost metrics for comparing qualifier attributes of rotation

Regarding prong information we have a similar situation, where there may be expansion, contraction, or no change in an object's outline. This last option is indicated by the lack of a prong over a search interval. The corresponding metrics are shown in table 3, and follow a rationale similar to the ones used to form the cost metrics of tables 1 and 2.

Helix 1 ↓ Helix 2 →	Expand	No	Contract
Expansion	0	1	2
No change	1	0	1
Contraction	2	1	0

Table 3. MST Cost metrics for comparing prong magnitudes

Combined, the above three metrics allow us to perform abstract comparisons of helixes. It should be mentioned again that during this stage of abstract comparisons we do not make use of information on the angular extent and magnitude of deformations, but save this information for the more detailed quantitative comparisons.

The above three cost metrics are combined in an integrated index Sim_r^c to express the similarity between a reference (H^r) and a matching candidate (H^c) helix as:

$$Sim_r^c = \frac{a_v \Sigma \cos t_{velocity} + a_r \Sigma \cos t_{rotation} + a_d \Sigma \cos t_{deformation}}{(number\_of\_nodes + number\_of\_prongs)} \quad (1)$$

where:

$cost_{velocity}$ are the cost metrics referring to table 1, aggregated over all nodes,

$cost_{rotation}$ are the cost metrics referring to table 2, aggregated over all nodes,

$cost_{deformation}$ are the cost metrics referring to table 3, aggregated over all prongs,

a_v, a_r, a_d, are the corresponding relative weights for each component, with $a_v+a_r+a_d=1$.

In general, all types of MST cost metrics receive equal weight ($a_v = a_r = a_d = 1/3$). It is possible for certain applications to put more emphasis on certain aspects than others (e.g. focusing on velocity variations more than rotations), and we can easily accommodate this by modifying the corresponding coefficients. The combined index Sim_r^c ranges between 0 and 2, with 0 corresponding to a perfect match and 2 reflecting the highest possible dissimilarity. Lower values reflect better matches to a reference helix, and this information is used to rank the matching candidates according to their similarity to a reference helix.

3.2 Quantitative Comparisons

The other type of comparison is quantitative, with specific differences computed between the values of nodes and prongs. In this case, instead of a somewhat arbitrary value of "2" assigned to a pair of dissimilar nodes, the angle of acceleration and the angle of deceleration are compared by taking the absolute value of the difference between them. Similar differences are found between angles of rotation and the magnitudes of expansion or contraction. In this type of comparison, the following equation is utilized:

$$Sim_r^c = a_n \Sigma (n_r^i - n_c^j) + a_p \Sigma (p_r^i - p_c^j) + a_q \Sigma (q_r^i - q_c^j) + \\ + a_r \Sigma (r_r^i - r_c^j) + a_a \Sigma (a_r^i - a_c^j) \quad (2)$$

where $(n_r^i\text{-}n_c^j)$ expresses the Euclidean spatiotemporal distance among a reference and a corresponding candidate node, aggregated across all nodes,

$(p_r^i\text{-}p_c^j)$ expresses the Euclidean spatiotemporal distance among a reference and a corresponding candidate prong, aggregated across all prongs,

$(q_r^i\text{-}q_c^j)$ expresses the difference in velocity gradient or rotation among corresponding nodes, aggregated across all nodes,

$(r_r^i\text{-}r_c^j)$ expresses the difference in deformation magnitude among corresponding prongs, aggregated across all prongs,

$(a_r^i\text{-}a_c^j)$ expresses the difference in deformation angle among corresponding prongs, aggregated across all prongs,

a_n, a_p, a_q, a_r, a_a are the corresponding relative weights for each component, with $a_n+a_p+a_q+a_r+a_a=1$

We normalize all quantities by dividing their actual values by their range, so that in this case, a value of 1 is assigned to the most dissimilar pairs and a value of 0 is given to pairs that are exactly the same. Once a degree of similarity has been determined, whether by abstract or quantitative methods, we can decide whether these helixes belong in a group or should remain as separate entities. Papers on similarity that address relevant issues include (Stefanidis, Agouris et al. 2002; Vlachos, Gunopulos et al. 2002; Stefanidis, Eickhorst et al. 2003).

4. GROUPING HELIXES

If the helixes in question are sufficiently similar, then it may be useful to group them together into a single entity and to express the behavior of their component objects with an "aggregate helix." The aggregate helix that is created could then be used for predictions about the future behavior of all polygons that begin in a similar way to the first few nodes and prongs of the aggregate (Figure 3). The user can select the level of similarity that must be reached in order to justify this decision, with more detailed applications needing helixes with comparison values approaching zero.

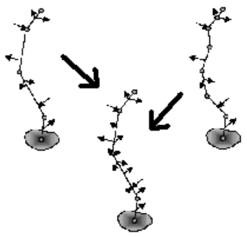

Figure 3: Aggregate helix as formed from individual helixes

When sufficiently low values are found, node and prong locations of the individual helixes can be averaged so that the new aggregate helix is a composite of the original helixes. Attributes that are associated with each node and prong can be calculated in a variety of ways, including averaging all values for each instance, looking for minimum or maximum value, or using categorical rules in order to choose the best value for a given application.

When dealing with the aggregate helix that has been constructed, there may be instances when component helixes become sufficiently different over time and should be split from each other. We can discover such instances by computing deviations of node/prong values from the aggregate average and splitting the helixes when a user specified threshold is crossed.

5. HELIX GENERATION

The spatiotemporal helix model is constructed in a four-step process, which will be detailed in this section: 1) find the center of mass for the object at each time instance, 2) detect changes in the object's outline in each cardinality quadrant, 3) construct a self-organizing map (SOM) that picks out only those nodes which are necessary to generalize the object's behavior and forms a "spine" for the helix, and 4) add information about outline changes to the spine with "prongs" that show expansion or contraction.

A 400x400 pixel grid has been utilized to create a synthetic dataset of five polygons, one in each frame. The polygons in these frames represent snapshots in the evolution of an object or phenomenon over time. Before reaching this stage, an object extraction procedure would need to be performed on our real-world data, but this is outside the scope of the current paper. For more information on our relevant activities in object extraction, the reader is referred to (Agouris, Beard et al. 2000; Agouris, Stefanidis et al. 2001; Doucette, Agouris et al. 2001).

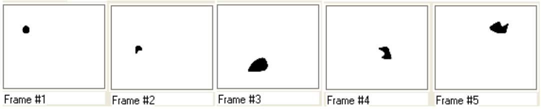

Figure 4: Five sample frames used for input in helix extraction

5.1 Center of Mass Extraction

In the first stage of helix construction, the object's center of mass is extracted and plotted on a three-dimensional grid. Each asterisk indicates the location of the object at a given time instance. In this example dataset, we assume that each of the frames used in this example was taken after a ten-minute delay. The first frame is thus linked to time t=10 and the fifth frame is linked to t=50. In this first stage, a trajectory is also constructed by linking the centers of mass for each frame (Figure 5).

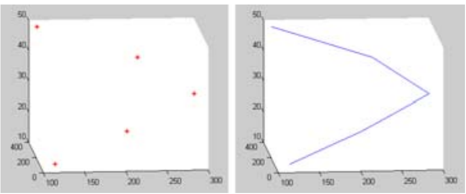

Figure 5: Object's center of mass and rough trajectory

In order to test the robustness of this method under more realistic conditions, we added random noise to our images using Matlab's "randerr" function, which introduces a user-selected number of nonzero elements into each row of a matrix. We multiplied our original frames by these new matrices in order to create new noisy images (Figure 6 left).

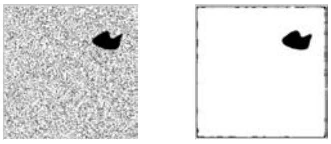

Figure 6: Frame #5 before and after noise removal

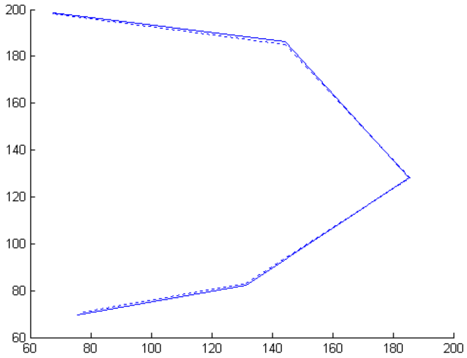

Figure 7: Object trajectories from initial (solid) and cleaned (dotted line) frames

When a 9x9 median filter was applied to the noisy image, the number of erroneous DN=0 pixels was reduced dramatically (Figure 6 right). Most of the pixels that are left are located around the edges of the frame, due to algorithm limitations. In order to determine how the few remaining noisy pixels will effect the center of mass calculations, we mapped the initial trajectory of the object's center of mass, and compared it to the center of mass after the noise removal procedure. We found that there is nearly a one-to-one correspondence between these trajectories (Figure 7). These results indicate that the procedure has been successful in removing noise.

5.2 Cardinality Changes

The second stage divides the object into four quadrants, based on the cardinal directions of north, south, east, and west (assuming an orientation where north is towards the top of the frame). This is done for each frame, and the center of mass found in the first step is used as the origin for each division (Figure 8). The object in frame n is then compared to the same object in frame n+1 in order to discover whether there has been an expansion or contraction during each time interval. For instance, the object grows significantly between frames 2 and 3, and this leads to an increase in area for all four quadrants. This change will be quantified in the final step of helix construction, which is discussed later in this section.

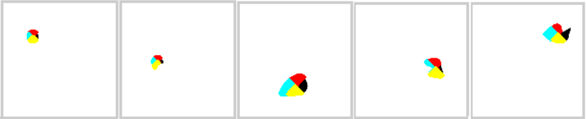

Figure 8: Object divided into cardinality quadrants

5.3 Self-Organizing Map Construction

The third stage is concerned with construction of a Self-Organizing Map (SOM) that generalizes the trajectory of the helix by picking out locations where changes such as rapid acceleration, deceleration, or rotation occur and marking them with nodes. A SOM is a neural network solution that organizes nodes into an ordered sequence through competitive learning (Kohonen 1997). In this example, the object is moving at a fairly uniform pace, so it does not experience much acceleration or deceleration. The major change is rotation, occurring most notably at frame 3, the apex of the object's trajectory.

When we ask for a generalized picture of the spine with 4 nodes we can see that the nodes for frames 1 and 2 from Figure 5 are merged into a single intermediate node to save space (Figure 9). When we ask for 3 nodes, only frame 3 retains its original node. This decreases the number of nodes used to define the polygon's location, and leads to a reduction in the amount of space needed to store this data while maintaining the most important characteristics of the object's spatiotemporal behavior. For more detailed information on our SOM work see (Kohonen 1982; Kohonen 1997; Doucette, Agouris et al. 2001).

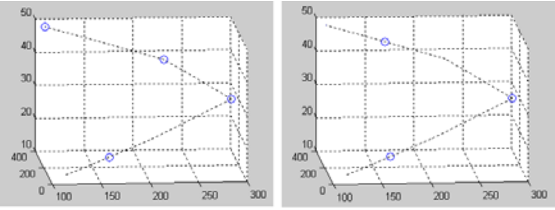

Figure 9: SOMs constructed with 4 (L) and 3 (R) nodes

5.4 Node Placement and Prong Information

The final stage in helix construction is to move each extracted node to the closest position recorded in the frames and to add prong information. For example, when four nodes are extracted in the SOM process, three of the nodes are located at the object's position in frames 3, 4, and 5. The fourth node is located between the object's positions in frames 1 and 2, but is closer to that of frame 2 (Figure 9). Thus, when constructing the helix, our algorithm places the final nodes in frames 2, 3, 4, and 5 (Figure 10 left).

When examining the SOM of 3 nodes, we end up with final node placement in frames 1, 3, and 5 (Figure 10 right). In our example, this would select frames from the original dataset, and use only them to define the placement of the polygon over time. It is more accurate than using the node placements from the third step, because it does not create interpolated positions, but uses locations that were already part of the dataset.

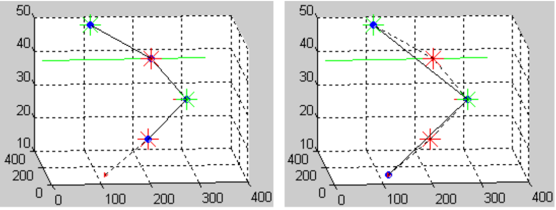

Figure 10: Complete helixes for 4(L) and 3(R) nodes with spines and prongs

In addition to selecting the most important object instances that should be recorded in our database, the fourth stage in our helix construction process also compares changes in object expansion or contraction to a user-defined threshold. In this example, the threshold has been set at 20%. The largest change that was found in our example dataset occurs between frames 3 and 4, where there is a large reduction of area in the west quadrant and a smaller reduction in the east quadrant. This is represented in our helixes by a long line emerging from the "west" side of the node at frame 4 and a shorter line on the "east" side of the same node. This indicates that the polygon has undergone the most significant change in outline between these two frames.

6. ADDITIONAL EXPERIMENTS

In addition to these basic experiments in extracting helix information, we have tested the integrity of our calculations, as well as their performance speed. For helix generation, constructed datasets of 700 frames and used differing user-defined thresholds to determine the number of nodes and prongs that define the helix. Figure 11 shows two helixes that were constructed during this phase. Both have 17 nodes, but helix "a" has more prongs than helix "b." Their respective prong thresholds are 10% and 20%.

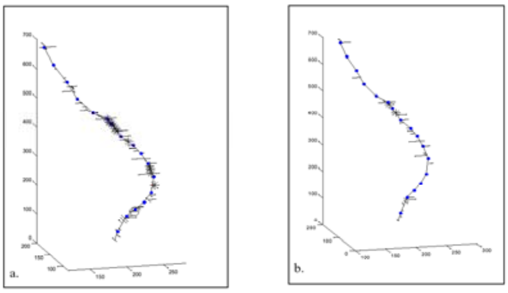

Figure 11: Helixes constructed from differing thresholds

In order to determine the usefulness of our prongs in reconstructing an object at any given time instance, we used only the image of the object at t=0 sec, and modified the initial object outline using only the expansions and contractions as indicated by the prong magnitudes and angles. We then compared these results to the actual object boundaries in frame 700. We found that with our dataset, we were able to reconstruct the object with 83% accuracy when using a prong threshold of 20% and with around 94% accuracy when using any prong threshold below 15%. There seems to be a level of prong definition beyond which no additional benefit is gained in storing the extra information. See (Stefanidis, Eickhorst et al. 2003) for a more detailed discussion of this topic.

Another type of experiment that we conducted involves the computation of similarity indices using the metrics discussed in section 3. We created a dataset of 100 helixes, comprising an average of 19 nodes and 7 prongs each, and used both the abstract and quantitative metrics to compare each helix to the larger pool of candidate helixes. We noted the time that it took to run each of these queries, and found that the abstract query averaged 2 seconds to run, while the more intensive quantitative query took 4 seconds. These are very encouraging results as many applications in the geospatial realm are large-scale efforts where computational times are of the utmost importance.

7. FUTURE DIRECTIONS

We are currently exploring various ways to visualize node and prong values with colors, various levels of shading, fuzziness, or other overlays. This information is intended to supplement the quantitative values of the helix components, to support quick decision-making though visual analysis. For instance, if one wanted to be visually alerted to nodes where accelerations occur rapidly, such nodes can be identified using red (or other bright colors). If accuracy metadata are also available, this could be expressed in the relative crispness or fuzziness of the node and prong representations (Pang, Wittenbrink et al. 1997). These future extensions of our model are expected to be useful in alerting users to potential situations where merging or splitting helixes might be helpful.

In addition to visual representations of helixes, we are also working on utilizing aggregation and splitting techniques when dealing with text and other data formats that may be associated with the helixes. These formats can complement and enhance the helixes. As an example, we might have metadata about accuracy. The level of metadata that deals with specific horizontal and vertical accuracies of angles is quite detailed. The FGDC lists a set of standard metadata that is set up in a hierarchical fashion. By exploiting this construction we can gradually zoom in to the level of detail that is required, both within the data itself and within the corresponding metadata (Eickhorst 2002; Eickhorst and Agouris 2002). Thus, the user would not need to actually see metadata on accuracy or fuzzy visualizations until a very detailed view of the helix itself is desired. Integrating this capability into our helix model is a future goal of our research.

8. CONCLUSIONS

Spatiotemporal helixes represent a new and promising theory for modelling and analyzing change in geospatial applications and other large-scale data-intensive projects in various fields. In this paper, we have presented our work on the theories of helixes and similarity metrics, and have given examples of how these theories can be implemented. We have presented examples of the experiments being conducted to test the robustness of our methods. Finally, we discussed some new directions that we are currently pursuing. These extensions will result in the development of a comprehensive model that will support users querying on object behaviour over time, using an easily understandable interface.

REFERENCES

Agouris, P., K. Beard, et al. (2000). Capturing and Modeling Geographic Object Change: A SpatioTemporal Gazetteer Framework. *Photogrammetric Engineering & Remote Sensing*, 66(10), pp. 1241-1250.

Agouris, P. and A. Stefanidis (2003). Efficient Summarization of Spatiotemporal Events. *Communications of the ACM*, 46(1), pp. 65-66.

Agouris, P., A. Stefanidis, et al. (2001). Differential Snakes for Change Detection in Road Segments. *Photogrammetric Engineering & Remote Sensing*, 67(12), pp. 1391-1399.

Doucette, P., P. Agouris, et al. (2001). Self-Organized Clustering for Road Extraction in Classified Imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(5-6), pp. 347-358.

Eickhorst, K. (2002). Hierarchical Structures for Video Query Systems. *DG.O 2002 Proceedings, National Conference on Digital Government Research*, Los Angeles, pp. 475-479.

Eickhorst, K. and P. Agouris (2002). On the Use of Hierarchies and Feedback for Intelligent Video Query Systems. *Proceedings of ISPRS 2002 Symposium of Commission IV*, Ottawa, pp. 5.

Hornsby, K. and M. Egenhofer (2002). Modeling Moving Objects over Multiple Granularities. *Annals of Mathematics and Artificial Intelligence*, 36(1-2), pp. 177-194.

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

Kohonen, T. (1997). *Self-Organizing Maps*. Springer-Verlag

Pang, A., C. Wittenbrink, et al. (1997). Approaches to Uncertainty Visualization. *The Visual Computer*, 13(8), pp. 370-390.

Plaisant, C., B. Milash, et al. (1996). LifeLines: Visualizing personal histories. *CHI '96*, New York, ACM, pp. 221-227.

Pope, A., R. Kumar, et al. (1998). Video Abstraction: Summarizing Video Content for Retrieval and Visualization. *Proc. 32nd Asilomar Conference of Signals, Systems & Computers*, pp. 915-919.

Stefanidis, A., P. Agouris, et al. (2002). Scale and Orientation-Invariant Scene Similarity Metrics for Image Queries. *International Journal of Geographical Information Science*, 16(8), pp. 749-772.

Stefanidis, A., P. Agouris, et al. (2002). SpatioTemporal Helixes for Event Modeling. *National Conference on Digital Government Research*, Los Angeles, pp. 219-224.

Stefanidis, A., K. Eickhorst, et al. (2003). Modeling and Comparing Change using Spatiotemporal Helixes. *ACM-GIS 2003*, New Orleans, LA, ACM Press, pp. 86-93.

Vlachos, M., D. Gunopulos, et al. (2002). Robust Similarity Measures for Mobile Object Trajectories. *DEXA 2002, 5th International Workshop on Mobility in Databases and Distributed Systems*, Aix-en-Provence, France, pp. 721-728.

Zhou, J. Y., E. P. Ong, et al. (2000). Video Object Segmentation and Tracking for Content-Based Video Coding. *IEEE International Conference on Multimedia and Expo (III)*, pp. 1555-1558.

ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation through grants DG-9983432 and ITR-0121269 and by the National Geospatial-Intelligence Agency (NGA) through NMA 401-02-1-2008. Kristin Eickhorst’s work has also been funded in part by NASA through a Maine Space Grant Fellowship, and the generous support of Goddard Space Flight Center.