TRACKING DEFORMABLE OBJECTS IN GEOSPATIAL APPLICATIONS

Sotirios Gyftakis, Peggy Agouris, Anthony Stefanidis

Department of Spatial Information Science and Engineering
University of Maine, Boardman Hall #348, Orono, ME 04469-5711, USA
{sotiris, peggy, tony}@spatial.maine.edu

ABSTRACT

In this paper we describe a novel approach for change detection of moving deformable objects. We assume that each object is represented as a closed polygon (convex or concave). First, we use differential snakes to track the object outline between two different frames. Next, we examine the geometric properties (area moments, and principal axes) of these two polygons and we estimate their changes (translation, rotation, area expanded or contracted). Additionally, using results from invariant moments theory we compute a similarity index among different instances of an object's history. This information is used to update geospatial databases and to support complex analysis. Experimental results on real imagery illustrate the capabilities of our approach.

1. INTRODUCTION

Change detection is a topic of great importance for modern geospatial information systems (GIS). Rapidly changing environments, and the availability of consistently increasing amounts of diverse, multiresolutional datasets bring forward the need for frequent revisions of modern geographic information systems (GIS).

In this paper we investigate the problem of tracking the movement, rotation, and/or deformation of a geospatial object as it is captured in a series of aerial images. Such series may comprise either frames of a digital video feed (e.g. captured by a sensor on-board an unmanned aerial vehicle that roams over an area of interest), or a sequence of different static images depicting the same area at different instances. Examples of events that may be monitored in this manner range from the rapid evolution of a fire and the progression of an oil spill, to the recurrent flooding of a lake. Our objective is to capture the motion, rotation, and deformation of this object. The results of this tracking can be used to update a GIS database, and are typically the subject of further geospatial analysis to identify notable trends and critical instances in these deformations (e.g. after they are used to populate spatio-temporal helix models [1]).

In the research area of moving objects tracking active contour models (or snakes) have been used extensively in areas such as robotic control [2] and medical imaging [3], because of their effectiveness and speed. At the core of our approach is the extension of the differential snakes method as they were introduced in [4]. The authors in [4] make use of prior information for an object (an older record of the shape of the object) and accompanying accuracy estimates to detect automatically changes in the object's outline, and update the corresponding GIS database (shape and corresponding accuracy). In this paper we enhance this model to identify object translation, rotation, and/or deformation from local outline variations, and to support the tracking of closed-outline areal objects.

In the next section we describe the tracking of deformable objects using differential snakes. A similarity index is introduced in section 3. In section 4 we describe our experiment and the results. We present our conclusions in section 5.

2. DIFFERENTIAL SNAKES FOR TRACKING OBJECTS WITH CLOSED CURVES

In this section we present our method for spatiotemporal change detection of moving closed objects using differential snakes. We represent each object at a specific time frame as a closed polygon (convex or concave). We assume that we investigate the changes of the same object between two different time frames. If we know the (approximate) location of the object on the first frame, we can use active contours (snakes) to extract its (polygonal) contour at this instance. This initial contour can then be used as an approximation to automatically extract the next object instance. This differential snakes technique has been demonstrated to function very well in cases where the object monitoring rate is high, thus ensuring relatively small variations between successive instances [4]. The result of differential snakes is the automated identification of the closed polygons that represent the object's outlines at different instances. Here we present a methodology that uses this information to estimate the movement, rotation, and deformation of the complete object. Our approach proceeds in the following manner.

First, we determine object translation. This is a relatively straight-forward issue, and we make use of the geometric properties (area A, and perimeter P) of the two polygons. We compute their respective geometric centers

(centers of area: x_c, y_c), and compare them to decide whether the areas and perimeters have changed as a result of uniform and/or radial expansion. In the case of uniform expansion the ratio of areas must be equal to the square of the ratio of perimeters (eq. 1).

$$\frac{A}{A'} = \left(\frac{P}{P'}\right)^2 \tag{1}$$

From the difference of the positions of the two geometric centers we decide if they have been translated and we compute their relative translation (eq. 2).

$$\begin{pmatrix} tr_x \\ tr_y \end{pmatrix} = \begin{pmatrix} x_c^2 \\ y_c^2 \end{pmatrix} - \begin{pmatrix} x_c^1 \\ y_c^1 \end{pmatrix} \tag{2}$$

Next, we estimate relative rotation. In order to do so we compute the centroidal principal moments of inertia for each polygonal object. These identify the principal axes (one for the min, one for the max moment) that by definition pass through the center of the area. We compute the angles of the principal axes of for each object. The difference of the angles of the principal axes will give us the relative rotation of the objects (Fig. 1).

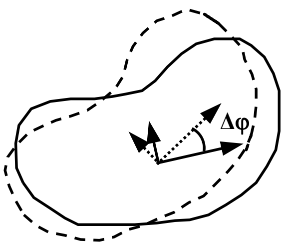

Figure 1: Use of Principal Moments for Computing of Rotation Angle (Δφ)

We compute the centroidal moments from the second moments:

$$I_{uu}=I_{xx}-A\cdot y_c^2, \qquad I_{vv}=I_{yy}-A\cdot x_c^2, \qquad I_{uv}=I_{xy}-A\cdot x_c\cdot y_c \tag{3}$$

The principal moments are the eigenvalues of the tensor $\underline{\underline{I}}$ and the principal axes are its eigenvectors.

$$\underline{\underline{I}} = \begin{bmatrix} I_{uu} & I_{uv} \\ I_{uv} & I_{vv} \end{bmatrix} \tag{4}$$

The eigenvectors e^{1p}, and e^{2p} satisfy the set of linear equations:

$$\begin{pmatrix} I_{uu}-\lambda & I_{uv} \\ I_{uv} & I_{vv}-\lambda \end{pmatrix} \cdot \begin{pmatrix} e_x^p \\ e_y^p \end{pmatrix} = 0 \tag{5}$$

Nontrivial solutions for the eigenvector exist only if the determinant is equal to zero:

$$\begin{vmatrix} I_{uu}-\lambda & I_{uv} \\ I_{uv} & I_{vv}-\lambda \end{vmatrix} = 0 \tag{6}$$

From the characteristic equation (binomial) we find the principal moments (I_1, I_2). We substitute these values back into the set of the linear equations and we compute the eigenvectors (e^{1p}, e^{2p}). We choose unit eigenvectors, since we are interested only in the direction of the principal axes.

For the relative orientation angle we compute the direction (φ) of the first eigenvector (for each frame) and we find the difference (Δφ) between them.

$$\varphi = \tan^{-1}\left(\frac{e_y^{1p}}{e_x^{1p}}\right) \quad \text{and} \quad \Delta\varphi = \varphi_2 - \varphi_1 \tag{7}$$

For radial deformation of the two objects, we use polygon clipping techniques. We assume that the objects are described as closed polygons, they do not contain holes, they are non-self-intersecting (convex or concave) and these polygons are clockwise oriented. The source code (in C) for polygon clipping was based on the thesis of K. Schutte [5] and it is available in public. It was compiled into a Matlab function (MEX). The worst case complexity of this polygon clipping algorithm is O(nm), where n is the number of vertices in polygon A and m is the number of vertices in polygon B.

The basic algorithm for polygon clipping has the following steps. Calculate the intersections between two input polygons; label edges as inside, outside, or shared; find the minimal polygons which are created by intersection; and classify all minimal polygons into the output sets A∩B, A/B, and A\B.

3. SIMILARITY ASSESSMENT USING MOMENT INVARIANTS

In order to increase the consistency of our change detection algorithm we evaluate the similarity between the two objects extracted from the two frames. There have been developed few shape description techniques for matching of geometric shapes, e.g. Fourier descriptors, boundary chain coding, moment invariants. In our research we have utilized moment invariants [6] because they are computationally efficient.

Moment invariants are second and third order moments of area that are invariant under general affine transformation [7]. They have been used for recognition of topographic objects [8] and for template matching [9].

For the calculation of moments of closed region using discrete boundary integrals we have used the results of [10]. The general boundary integral (eq. 8 is the cross moment) is derived analytically using Gauss' divergence theorem and its finite discretization is given in eq. 9.

$$\mathrm{M}_{pq} = \iint x^p y^q f(x,y)dxd \tag{8}$$

$$\int_D x^n y^m dD = \frac{-1}{m+1} \cdot \sum_{j=1}^{N} \left\{ \frac{\left(x_{j+1}-x_j\right)}{L_j^{n+m+2}} \cdot \sum_{k=0}^{n} \sum_{l=0}^{m+1} \binom{n}{k} \cdot \binom{m+1}{l} \cdot a_j^k \cdot c_j^l \cdot b_j^{n-k} \cdot d_j^{m+1-l} \cdot \frac{(0.5\cdot L_j)^{k+l+1} - (-0.5\cdot L_j)^{k+l+1}}{k+l+1} \right\} \tag{9}$$

where

$$a_j = (x_{j+1} - x_j)\,,\ b_j = 0.5 \cdot L_j \cdot (x_{j+1} + x_j)\,,\ d_j = 0.5 \cdot L_j \cdot (y_{j+1} + y_j)$$

$$c_j = (y_{j+1} - y_j)\,, \qquad L_j = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2} \tag{10}$$

The central moments are defined as:

$$\mu_{pq} = \sum_{(X,Y)\in C} (x - \overline{x})^p (y - \overline{y})^q \tag{11}$$

where the coordinates of the center of gravity are:

$$\overline{x} = \frac{M_{10}}{M_{00}} \quad \text{and} \quad \overline{y} = \frac{M_{01}}{M_{00}} \tag{12}$$

As it is proven in [9] moment invariants have the form of a polynomial of the central moments and the simplest of these are:

$$I_1 = (\mu_{20} \cdot \mu_{02} - \mu_{11}^2) / \mu_{00}^4 \tag{13}$$

$$I_2 = (\mu_{30}^2 \cdot \mu_{03}^2 - 6 \cdot \mu_{30} \cdot \mu_{21} \cdot \mu_{12} \cdot \mu_{03} + 4 \cdot \mu_{30} \cdot \mu_{12}^3 + 4 \cdot \mu_{21}^3 \cdot \mu_{03} - 3 \cdot \mu_{21}^2 \cdot \mu_{12}^2) / \mu_{00}^{10} \tag{14}$$

These polynomials of moments are derived by means of algebraic invariants and they are invariant under the general affine transformation.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of our technique we performed a series of experiments, tracking changes in the shape of an object. In the experiments reported in this paper we use four aerial images of a lake that have been synthetically altered to show the deformations of this lake.

First, we extract (using snakes) the shape of the lake from the first frame and next using differential snakes we extract the shape of the lake in the next three frames. Next, for change detection between any two frames we calculate the areas, perimeters and centers of the polygonal objects. We examine for possible uniform expansion and we calculate the angles of the axes of the centroidal principal moments for each polygon. The difference of these angles gives the angle of rotation. Then we translate and rotate the second object so that their geometric centers and first eigenvectors coincide. Finally, we apply polygon clipping to calculate the areas that have been expanded or contracted. We can set an area threshold if we want to ignore small polygon changes.

For the similarity index we use the invariant moments I_1 and I_2 from equations (13) and (14). For two objects A and B, we define the similarity index (sI) as the geometrical mean of the ratios of their invariant moments (eq. 15). For two perfectly similar objects sI is one, and its value deviates as dissimilarity increases.

$$sI_1 = \frac{I_1^B}{I_1^A}\,, \qquad sI_2 = \frac{I_2^B}{I_2^A}\,, \qquad sI = \sqrt{sI_1 \cdot sI_2} \tag{15}$$

We have integrated the process for spatiotemporal change detection in a GUI (written in Matlab) and shown in Fig. 2. First, the user selects the sequence of images that contains the frames we want to analyze further. Next, the user can view the movie frame by frame and select any two of these frames (initial, final) for change detection analysis. After selecting the frames, the user can run the change detection algorithm. The results of change detection are shown in two regions (text, graphics) of the GUI. In the text region, we display the similarity index, the translation (in pixels on x, y axes), rotation (in degrees) of the second object with respect to the first object, as well as the number of expanded and contracted polygons and their respective areas (in pixels). In the graphics region of the GUI we display the two extracted objects (with their centers and first eigenvectors coinciding), while their expanded and contracted areas are indicated with different colors. Finally, the user can save the results (similarity, polygon coordinates, areas, translation, rotation) for further processing. In our computer environment (3.2 GHz processor, 1GB RAM) the change detection algorithm runs in average time of 2.7 sec. The imagery used is 500X500 pixels and the snake nodes are 100 for each object.

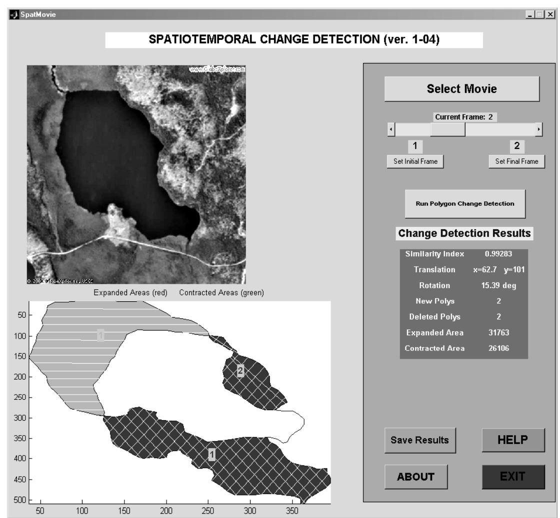

Figure 2: GUI for Spatiotemporal Tracking of Object Deformation

To demonstrate the robustness of our similarity assessment technique with respect to severe variations (e.g. occlusions, catastrophic changes) we performed the following experiment. We truncated repeatedly the area of the lake and computed the similarity indices sI_1 and sI_2 each time. We have to note here that in order for the comparison of these two objects to be meaningful, we have to use the same reference point in them. This is accomplished by transferring the center of the reference object to the one that we compare with it.

In Fig. 3 we show the lake and the boundaries for lake coverage 90%, 80%, 70% and 50% respectively. In Fig. 4

we display the results in two graphs. The x axis is the area coverage (in %) relative to the original, and the y axis is sI_1 (left) and sI_2 (right). It should be noted here that the sI_2 (y) axis is logarithmic to account for its high values.

The two graphs demonstrate that sI_2 is more sensitive to object variations than sI_1. The sI_1 index displays relative stability even for variations as high as 30% of the object size, while sI_2 displays a nearly exponential deterioration with area changes.

5. CONCLUDING REMARKS

In this paper we presented a methodology for spatiotemporal tracking of the deformations of an object. Using differential snakes, a variation of active contours, we are able to detect the boundary of the object in two frames. From the geometric properties of the object, we showed how we could track the translation, rotation and radial deformation of it.

Additionally, we showed that the area moment invariants are very useful as a similarity metric. In a future expansion of this methodology of change detection and tracking in a multi-object environment this similarity metric will be able to identify similar objects for comparison.

6. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation through grants ITR-0121269, and the National Geospatial-Information Agency through NMA 401-02-1-2008.

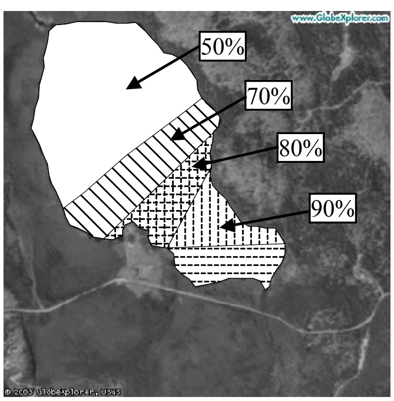

Figure 3: Experiment for Similarity Index Robustness (lake coverage in %)

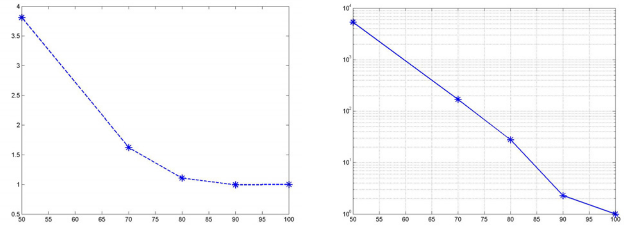

Figure 4: Results of Similarity Index Robustness Experiment (area of lake vs. similarity index, sI_1 left, sI_2 right)

7. REFERENCES

[1] P. Agouris and A. Stefanidis, "Efficient Summarization of SpatioTemporal Events," *Communications of the ACM*, 46(1), pp. 65-66.

[2] J. Denzler and D.W.R. Paulus, "Active Motion Detection and Object Tracking," *International Conference on Image Processing*, Austin, TX, pp. 635-639, 1994.

[3] D. Rueckert and P. Burger, "Geometrically Deformable Templates for Shape-Based Segmentation and Tracking in Cardiac MR Images," *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 83-98, Springer-Verlag, 1997.

[4] P. Agouris, S. Gyftakis, and A. Stefanidis, "Quality-Aware Deformable Models for Change Detection," *International Conference on Image Processing 2001*, Thessaloniki, Greece, pp. 805-808, October 2001.

[5] K. Schutte, *"Knowledge Based Recognition of Man-Made Objects,"* Ph.D. diss., University of Twente, 1994.

[6] J. Mundy and A. Zisserman, *"Geometric Invariance in Computer Vision,"* MIT Press, Cambridge, MA, 1992.

[7] T. Suk and J. Flusser, "Features Invariants Simultaneously to Convolution and Affine Transformation," *Computer Analysis of Images and Patterns*, Lecture Notes in Computer Science (2124), W. Skarbek ed., Springer, Berlin, pp. 183-190, 2001.

[8] L. Keyes and A. Winstanley, "Using Moment Invariants for Classifying Shapes on Large Scale Maps," *Computers, Environment and Urban Systems*, vol. 25:1, pp. 119-130, 2001.

[9] J. Flusser and T. Suk, "Pattern Recognition by Affine Moment Invariants," *Pattern Recognition*, vol. 26:1, pp. 167-174, 1993.

[10] W. Yeih, J.R. Chang, and J.T. Chen, "A Boundary Formulation for Calculating Moments of an Arbitrary Closed Planar Region," *Engineering Analysis with Boundary Elements*, vol. 23:7, pp. 611-617, 1999.