

# Statistical Shape Learning for 3D Tracking

Romeil Sandhu, Shawn Lankton, Samuel Dambreville, Allen Tannenbaum  
School of Electrical and Computer Engineering,  
Georgia Institute of Technology,  
Atlanta, GA, 30332

**Abstract**—In this note, we consider the use of 3D models for visual tracking in controlled active vision. The models are used for a joint 2D segmentation/3D pose estimation procedure in which we automatically couple the two processes under one energy functional. Further, employing principal component analysis from statistical learning, can train our tracker on a catalog of 3D shapes, giving *a priori* shape information. The segmentation itself is information-based. The allows us to track in uncertain adversarial environments. Our methodology is demonstrated on some real sequences which illustrate its robustness on challenging scenarios.

## I. INTRODUCTION

The use of 3D models can be a major advantage for various tasks in visual tracking in controlled active vision. Specifically, if one has a 3D model of a target, the model can be used to guide the segmentation of the object as well as estimate its location and pose in the world as shown recently in [3]. The ability to measure location and pose of a target can drastically improve the usefulness of tracking results over those obtained without knowledge of the target's 3D characteristics.

The ability to acquire 3D models has become very accessible. Models can be obtained off-line using methods such as high accuracy laser scanning and multi-view stereo reconstruction. Additionally, models can be learned on-line by registering multiple views of an object acquired by 3D imaging modalities such as light detection and ranging (LADAR) and stereoscopic vision [18].

In this work, using statistical learning techniques, we show how to train a tracker based on a catalog of 3D shapes, which enables one to employ a shape prior in 3D tracking tasks. Moreover, this allows us to detect and track deformable objects even in cluttered and noisy environments. The image segmentation part of this method is crucial, and so we briefly outline here what is involved.

Image segmentation consists of partitioning a scene into an “object” and a “background.” We propose to

employ the geometric active contour (GAC) framework [15], [10], [14], whereby a curve is evolved continuously until it satisfies a stopping criterion that coincides with the object's boundaries. The model used will be based on statistical information theory, in particular, the Bhattacharyya coefficient from information theory [7]. Although this improves segmentation results, one still may encounter problems with tracking in cluttered adversarial scenarios. Thus, we will use a shape prior to restrict the evolution of the active contour using principal component analysis (PCA) [6]. To this end, we derive a novel 3D shape prior from a dictionary of 3D shapes to do 2D image segmentation rather than to derive a 2D shape prior from a collection of 2D images. As a result, we are then able to reduce computational complexity in statistical shape learning approaches through a compact shape representation.

## II. STATISTICAL LEARNING

The statistical learning procedure we will employ in our tracking model is Principal Component Analysis (PCA). This is a popular technique for shape analysis and dimensional reduction. The overall method gives an implementable solution to the following problem: Given a number of observations about a complex system describing different states, estimate the number of independent parameters of the system (i.e., its dimension) and the way these are related. Thus PCA is a linear version of *manifold learning*. All manifold learning methods are based on the assumption that the data (usually a point cloud in some  $n$  dimensional space) lie on or are close to a submanifold  $M \subset \mathbf{R}^n$ .

There have been a number of different approaches proposed for this problem area including PCA, local linear embeddings, Laplacian eigenmaps, and diffusion maps to name a few; see [1], [5], [8], [16], [13] and references therein. A number of the most important methods may be classified as spectral, in the sense a symmetric matrix is derived from the point cloud data and the solution to a given optimization problem

can be obtained from the eigenvectors of this matrix. The geometric optimization problems that lead to a spectral technique are usually of a least squares type, e.g., finding the  $k$ -dimensional subspace that approximates the point cloud best in a least squares sense, or computing the embedding of the point cloud in a  $k$ -dimensional space that preserves the distances between the points optimally in a least squares sense. The former method is principal component analysis (PCA): every data point gets replaced by its projection onto the best approximating  $k$ -dimensional subspace which we will employ in the present work.

Specifically, here we will follow the work of [6], [17]. Accordingly, we let  $\varphi_i$  represent the signed distance function corresponding to a 3D surface  $X_i$ . The average shape  $\bar{\varphi}$  can then be computed from the  $n$  surfaces as  $\bar{\varphi} = (\frac{1}{n}) \sum \psi_i$ . From this, we can exploit the variability in the training data through PCA by first creating a mean-offset map  $\tau = \{\tilde{\varphi}_1, \tilde{\varphi}_2, \dots, \tilde{\varphi}_n\}$  where  $\tilde{\varphi}_i = \varphi_i - \bar{\varphi}$ . Each map  $\tilde{\varphi}_i$  is then reorganized into a  $N \times 1$  column vector with  $N$  being the number of elements within  $\tilde{\varphi}_i$ . The resulting column stacking transformation of  $\tau$  yields an  $N \times n$  matrix  $M$ . Using Singular Value Decomposition (SVD), the covariance matrix  $\frac{1}{n}MM^T$  is decomposed as:

$$U\Sigma U^T = \frac{1}{n}MM^T \quad (1)$$

where  $U = \{\psi_1, \psi_2, \dots, \psi_n\}$  is a matrix whose column vectors represent the set of orthogonal modes of shape variation and  $\Sigma$  is a diagonal matrix of the corresponding singular values. Rearranging the column vectors back into the structure of  $\varphi_i$ , we can then estimate a novel 3D shape  $\hat{\varphi} = \bar{\varphi} + \sum_{i=0}^k w_i \psi_i$ , where  $w_i$  is the shape weight and  $k$  is the number of principal modes used (see [17] for details). It is important to note here that we are concerned only with the zero level surface of the derived shape. This detail will be essential in solving for the shape parameters.

### III. INFORMATION-THEORETIC APPROACH TO SEGMENTATION

The next ingredient in our statistical learning tracking scheme is an information-theoretic approach to segmentation. We follow here the approach in [7] to which we refer the interested reader for all of the details. The overall method is based on active contours implemented via level set techniques [10], [15], [14], [2]. In our case, the evolution is driven by the gradient flow derived from an energy functional that is based

on the Bhattacharyya distance. Because of the non-local nature of the flow, it is very useful for target tracking and can easily be combined with statistical learning as described above. In particular, given the values of a photometric variable, which is to be used for classifying the image pixels, the active contours are designed to converge to the shape that results in *maximal* discrepancy between the distributions of the photometric variable inside and outside of the contours. This discrepancy is measured by means of the Bhattacharyya distance that proves to be an extremely useful tool for solving the problem at hand [7].

#### A. Bhattacharyya Flow

For simplicity, we consider the case of two classes (i.e., the problem of segmenting an object of interest from the background). This may be extended to multi-object scenarios [7].

In the two class case, the segmentation problem is reduced to the problem of partitioning the domain of definition  $\Omega \subset \mathbb{R}^2$  of an image  $I(z)$  (with  $z \in \Omega$ ) into two mutually exclusive and complementary subsets  $\Omega_-$  and  $\Omega_+$ . These subsets can be represented by their respective characteristic functions  $\chi_-$  and  $\chi_+$ , which can in turn be defined by means of a level set function  $\Psi(z) : \Omega \rightarrow \mathbb{R}$  as  $\chi_-(z) := H(-\Psi(z))$ ,  $\chi_+(z) := H(\Psi(z))$  with  $z \in \Omega$ , where  $H$  denotes the Heaviside function. Given a level set function  $\Psi(z)$ , its zero level set  $\{x \mid \Psi(z) = 0, z \in \Omega\}$  is used to implicitly represent a curve as in [10], [15]. We associate the subset  $\Omega_-$  with the support of the object of interest, while  $\Omega_+$  is associated with the support of corresponding background. In this case, the objective of active contour based image segmentation is given an initialization  $\Psi_0(z)$ , construct a *convergent* sequence of level set functions  $\{\Psi_t(z)\}_{t>0}$  (with  $\Psi_t(z)|_{t=0} = \Psi_0(z)$ ) such that the zero level set of  $\Psi_T(z)$  coincides with the boundary of the object of interest for some  $T > 0$ .

We construct the sequence of level set functions via a gradient flow that minimizes a certain cost functional which we will now specify. First for the level set function  $\Psi(z)$ , the following two quantities are computed:

$$P_-(\mathbf{x} \mid \Psi(z)) = \frac{\int_{\Omega} K_-(\mathbf{x} - I(z)) H(-\Psi(z)) dz}{\int_{\Omega} H(-\Psi(z)) dz}, \quad (2)$$

and

$$P_+(\mathbf{x} \mid \Psi(z)) = \frac{\int_{\Omega} K_+(\mathbf{x} - I(z)) H(\Psi(z)) dz}{\int_{\Omega} H(\Psi(z)) dz}, \quad (3)$$

where  $\mathbf{x} \in \mathbb{R}^N$ , and  $K_-(\mathbf{x})$  and  $K_+(\mathbf{x})$  are two scalar-valued *kernels* having compact or effectively compact supports, normalized to have unit integrals. Thus the functions the functions  $P_-(\mathbf{x} | \Psi(z))$  and  $P_+(\mathbf{x} | \Psi(z))$  given by (2) and (3) are *kernel-based estimates* of the probability density functions pdf of the image features observed over the sub-domains  $\Omega_-$  and  $\Omega_+$ .

The key idea underpinning the segmentation approach of [7] is that for a properly selected subset of image features, the ‘‘overlap’’ between the informational contents of the object and of the background has to be minimal. In other words, if one thinks of the active contour as a *discriminator* that separates the image pixels into two subsets, then the optimal contour should minimize the *mutual information* between these subsets. Note that for the case at hand, minimizing the mutual information is equivalent to maximizing the Kullback-Leibler divergence between the pdf’s associated with the ‘‘inside’’ and ‘‘outside’’ subsets of pixels. However, because of computational efficiency instead of the divergence, we propose to maximize the Bhattacharyya distance between the pdf’s. (The *Bhattacharyya distance* is defined to be  $-\log$  of the integral given in (4) below which defines the *Bhattacharyya coefficient*.) Specifically, the optimal active contour  $\Psi^*(z)$  is defined as:  $\Psi^*(z) = \arg \inf_{\Psi(z)} \{\tilde{B}(\Psi(z))\}$ , where

$$\tilde{B}(\Psi(z)) = \int_{\mathbf{x} \in \mathbb{R}^N} \sqrt{P_-(\mathbf{x} | \Psi(z)) P_+(\mathbf{x} | \Psi(z))} d\mathbf{x}, \quad (4)$$

with  $P_-(\mathbf{x} | \Psi(z))$  and  $P_+(\mathbf{x} | \Psi(z))$  being given by the equations (2) and (3), respectively.

*Gradient Flow:* In order to derive a scheme for minimizing (4), we need to compute its first variation. Accordingly, the first variation of  $\tilde{B}(\Psi(z))$  (with respect to  $\Psi(z)$ ) is given by:

$$\frac{\delta \tilde{B}(\Psi(z))}{\delta \Psi(z)} = \quad (5)$$

$$\frac{1}{2} \int_{\mathbf{x} \in \mathbb{R}^N} \frac{\partial P_-(\mathbf{x} | \Psi(z))}{\partial \Psi(z)} \sqrt{\frac{P_+(\mathbf{x} | \Psi(z))}{P_-(\mathbf{x} | \Psi(z))}} + \quad (6)$$

$$\quad (7)$$

$$\frac{\partial P_+(\mathbf{x} | \Psi(z))}{\partial \Psi(z)} \sqrt{\frac{P_-(\mathbf{x} | \Psi(z))}{P_+(\mathbf{x} | \Psi(z))}} d\mathbf{x}. \quad (8)$$

Differentiating (2) and (3) with respect to  $\Psi(z)$ , one

obtains:

$$\delta(\Psi(z)) \left( \frac{\frac{\partial P_-(\mathbf{x} | \Psi(z))}{\partial \Psi(z)}}{A_-} - \frac{P_-(\mathbf{x} | \Psi(z)) - K_-(\mathbf{x} - I(z))}{A_-} \right), \quad (9)$$

and

$$\delta(\Psi(z)) \left( \frac{\frac{\partial P_+(\mathbf{x} | \Psi(z))}{\partial \Psi(z)}}{A_+} - \frac{K_+(\mathbf{x} - I(z)) - P_+(\mathbf{x} | \Psi(z))}{A_+} \right), \quad (10)$$

where  $\delta(\cdot)$  is the delta function, and  $A_-$  and  $A_+$  are the areas of  $\Omega_-$  and  $\Omega_+$  given by  $\int_{\Omega} \chi_-(z) dz$  and  $\int_{\Omega} \chi_+(z) dz$ , respectively.

By substituting (9) and (10) in (5) and combining the corresponding terms, one can arrive at:

$$\frac{\delta \tilde{B}(\Psi(z))}{\delta \Psi(z)} = \delta(\Psi(z)) V(z), \quad (11)$$

where

$$V(z) = \frac{1}{2} \tilde{B}(\Psi(z)) (A_-^{-1} - A_+^{-1}) + \quad (12)$$

$$\frac{1}{2} \int_{\mathbf{x} \in \mathbb{R}^N} K_+(\mathbf{x} - I(z)) \frac{1}{A_+} \sqrt{\frac{P_-(\mathbf{x} | \Psi(z))}{P_+(\mathbf{x} | \Psi(z))}} d\mathbf{x} -$$

$$\frac{1}{2} \int_{\mathbf{x} \in \mathbb{R}^N} K_-(\mathbf{x} - I(z)) \frac{1}{A_-} \sqrt{\frac{P_+(\mathbf{x} | \Psi(z))}{P_-(\mathbf{x} | \Psi(z))}} d\mathbf{x}.$$

Assuming the same kernel  $K(\mathbf{x})$  is used for computing the last two terms in (12), i.e.  $K(\mathbf{x}) = K_-(\mathbf{x}) = K_+(\mathbf{x})$ , the latter can be further simplified to the following form:

$$V(z) = \frac{1}{2} \tilde{B}(\Psi(z)) (A_-^{-1} - A_+^{-1}) + \quad (13)$$

$$\frac{1}{2} \int_{\mathbf{x} \in \mathbb{R}^N} K(\mathbf{x} - I(z)) L(\mathbf{x} | \Psi(z)) d\mathbf{x},$$

where

$$L(\mathbf{x} | \Psi(z)) = \frac{1}{A_+} \sqrt{\frac{P_-(\mathbf{x} | \Psi(z))}{P_+(\mathbf{x} | \Psi(z))}} - \quad (14)$$

$$\frac{1}{A_-} \sqrt{\frac{P_+(\mathbf{x} | \Psi(z))}{P_-(\mathbf{x} | \Psi(z))}}.$$

Introducing an artificial *time* parameter  $t$ , the gradient flow of  $\Psi(z)$  that minimizes (4) is given by:

$$\Psi_t(z) = -\frac{\delta \tilde{B}(\Psi(z))}{\delta \Psi(z)} = -\delta(\Psi(z)) V(z), \quad (15)$$

where the subscript  $t$  denotes the corresponding partial derivative, and  $V(z)$  is defined as given by either (12) or (13).

From the viewpoint of statistical estimation, the cost function (4) can be thought of as accounting for the fidelity of estimation of the optimal level set function to the features of  $I(z)$ .

#### IV. TRACKING

We will now see how to apply the above framework to 3D tracking. We begin by assuming that we have a 3D smooth surface  $S \subset \mathbb{R}^3$ . We denote by  $\mathbf{X} = [X, Y, Z]^T$  the spatial coordinates that are measured with respect to the imaging camera's referential. The (outward) unit normal to  $S$  at each point  $\mathbf{X} \in S$  is referred to as  $\mathbf{N} = [N_1, N_2, N_3]^T$ . Moreover, we assume a pinhole camera realization  $\pi : \mathbb{R}^3 \mapsto \Omega; \mathbf{X} \mapsto \mathbf{x}$ , where  $\mathbf{x} = [x, y]^T = [X/Z, Y/Z]^T$  and  $\Omega \in \mathbb{R}^2$  denotes the domain of the image  $I$  with the corresponding area element  $d\Omega$ . From this, we define  $R = \pi(S)$  to be the region onto which  $S$  is projected. Similarly, we can form the complementary region and boundary or "silhouette" curve as  $R^c = \Omega \setminus R$  and  $\hat{c} = \partial R$ , respectively. Alternatively stated, if we define the "occluding" curve  $C$  to be the intersection of the visible and non-visible region of  $S$ , then the image curve can be reinterpreted as  $\hat{c} = \pi(C)$ .

Next let  $\mathbf{X}_0$  and  $S_0 \in \mathbb{R}^3$  be the coordinates and surface that correspond to the 3D world.  $S_0$  itself is given by the zero-level surface of the following PCA functional:  $\hat{\varphi}(\mathbf{X}_0, w) = \bar{\varphi}(\mathbf{X}_0) + \sum_0^k w_i \psi_i(\mathbf{X}_0)$ . That is,  $S_0 = \{\mathbf{X}_0 \in \mathbb{R}^3 : \hat{\varphi}(\mathbf{X}_0, w) = 0\}$ . Then one can locate the  $S$  in the camera referential via the transformation  $g \in SE(3)$ , such that  $S = g(S_0)$ . Writing this point-wise yields  $\mathbf{X} = g(\mathbf{X}_0) = \mathbf{R}\mathbf{X}_0 + \mathbf{T}$ , where  $\mathbf{R} \in SO(3)$  and  $\mathbf{T} \in \mathbb{R}^3$ .

##### A. Gradient Flow for 3D Tracking

We follow here the set-up of [3] to which we refer the reader for all of the details. Assume that if the correct 3D pose and shape are given, then the projection of the "occluding curve", i.e.  $\hat{c} = \pi(C)$ , delineates the boundary that optimally separates or segments a 2D object from its background. Further assuming that the image statistics between the 2D object and its background are distinct and are generally separable, we use the Bhattacharyya energy function defined previously. Since in our derivation here, we do not need its exact form, we simply note that it may be written in the

following general manner:

$$E = \int_R r_o(I(\mathbf{x}), \hat{c}) d\Omega + \int_{R^c} r_b(I(\mathbf{x}), \hat{c}) d\Omega \quad (16)$$

where  $r_o : \chi, \Omega \mapsto \mathbb{R}$  and  $r_b : \chi, \Omega \mapsto \mathbb{R}$  are suitably defined functionals derived from the from the Bhattacharyya formulation. Thus these functionals measure the similarity of the image pixels with a statistical model over the regions  $R$  and  $R^c$ , respectively, and  $\chi$  corresponds to the photometric variable of interest.

We need to optimize Equation (16) with respect to a finite parameter set denoted as  $\xi = \{\xi_1, \xi_2, \dots, \xi_m\}$  where  $m$  being the number of elements in the respective set:

$$\frac{\partial E}{\partial \xi_i} = \int_{\hat{c}} \left( r_o(I(\mathbf{x})) - r_b(I(\mathbf{x})) \right) \left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} \quad (17)$$

where the "silhouette" curve is parameterized by the arc length  $\hat{s}$  with the corresponding outward normal  $\hat{\mathbf{n}}$ .

Assuming that parameter  $\xi_i$  acts on 3D coordinates, the above line integral may be difficult to compute since  $\hat{c}$  and  $\hat{\mathbf{n}}$  lie in the 2D image plane. Hence it is more convenient to express the above line integral around the "occluding curve"  $C$ , which is parameterized by  $s$ . This can be done as follows. Write

$$\left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \left\langle \frac{\partial \pi(C)}{\partial \xi_i}, J \frac{\partial \pi(C)}{\partial s} \right\rangle ds \quad (18)$$

where  $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ , and this yields the following expression

$$\begin{aligned} \left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} &= \frac{1}{Z^3} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix} \frac{\partial \mathbf{X}}{\partial s} \right\rangle ds \\ &= \frac{1}{Z^3} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} \right\rangle ds \\ &= \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_X \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \right\rangle ds \end{aligned} \quad (19)$$

where  $K$  denotes the Gaussian curvature, and  $\kappa_X$  and  $\kappa_t$  denote the normal curvatures in the directions  $\mathbf{X}$  and  $\mathbf{t}$ , with  $\mathbf{t}$  being the vector tangent to the curve  $C$  at the point  $\mathbf{X}$ , i.e.  $\mathbf{t} = \frac{\partial \mathbf{X}}{\partial s}$ . If we now plug the result of Equation (19) into Equation (17), we arrive at the following flow

$$\begin{aligned} \frac{\partial E}{\partial \xi_i} &= \int_C \left( r_o(I(\pi(\mathbf{X}))) - r_b(I(\pi(\mathbf{X}))) \right) \cdot \\ &\quad \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_X \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \right\rangle ds \end{aligned} \quad (20)$$

Note, that in the above derivation we made no assumption on the type of finite set. That is, we show that the overall framework is essentially “blind” to whether we optimize over the shape weights or pose parameters. What is important is how the functional in Equation 17 is lifted from the “silhouette” curve to the “occluding curve” so that the gradient can be readily computed. The remaining terms in Equation (20) can be seen as “voting” weights for a particular point on the  $C$ .

## V. RESULTS

In this section, we present experimental tracking results that demonstrate the algorithm’s robustness to changes in scale as well as being able to cope with varying degrees of occlusion. Like that of [4], we generate a catalog of 8 shapes that depict several 3D Toy Elephants to perform the task of shape analysis. In particular, one can use stereo reconstruction techniques, or range scanners to obtain accurate 3D models. Moreover, we have calibrated the camera that is responsible for acquiring the images. That is, the focal length is 671 with principle point to be roughly the center of the image.

In the work of [20], the task of tracking can be decoupled into two fundamental parts: **deformation**, which is a finite group acting on the target, and **deformation**, which is the small, but infinite dimensional, perturbations that occur. However, because we approach tracking with only a finite set of parameters, namely the Euclidean group of  $g \in SE(3)$ , we do not need to directly identify the types of motion. For example, this can be seen if one were to pan a camera causing a deformation of the object projected onto the 2D image plane.

Thus, in the first set of experiments, we present a typical tracking example that exhibits large changes of scale. This can be seen in Figure 1. Interestingly, if one were to use a purely 2D shape based learning methodology like that of [17], one would have to learn every possible projection of the 3D object object onto to the 2D image plane (if no prior knowledge is given about the aspect of the projection). Moreover, and more importantly, we are able to return the 3D pose of the object, which is a drawback to the method proposed in typical 2D tracking algorithms [11]. Note, the principle modes was chosen to be  $k = 4$

In Figure 2, we demonstrate the algorithm’s robustness to occlusion and clutter. In this tracking sequence, the toy elephant is again stationary while we pass a

white marker through the camera’s view. Interestingly, even though the sequence exhibits partial occlusion, we note the differences in intensity between the elephant and the marker. Without any notion of dynamics, we are able to successfully maintain track throughout the sequence. Moreover, the object is presented in clutter that have similar intensity patterns and textures (e.g., black laptop behind the elephant). Employing a shape prior, we are able to properly segment the object. Note, the principle modes was chosen to be  $k = 4$ .

## VI. CONCLUSIONS

In this work, we exploited knowledge of 3D models for tracking in controlled active vision. A number of modalities now provide such data including LADAR. A key to the success of the tracker is a good segmentation procedure as well being able to estimate the 3D pose. In this work, using statistical learning techniques, we show how to train the tracking on a catalogue of 3D shapes, which enables one to employ a shape prior in 3D tracking tasks. Moreover, this allows us to detect and track deformable objects in cluttered, uncertain, and noisy environments. In future work we plan to add an estimation scheme to our 3D tracker based on particle filtering as in [11]. The geometric observer framework (which is based on shape metrics) of [9] is also very relevant in this regard.

## ACKNOWLEDGEMENTS

This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149. Information on the National Centers for Biomedical Computing can be obtained from <http://nihroadmap.nih.gov/bioinformatics>. This work was also funded in part by grants from NSF, AFOSR, ARO as well as by a grant from NIH (NAC P41 RR-13218) through Brigham and Women’s Hospital.

## REFERENCES

- [1] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation* **15** (2003), pp. 1373-139615.
- [2] A. Blake and M. Isard, *Active Contours*, Springer-Verlag, 1998.
- [3] S. Dambreville, R. Sandhu, A. Yezzi, and A. Tannenbaum, “A geometric approach to joint 2D region-based segmentation and 3D pose estimation using a 3D shape prior,” submitted to *SIAM Imaging Science*.
- [4] R. Sandhu, S. Dambreville, A. Yezzi, and A. Tannenbaum, “Non-rigid 2D-3D pose estimation and 2D image segmentation,” *IEEE CVPR*, 2009.

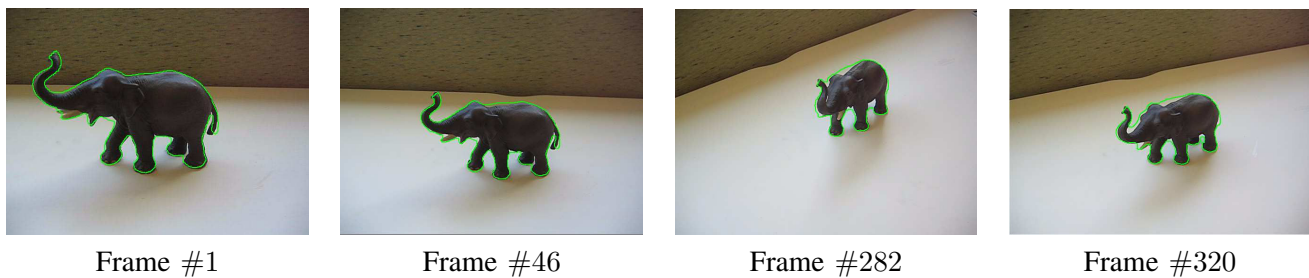


Fig. 1. Tracking through changes in scale and view. Employing a catalog of 3D shapes, we are able to derive a single novel 3D shape that account for varying 2D projections.

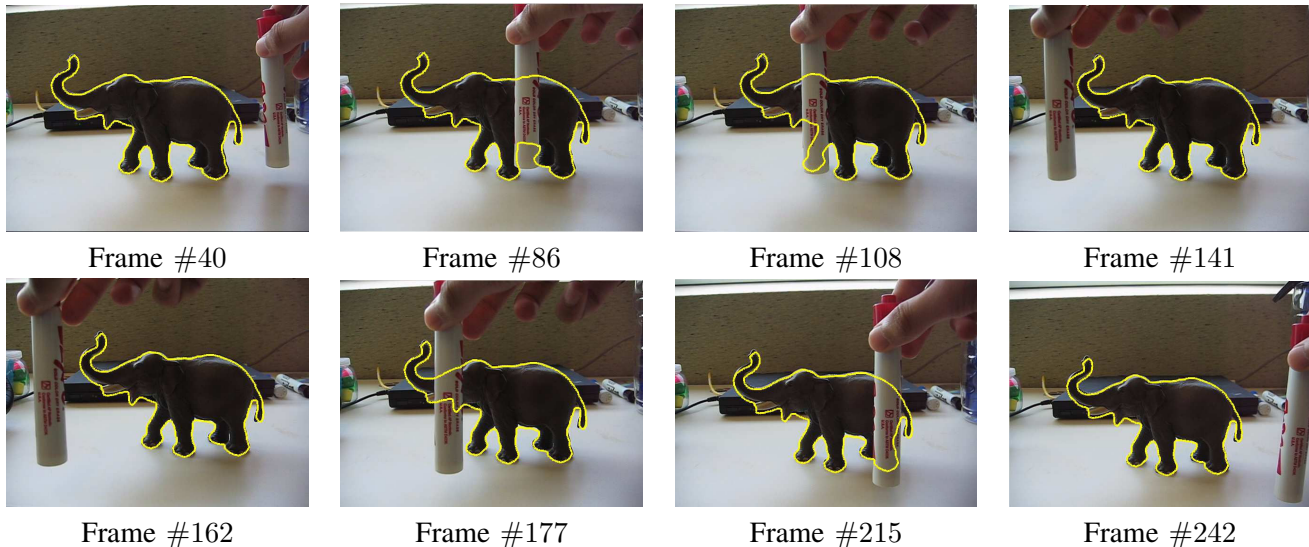


Fig. 2. Tracking through occlusion. In particular, several images are shown for a tracking sequence in which a white marker occludes parts of the elephant. Note the strong differences in intensity between the elephant and marker.

- [5] D. L. Donoho and C. Grimes, "Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data," Technical Report TR-2003-08, Department of Statistics, Stanford University, 2003.
- [6] M. Leventon, E. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," *EEE CVPR*, pages 1316-1324, 2000.
- [7] O. Michailovich, Y. Rathi, and A. Tannenbaum, "Image segmentation using active contours driven by the Bhattacharyya gradient flow," *IEEE Trans. Image Processing*, vol. 16, pp. 2787-2801, 2007.
- [8] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems," arXiv:math/0503445v1, 2005.
- [9] M. Niethammer, P. Vela, and A. Tannenbaum, "Geometric observers for dynamically evolving curves," *IEEE PAMI* **30** (2008), pp. 1093-1108.
- [10] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer Verlag, 2003.
- [11] Y. Rathi and A. Tannenbaum, "A generic framework for tracking using particle filter with dynamic shape prior," *IEEE Trans. Image Processing* **16** (2007), pp. 1370-1382.
- [12] D. Ridder and R. Duin, "Locally linear embedding for classification," Tech. Rep. PH-2002-01, Pattern Recognition Group, Delft University of Technology, 2002.
- [13] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323-2326, 2000.
- [14] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis* Cambridge University Press, 2001.
- [15] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 2nd edition, 1999.
- [16] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science* **290** (2000), pp. 2319-2323.
- [17] A. Tsai, T. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, E. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Trans. on Medical Imaging* **22** (2003), pp. 1371-1383.
- [18] R. Sandhu, S. Dambreville, and A. Tannenbaum, "Particle filtering for registration of 2D and 3D point sets with stochastic dynamics," *IEEE CVPR* (2008).
- [19] S. Dambreville, R. Sandhu, A. Yezzi, and A. Tannenbaum, "Robust 3D Pose Estimation and Efficient 2D Region-Based Segmentation from a 3D Shape Prior," *Proceedings of ECCV* pp. 169-182, 2008.
- [20] A. Yezzi and S. Soatto, "Deformation: Deforming motion, shape average and the joint registration and approximation of structures in images," *Int. Journal Comp. Vision* **53** (2003), pp. 153-167.