Published in final edited form as:

IEEE Trans Pattern Anal Mach Intell. 2011 June ; 33(6): . doi:10.1109/TPAMI.2010.162.

A Nonrigid Kernel-Based Framework for 2D-3D Pose Estimation and 2D Image Segmentation

Romeil Sandhu [Member, IEEE],

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

Samuel Dambreville [Member, IEEE],

Georgia Institute of Technology, 313 Ferst Drive, Atlanta, GA 30318, and the Boston Consulting Group.

Anthony Yezzi [Member, IEEE], and

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332.

Allen Tannenbaum [Fellow, IEEE]

Department of Biomedical Engineering, Georgia Institute of Technology, 313 Ferst Drive, Atlanta, GA 30318, and the Department of Electrical Engineering, Technion-Israel Institute of Technology, Haifa, Israel.

Romeil Sandhu: rsandhu@gatech.edu; Samuel Dambreville: dambreville_samuel@yahoo.fr; Anthony Yezzi: anthony.yezzi@ece.gatech.edu; Allen Tannenbaum: tannenba@ece.gatech.edu

Abstract

In this work, we present a nonrigid approach to jointly solving the tasks of 2D-3D pose estimation and 2D image segmentation. In general, most frameworks that couple both pose estimation and segmentation assume that one has exact knowledge of the 3D object. However, under nonideal conditions, this assumption may be violated if only a general class to which a given shape belongs is given (e.g., cars, boats, or planes). Thus, we propose to solve the 2D-3D pose estimation and 2D image segmentation via nonlinear manifold learning of 3D embedded shapes for a general class of objects or deformations for which one may not be able to associate a skeleton model. Thus, the novelty of our method is threefold: First, we present and derive a gradient flow for the task of nonrigid pose estimation and segmentation. Second, due to the possible nonlinear structures of one's training set, we evolve the preimage obtained through kernel PCA for the task of shape analysis. Third, we show that the derivation for shape weights is general. This allows us to use various kernels, as well as other statistical learning methodologies, with only minimal changes needing to be made to the overall shape evolution scheme. In contrast with other techniques, we approach the nonrigid problem, which is an infinite-dimensional task, with a finite-dimensional optimization scheme. More importantly, we do not explicitly need to know the interaction between various shapes such as that needed for skeleton models as this is done implicitly through shape learning. We provide experimental results on several challenging pose estimation and segmentation scenarios.

Index Terms

3D pose estimation; image segmentation; statistical learning; kernel PCA

1 Introduction

TWO well-studied problems in computer vision are the fundamental tasks of 2D image segmentation and 3D pose estimation from a 2D scene. In this paper, by leveraging the advantages of certain techniques from each problem, we couple both tasks in a variational and *nonrigid* manner through a single energy functional. Similarly to shape derivatives, this can be accomplished by first deriving a gradient flow that is valid for any arbitrary finite set of parameters (i.e., shape coefficients, wavelet coefficients, and pose transformations). We are then able to use nonlinear manifold learning techniques such as kernel PCA to solve the nonrigid 2D-3D pose estimation and 2D segmentation task by evolving both the shape weights and the pose parameters in 3D space. In other words, this work can be viewed as a generalization to our previous framework presented in [15], in which we include the knowledge of multiple 3D shapes rather than assuming the exact knowledge of a single 3D shape prior. However, to appreciate the contributions presented in this paper, we briefly revisit some of the key results that have been made pertaining to both fields of interest.

2D-3D pose tracking or pose estimation is concerned with relating the spatial coordinates of an object in the 3D world (with respect to a calibrated camera) to that of a 2D scene [22], [27]. Although the complete literature review is beyond the scope of this paper, most methodologies can be described as follows: First, one chooses a local geometric descriptor (e.g., points [33], lines [18], [28], or curves [19], [39]) or image intensity [4] that can best quantify features on the image to its corresponding 3D counterpart. Then, explicit point correspondences are established in order to solve for the pose transformation. As with most correspondence-based algorithms which rely on local features, it can be readily seen that these techniques may suffer from the existence of homologies (due to noise, clutter, or occlusions). Aside from the nontrivial task of establishing correspondences, many 2D-3D pose estimation techniques make certain (sometimes rather restrictive) assumptions on the class of shapes that they can handle. Recently, the authors of [39] have proposed relaxing such restrictions by focusing on free-form objects. However, even for this type of algebraic approach it may become increasingly difficult to estimate the pose for an arbitrary or complex shape. Moreover, and more importantly, the above methods typically constrain their approaches to the knowledge of a prespecified 3D model. To overcome this constraint, nonrigid algorithms have appeared in the area of human pose estimation [17], [38], [1]. While we should note that the focus of our paper is not specific to this area of computer vision, the proposed framework is closely related if one were to learn a large class of deformations, as opposed to rigid objects. However, in contrast to methods such as and [38] and [17], our approach relies on the surface differential geometry of a 3D model. This allows us to eliminate the need for point correspondences altogether while still being able to deal with a complex shape.

Image segmentation consists of partitioning a scene into an “object” and a “background” [3]. In particular, we will restrict our approach to segmentation to that of the geometric active contour (GAC) framework, whereby a curve is evolved continuously until it satisfies a stopping criterion that coincides with the object’s boundaries. Certain variational approaches relied on characterizing the object by local features such as edges to drive the curve evolution; see [7], [23] and the references therein. However, these edge-based techniques were shown to be susceptible to noise and missing information. Consequently, an alternative characterization, based on so-called region-based methods, is to assume that the “object”

and “background” possess differing image statistics (see [8], [32], [30]). Although this improves segmentation results, the assumption may not hold due to clutter or occlusions. This has resulted in the proposed use of a shape prior to restrict the evolution of the active contour [26], [12], [10], [14], [47]. We should note that even though the framework presented in this paper shares similarities with shape-based methods, one fundamental difference with our methodology is that we derive a novel 3D shape prior from a catalog of 3D shapes to do 2D image segmentation rather than deriving a 2D shape prior from a collection of 2D images. Thus, a key benefit is that we are able to reduce computational complexity with regard to pure 2D shape-based schemes via a compact shape representation. Fig. 1 illustrates this notion.

1.1 Relation to Shape Optimization

Given that we assume and utilize a training set of rigid shapes to perform segmentation (and registration in 3D) for a particular object of interest, our algorithm proposes solving a typical shape optimization problem. Thus, it can be related to a wealth of research found in the computer vision community [16], [9], [11]. Although it is beyond the scope of this work, we refer the reader to [9], [10] for a discussion on how one can define the notion of “distance” between the space of shapes. However, it is noteworthy to mention general differences of the proposed approach with regard to some of these alternative methods. For example, in [46], the authors introduced an interesting solution for nonrigid registration and segmentation. Specifically, they employ a global transformation in conjunction with a local deformation by incorporating the notion of uncertainty through a series of control points. Using kernel methods to model the shape statistics, their approach yields impressive results. In theoretically comparing our algorithm, we note that we not require a two-step process of global segmentation and local deformation as this is done in a coupled process. Moreover, our algorithm attempts to solve a 2D shape optimization task through a set of 3D shapes belonging to a specific class.

1.2 Relation to Previous Work

It is interesting to note that while 2D-3D pose estimation and 2D image segmentation are closely related, there exist few methodologies that try to couple both tasks in a unified framework. An early attempt to solve the problem of viewpoint dependence for differing aspects of a 3D object is given in [36]. In their work, the authors proposed a region-based active contour that employs a unique shape prior, which is represented through a generalized cone based on a single reference view of an object. Although the method performs well under different changes in aspect, it is not able to cope with a view of an object that is substantially different from the reference view.

In addition, even though we have restricted our discussion to the GAC framework, we should also note that recent work has been done in simultaneous pose estimation and segmentation via dynamic graph cuts [6], [24]. In this approach, the authors proposed an articulated shape prior through a stick-man or skeleton model. Then, in order to capture deformations occurring to the object, one must optimize over a set of predefined parameters corresponding to specific motions in a model’s movement. In relation to this work, our focus would be to accurately quantify the deformation through a set of 3D models like that of [1], through kernel principal component analysis (KPCA). More importantly, we are able to incorporate a general class of shapes for which one may not be able to associate a skeleton model. Also, the above methodologies differ in the segmentation approach used (i.e., graph cuts versus active contours), in which we note that each method has its advantages and disadvantages.

While it is not the intended contribution, one could also alternatively view the proposed framework as 3D reconstruction from a single or multiple views. Although this area is abundant with methodologies, we refer the reader to several works that propose solving this difficult task [5], [44], [21], [40]. In particular, and like that of our paper, in [44], the authors proposed solving this task by inducing a prior on the possible reconstructions. However, aside from using several views, the prior does not incorporate information about a specific class of objects, but rather is employed as a prior for smoothness. Recent work in [40] utilized a probabilistic graph model to reconstruct an object of a certain class from a single 2D view. A key difference between our method and that of [40] is the manner in which we approach the task itself. That is, although we use the image as measure of fidelity, we do not incorporate a graphical model. Nevertheless, we do not consider this to be a limitation, but rather a philosophical difference.

In relation to the framework presented in this paper, the authors in [37] and [43] also proposed a solution to solve the joint task of pose estimation and segmentation for the case of *rigid* objects. In [37], the authors accounted for a variation in the projection of the 3D shape by evolving an active contour in conjunction with the 3D pose parameters to minimize a joint energy functional. While this is less restrictive, the algorithm optimizes over an infinite-dimensional active contour as well as the set of finite pose parameters. Moreover, in order for one to determine the shape prior and the corresponding 3D pose, costly back-projections must be made through ICP-like correspondences. An extension is considered in [43] whereby the authors successfully eliminate the need to evolve the active contour by performing a minimization of 3D pose parameters instead. However, the costly back-projections and correspondences remain.

Thus, while one may try to expand upon other frameworks such as [37], [43] for multiple shapes, the reasoning for why we have chosen to extend the methodology presented in [15] is as follows: In previous work, we derived a variational framework to jointly segment a *rigid* object in a 2D image and estimated the corresponding 3D pose through the use of a 3D shape prior. Specifically, our algorithm uses a region-based segmentation method to continuously drive the pose estimation process. This results in a global approach that avoids using local features or ICP-like correspondences by relying on surface differential geometry to link geometric properties of the model surface and its corresponding projection in the 2D image domain. The methodology is motivated by similar approaches that were originally constructed for stereo reconstruction from multiple cameras [49], [50] and further extended for camera calibration [48]. Consequently, knowledge of a single 3D object is exploited to its full extent within our framework. The question that is to be addressed in this work is *how can one fully exploit the knowledge of a general class of 3D shapes as opposed to a single 3D model?*

1.3 Proposed Contributions

Thus, the key contribution in this paper is to extend the method in [15] to include the knowledge of multiple 3D shapes. This is done by adopting one of the shape-based approaches of [47], [31], [13] to the problem at hand. In other words, we evolve shape parameters or modes of variations that are obtained from performing KPCA on a collection of 3D shapes. Unlike the conference version of this work [42], whereby we successfully employed the classical PCA methodology for shape analysis, KPCA allows us to exploit the nonlinearities typically found in one's training set. In addition, we show that the derivation for shape weights is general, allowing for various kernels and statistical learning methodologies to be employed without the need to make drastic changes to the overall shape evolution scheme. As a result, we approach the nonrigid task through an optimization of a *finite* set of parameters.

The remainder of this paper is organized as follows: In the next section, we review the fundamental concepts associated with Kernel PCA, which include a pre-image approximation. In Section 3, we begin with generalization of the 2D-3D pose estimation and 2D segmentation gradient flow for an arbitrary set of finite parameters. We then provide details for evolving both the shape parameters, which are obtained from performing KPCA on a collection of 3D shapes, and the corresponding pose of an object. Numerical implementation details are given in Section 4. In Section 5, we present experimental results that highlight the robustness of the technique to noise, clutter, and occlusions, as well as the ability to segment a novel shape that is not apart of the specified training set. Finally, we discuss future work in Section 6.

2 Kernel Principal Component Analysis Review

In this section, we review the fundamental concepts associated with kernel PCA as well as the pre-image approximation used, which we will need in the sequel.

2.1 KPCA Formulation

Let us begin with a set of data points $\{X_1, X_2, \dots, X_N\}$ that are members of the input space $\mathcal{M} \in \mathbb{R}^n$. This set can be mapped to a possibly higher dimensional feature space, denoted by $\mathcal{H}$, via the nonlinear map $\phi : \mathbb{R}^n \mapsto \mathcal{H}$. Moreover, this map does not need to be explicitly known. Indeed, one can introduce a Mercer kernel, which is defined to be a function $k(X_a, X_b)$ such that for all data points X_i , the kernel matrix

$$\mathbf{K} = \begin{pmatrix} k(X_1, X_1) & k(X_1, X_2) & \dots & k(X_1, X_N) \\ k(X_2, X_1) & \ddots & & \\ \vdots & & k(X_i, X_j) & \\ k(X_N, X_1) & & & k(X_N, X_N) \end{pmatrix}$$

is symmetric positive [31]. According to Mercer's Theorem [29], computing $k(X_a, X_b)$ as a function of $\mathcal{M} \times \mathcal{M}$ amounts to computing the inner scalar product in $\mathcal{H}$: $k(X_a, X_b) = \langle \phi(X_a), \phi(X_b) \rangle$, with $(X_a, X_b) \in \mathcal{M} \times \mathcal{M}$. This scalar product in $\mathcal{H}$ defines a distance $d_{\mathcal{H}}$. For example, the L_2 distance is

$$d_{\mathcal{H}}^2(\phi(X_a), \phi(X_b)) = \|\phi(X_a) - \phi(X_b)\|^2 = k(X_a, X_a) - 2k(X_a, X_b) + k(X_b, X_b).$$

If we now consider the input to be a collection of shapes, we can then perform the KPCA method as presented in [31]. Let $\mathcal{T} = \{X_1, X_2, \dots, X_N\}$ be a set of training data. The centered kernel matrix $\tilde{\mathbf{K}}$ corresponding to $\mathcal{T}$ is defined as

$$\tilde{\mathbf{K}}(i, j) = \langle (\phi(X_i) - \bar{\phi}), (\phi(X_j) - \bar{\phi}) \rangle = \langle \tilde{\phi}(X_i), \tilde{\phi}(X_j) \rangle = \tilde{k}(X_i, X_j), \text{ for } (i, j) \in [1, N], \quad (1)$$

with

$$\bar{\phi} = \frac{1}{N} \sum_{i=1}^N \phi(X_i),$$

where

$$\tilde{\phi}(X_i) = \phi(X_i) - \bar{\phi}.$$

In addition, since $\tilde{\mathbf{K}}$ is symmetric, it can be decomposed as

$$\tilde{\mathbf{K}} = \mathbf{U} \mathbf{S} \mathbf{U}^T, \quad (2)$$

where $\mathbf{S} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$ is a diagonal matrix containing the eigenvalues of $\tilde{\mathbf{K}}$. $\mathbf{U} = [u_1, u_2, \dots, u_N]$ is an orthonormal matrix, where the columns $u_i = [u_{i1}, u_{i2}, \dots, u_{iN}]^T$ are the eigenvectors corresponding to the eigenvalues λ_i s. Furthermore, it can be shown that

$$\tilde{\mathbf{K}} = \mathbf{H} \mathbf{K} \mathbf{H}, \quad (3)$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{N} \mathbf{1} \mathbf{1}^T$, $\mathbf{1} = [1, 1, \dots, 1]^T$ is an $N \times 1$ vector, and $\mathbf{I}$ is an $N \times N$ identity matrix.

Let $\mathbf{C}$ denote the covariance matrix of the elements of the training set mapped by ϕ . Then the n th (orthonormal) eigenvector of $\mathbf{C}$ in the feature space is given as follows:

$$V_n = \sum_{i=1}^N \frac{u_{ni}}{\sqrt{\lambda_n}} \tilde{\phi}(X_i) = \frac{1}{\sqrt{\lambda_n}} \tilde{\phi} u_n, \quad (4)$$

where $\tilde{\phi} = [\tilde{\phi}(X_1), \tilde{\phi}(X_2), \dots, \tilde{\phi}(X_N)]$. That is, $\tilde{\phi} \in \mathbb{R}^{n \times N}$. The subspace of the feature space $\mathcal{H}$ will be referred to as the *kernel PCA space*.

Now, let X be *any* element of the input space $\mathcal{M}$. The projection of X on the kernel PCA space, spanned by the first l eigenvectors of $\mathbf{C}$, is then given by

$$P^l \phi(X) = \sum_{n=1}^l \omega_n V_n + \bar{\phi}, \quad (5)$$

with $\omega = [\omega_1, \omega_2, \dots, \omega_l]$ being the KPCA shape weights or coefficients.

We note that while the above formulation allows us to construct the projection from a linear combination involving the shape coefficients, it is only valid in the feature space. That is, given a test point $\mathbf{x} \in \mathcal{M}$, we would ideally like to compute its projection in the image space $\hat{\mathbf{x}} = \phi^{-1}(P^l \phi(\mathbf{x}))$. Unfortunately, because the map $\phi : \mathbb{R}^n \mapsto \mathcal{H}$ is unknown, one cannot directly obtain this projection. This is referred to in the literature as the *preimage problem* [25]. In this work, we use the noniterative preimage result of [13] to directly evolve the KPCA coefficients. However, before doing so, we first present a few popular kernels and their corresponding pre-image results used in the literature for shape-based learning.

2.2 KPCA Kernels

We now present several kernels that allows us to perform learning of shapes using linear and nonlinear PCA.

2.2.1 Linear PCA—In [26], [47], a method is presented to learn shape variations by employing PCA on a training set of shapes (closed curves) represented as the zero-level sets of signed distance functions (SDF). In order to perform PCA, let us begin with the polynomial kernel. This is given by

$$k_{\varphi_P}(\psi_i, \psi_j) = (c + \langle \psi_i, \psi_j \rangle)^d, \quad (6)$$

where c is any constant, d is the degree (odd) of the polynomial, and ψ_i is the SDF associated with a shape in one's training set. Then, by choosing $c = 0$ and $d = 1$, we arrive at the following kernel used to perform classical PCA on SDFs:

$$k_{id}(\psi_i, \psi_j) = \langle \psi_i, \psi_j \rangle = \int \int \int \psi_i(u, v, k) \psi_j(u, v, k) du dv dk, \quad (7)$$

for the SDFs ψ_i and $\psi_j: \mathbb{R}^3 \mapsto \mathbb{R}$. The subscript id stands for the identity function: When performing linear PCA, the kernel used is the inner scalar product in the input space. Hence, the corresponding mapping function $\phi = id$. One should note that this latter integral is infinite if there is no bound. However, since this is being used in a shape learning context PCA, there is an assumption on the size of the shapes within the given training set that gives the necessary boundedness and finiteness result. This certainly affects the distance, and makes it nonintrinsic.

2.2.2 Nonlinear PCA—Choosing various types of nonlinear kernel functions $k(X_a, X_b)$ is the basis of nonlinear PCA. The exponential kernel has been a popular choice in the machine learning community and has proven to nicely extract nonlinear structures from data sets; see, e.g., [31]. Using SDFs for representing shapes, this kernel is given by

$$k_{\phi_\sigma}(\psi_i, \psi_j) = e^{\frac{-\|\psi_i - \psi_j\|^2}{2\sigma^2}}, \quad (8)$$

where σ^2 is a variance parameter estimated a priori and $\|\psi_i - \psi_j\|^2$ is the squared L_2 -distance between two SDFs ψ_i and ψ_j . The subscript ϕ_σ stands for the nonlinear mapping corresponding to the exponential kernel. This mapping also depends on the choice of σ .

2.3 Pre-Image Approximation

In this section, we revisit the closed-form pre-image approximation proposed in [13], [34]. It can be seen that although the ϕ map is not necessarily known, we would ideally like to reconstruct the pre-image $\hat{\mathbf{x}}$ of a corresponding test point $\mathbf{x} \in \mathcal{M}$ such that the distance between the feature point $\phi(\hat{\mathbf{x}})$ and the projection in the PCA space $P^l \phi(\mathbf{x})$ is minimized, i.e.,

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{M}} \|\phi(\hat{\mathbf{x}}) - P^l \phi(\mathbf{x})\|^2.$$

In terms of level sets, this can be achieved by minimizing the following error:

$$\rho(\hat{\psi}) = \|\phi(\hat{\psi}) - P^l \phi(\psi)\|^2 = k(\hat{\psi}, \hat{\psi}) - 2\langle \phi(\hat{\psi}), P^l \phi(\psi) \rangle + \|P^l \phi(\psi)\|^2, \quad (9)$$

where $P^l \phi(\psi)$ is the projection of *any* SDF in kernel PCA space. Using the kernel notation, we can rewrite the middle term:

$$\langle \phi(\hat{\psi}), P^l \phi(\psi) \rangle = \phi(\hat{\psi})^T \left[\sum_{n=1}^l \omega_n V_n + \bar{\varphi} \right] = \phi(\hat{\psi})^T \left[\sum_{n=1}^l \omega_n \left[\left(\sum_{i=1}^N \frac{u_{ni}}{\sqrt{\lambda_n}} \phi(\psi_i) \right) - \bar{\varphi} \left(\sum_{i=1}^N \frac{u_{ni}}{\sqrt{\lambda_n}} \right) \right] + \bar{\varphi} \right] = \sum_{i=1}^N \tilde{\gamma}_i k(\hat{\psi}, \psi_i), \quad (10)$$

where

$$\gamma_i = \sum_{n=1}^l \frac{\omega_n u_{ni}}{\sqrt{\lambda_n}} \text{ and } \tilde{\gamma}_i = \gamma_i + \frac{1}{N} \left(1 - \sum_{j=1}^N \gamma_j \right).$$

Plugging the result of (10) into (9), the extremum can be obtained by setting $\nabla_{\psi}\hat{\rho} = 0$. Also, we prefer that the preimage have a closed-form solution so that it can be used as a basis for our energy functional in which we would like to evolve the KPCA shape weights directly in the input space.

2.3.1 Preimage for Linear PCA—We begin with a pre-image approximation for the polynomial kernel presented in Section 2.2.2. From this, we can then simplify the resulting pre-image approximation to that of linear PCA. By setting $\nabla_{\psi}\hat{\rho} = 0$, the following pre-image of the projection in the KPCA space is:

$$\hat{\psi} = \sum_{i=1}^N \tilde{\gamma}_i \left(\frac{k(\hat{\psi}, \psi_i)}{k(\hat{\psi}, \hat{\psi})} \right) \cdot \left(\frac{k(\hat{\psi}, \psi_i)}{k(\hat{\psi}, \hat{\psi})} \right)^{-\frac{1}{d}} \psi_i = \sum_{i=1}^N \tilde{\gamma}_i \left(\frac{k(\hat{\psi}, \psi_i)}{k(\hat{\psi}, \hat{\psi})} \right)^{\frac{d-1}{d}} \psi_i = \sum_{i=1}^N \tilde{\gamma}_i \left(\frac{c + \langle \hat{\psi}, \psi_i \rangle}{c + \langle \hat{\psi}, \hat{\psi} \rangle} \right)^{d-1} \psi_i. \quad (11)$$

However, if we use the approximation $\phi(\psi) \approx P^l \phi(\psi)$, which amounts to assuming that $d_{\mathcal{H}}^2(P^l \phi(\psi), \phi(\psi_i)) \propto d_{\mathcal{H}}^2(\phi(\hat{\psi}), \phi(\psi_i))$, one has

$$k(\hat{\psi}, \psi_i) = \frac{1}{2} (\|P^l \phi(\psi)\|^2 + k(\psi_i, \psi_i) - d_{\mathcal{H}}^2(P^l \phi(\psi), \phi(\psi_i))).$$

Then, the following expression to reconstruct the preimage is obtained:

$$\hat{\psi} = \sum_{i=1}^N \tilde{\gamma}_i \left(\frac{\|P^l \phi(\psi)\|^2 + k(\psi_i, \psi_i) - d_{\mathcal{H}}^2(P^l \phi(\psi), \phi(\psi_i))}{2\|P^l \phi(\psi)\|^2} \right)^{\frac{d-1}{d}} \psi_i.$$

Interestingly, if we now take $d=1$ in (11), we get the expression for performing linear PCA, i.e.,

$$\hat{\psi} = \sum_{i=1}^N \tilde{\gamma}_i \psi_i. \quad (12)$$

More importantly, we can now see that the pre-image approximation $\hat{\psi}$ presented in (12) depends on the shape weight ω_i . In a similar manner, we can arrive at an energy formulation for nonlinear PCA. This is discussed next.

2.3.2 Pre-Image for Nonlinear PCA—For the exponential kernel in (8), which involves the implicit representation of SDFs, setting $\nabla_{\psi}\hat{\rho} = 0$ yields

$$\hat{\psi} = \frac{\sum_{i=1}^N \tilde{\gamma}_i k(\hat{\psi}, \psi_i) \psi_i}{\sum_{i=1}^N \tilde{\gamma}_i k(\hat{\psi}, \psi_i)} = \frac{\sum_{i=1}^N \tilde{\gamma}_i \exp(-\|\hat{\psi} - \psi_i\|^2 / (2\sigma^2)) \psi_i}{\sum_{i=1}^N \tilde{\gamma}_i \exp(-\|\hat{\psi} - \psi_i\|^2 / (2\sigma^2))}. \quad (13)$$

While this expression has been used to estimate the preimage via an iterative-based approach [25], we use the close-form approximation proposed by [34]. In their work, the authors assume that $\psi(\hat{\psi}) \approx P^l \psi(\hat{\psi})$. Moreover, they also assume that the Mercer kernel which is chosen can be inverted. Thus, for an exponential kernel of (8), one has

$$d_{\mathcal{H}}^2(\hat{\psi}, \psi_i) = -2\sigma^2 \log \left\{ \frac{1}{2} (2 - d_{\mathcal{H}}^2(\phi(\hat{\psi}), \phi(\psi_i))) \right\}.$$

Plugging this result into (13) with the above approximation yields the following:

$$\hat{\psi} = \frac{\sum_{i=1}^N \tilde{\gamma}_i \exp(-d_{\mathcal{H}}^2(\hat{\psi} - \psi_i)/(2\sigma^2)) \psi_i}{\sum_{i=1}^N \tilde{\gamma}_i \exp(-d_{\mathcal{H}}^2(\hat{\psi} - \psi_i)/(2\sigma^2))} \approx \frac{\sum_{i=1}^N \tilde{\gamma}_i (1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i))) \psi_i}{\sum_{i=1}^N \tilde{\gamma}_i (1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i)))}. \quad (14)$$

Of course, the above approximations were made so that one can use kernel PCA to form a 3D shape embedded in a feature space $\mathcal{H}$ via the finite shape weights associated with the input space $\mathcal{M}$. However, before deriving the evolution scheme, let us first discuss the proposed pose estimation and segmentation framework.

3 Proposed Framework

We assume as with many machine learning techniques that we have a catalog of 3D shapes describing a particular object. Specifically, one can use stereo reconstruction methods [49] or range scanners to obtain accurate models, as shown in Fig. 2. From this, we derive a variational approach to perform the task of nonrigid 3D pose estimation and 2D image segmentation.

3.1 Some Notation and Terminology

Let S be the smooth surface in $\mathbb{R}^3$ defining the shape of the object of interest. With a slight abuse of notation, we denote by $\mathbf{X} = [X, Y, Z]^T$ the spatial coordinates that are measured with respect to the referential of the imaging camera. The (outward) unit normal to S at each point $\mathbf{X} \in S$ will then be denoted 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 on which the surface 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. In other words, if we define the “occluding” curve C to be the intersection of the visible and nonvisible regions of S , then the image curve is $\hat{c} = \pi(C)$.

Now, let $\mathbf{X}_0 \in \mathbb{R}^3$ and S_0 be the coordinates and surface that correspond to the 3D world, respectively. For example, if we choose the exponential kernel, S_0 is given as the zero-level surface of the following functional:

$$\hat{\psi}(\mathbf{X}_0, w) = \frac{\sum_{i=1}^N \tilde{\gamma}_i (1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i))) \psi_i(\mathbf{X}_0)}{\sum_{i=1}^N \tilde{\gamma}_i (1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i)))}. \quad (15)$$

That is, $S_0 = \{\mathbf{X}_0 \in \mathbb{R}^3 : \psi(\mathbf{X}_0, w) = 0\}$. Note, we have also kept the explicit dependence on w for ease of reading (e.g., when we compute the variation of this shape w.r.t. w in Section 3.3). 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 $\mathbf{T} \in \mathbb{R}^3$.

3.2 Gradient Flow

Let us begin with the assumption that if the correct 3D pose and shape were given, then the projection of the “occluding curve,” i.e., $\hat{c} = \pi(C)$, would delineate 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, we define an energy functional of the following:

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

where $r_o : \chi, \Omega \mapsto \mathbb{R}$ and $r_b : \chi, \Omega \mapsto \mathbb{R}$ are functionals measuring the similarity of the image pixels with a statistical model over the regions R and R^c , respectively. Also, χ corresponds to the photometric variable of interest. In the present work, r_o and r_b are chosen to be region-based functionals of [8], [32].

Now we want to optimize (16) with respect to a finite parameter set denoted as $\xi = \{\xi_1, \xi_2, \dots, \xi_m\}$. This is given as follows:

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

where the “silhouette” curve is parameterized by the arc length $\hat{s}$ with the corresponding outward normal $\hat{\mathbf{n}}$. Note, because we are only restricting r_o and r_b to be [8], [32], the last two terms can be shown to be 0. However, this is a special case, and one must take careful consideration when choosing the proper energy functional as these terms may tend not to be 0. We consider these energies for their simplicity, and more importantly, not to detract from the main contribution of the proposed method. In doing so, the result is as follows:

$$\frac{\partial E}{\partial \xi_i} \int_{\hat{c}} (r_o(I(\mathbf{x})) - r_b(I(\mathbf{x}))) \left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s}. \quad (18)$$

If we further assume that the parameter ξ_i acts on the 3D coordinates, the above line integral will be difficult to compute since $\hat{c}$ and $\hat{\mathbf{n}}$ are defined in the 2D image plane. Thus, it would be much more convenient if we can express the above line integral around the “occluding curve” C that lives in the 3D space and is parameterized by s . We briefly describe this lifting procedure, and refer the reader to [15] for all of the details. The image plane and surface are then related by

$$\left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \left\langle \frac{\partial \pi(C)}{\partial \xi_i}, J \frac{\partial \pi(C)}{\partial s} \right\rangle ds, \quad (19)$$

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

$$\left\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \frac{1}{Z^3} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix} \frac{\partial \mathbf{X}}{\partial s} \right\rangle ds = \frac{1}{Z^3} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} \right\rangle ds = \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_x \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \right\rangle ds. \quad (20)$$

Here, K denotes the Gaussian curvature, κ_x and κ_t denote the normal curvatures in the directions $\mathbf{X}$ and $\mathbf{t}$, respectively, where $\mathbf{t}$ is the vector tangent to the curve C at the point $\mathbf{X}$,

$$\text{i.e., } \mathbf{t} = \frac{\partial \mathbf{X}}{\partial s}.$$

If we now plug the result of (20) into (18), we arrive at the following flow:

$$\frac{\partial E}{\partial \xi_i} = \int_C (r_o(I(\pi(\mathbf{X}))) - r_b(I(\pi(\mathbf{X})))) \cdot \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_x \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \right\rangle ds. \quad (21)$$

Note that in the above derivation, we made no assumptions about the 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 (18) is lifted from the “silhouette” curve to the “occluding curve” so that the gradient can be readily

computed. In particular, the term $\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \rangle$ is what we will focus on in Sections 3.3 and 3.4.

3.3 Evolving the Shape Parameters

In this section, we compute the term $\langle \frac{\partial \mathbf{X}}{\partial \xi_i}, \mathbf{N} \rangle$ when the ξ_i corresponds to the shape weight obtained from performing KPCA on a collection of 3D models. Let $\xi = \omega = \{\omega_1, \omega_2, \dots, \omega_l\}$ with l being the number of principal modes used. In addition, the 3D coordinates $\mathbf{X}_0$, which are derived from the surface S_0 , are related by the constraint

$$\hat{\psi}(\mathbf{X}_0, w) = \frac{\sum_{i=1}^N \tilde{\gamma}_i (1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i))) \psi_i(\mathbf{X}_0)}{\sum_{i=1}^N \tilde{\gamma}_i (1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i)))} \text{ s.t. } \hat{\psi}(X_0(w), w) = 0. \quad (22)$$

The term $\langle \frac{\partial \mathbf{X}}{\partial \omega_n}, \mathbf{N} \rangle$ can then be computed as follows:

$$\left\langle \frac{\partial \mathbf{X}}{\partial \omega_n}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R} \mathbf{X}_0 + \mathbf{T}}{\partial \omega_n}, \mathbf{N} \right\rangle = \left\langle \mathbf{R} \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{R}^T \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{R}^T \mathbf{R} \mathbf{N}_0 \right\rangle = \left\langle \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{N}_0 \right\rangle. \quad (23)$$

Using the constraint on the zero-level surface and noting that $\frac{\nabla_{\mathbf{x}_0} \hat{\psi}}{\|\nabla_{\mathbf{x}_0} \hat{\psi}\|} = \mathbf{N}_0$, we then have that

$$0 = \frac{\partial}{\partial \omega_n} \hat{\psi}(\mathbf{X}_0(w), w) = \langle \nabla_{\mathbf{x}_0} \hat{\psi}, \frac{\partial \mathbf{X}_0}{\partial \omega_n} \rangle + \frac{\partial \hat{\psi}}{\partial \omega_n} = \langle \|\nabla_{\mathbf{x}_0} \hat{\psi}\| \mathbf{N}_0, \frac{\partial \mathbf{X}_0}{\partial \omega_n} \rangle + \frac{\partial \hat{\psi}}{\partial \omega_n},$$

which yields the following compact expression:

$$\left\langle \frac{\partial \mathbf{X}_0}{\partial \omega_n}, \mathbf{N}_0 \right\rangle = - \frac{1}{\|\nabla_{\mathbf{x}_0} \hat{\psi}\|} \cdot \frac{\partial \hat{\psi}}{\partial \omega_n}. \quad (24)$$

The general result presented in (24) provides the variation of the energy with respect to the shape parameters, and is one of the major contributions of this work. It was previously

shown that if one uses the linear PCA kernel, then $\frac{\partial \hat{\psi}}{\partial \omega_n} = V_n(\mathbf{X}_0)$. However, to exploit nonlinearities in the catalog of shapes, the exponential kernel is employed. The variation of the preimage for this kernel with respect to the shape weights is given as

$$\frac{\partial \hat{\psi}}{\partial \omega_i} = \frac{\sum_i^N \eta_i \cdot \psi_i}{\sum_i^N \mathcal{J}_i} - \frac{(\sum_i^N \eta_i)(\sum_i^N \mathcal{J}_i \cdot \psi_i)}{(\sum_i^N \mathcal{J}_i)^2}, \quad (25)$$

where

$$\mathcal{J}_i = \tilde{\gamma}_i \left(1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i)) \right),$$

$$\eta_i = \frac{u_{ni}}{\sqrt{\lambda_n}} - \frac{1}{N} \sum_j \frac{u_{nj}}{\sqrt{\lambda_n}} \cdot \left(1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i)) \right) + \dots \frac{\tilde{\gamma}_i}{\sqrt{\lambda_n}} \left(\frac{1}{N^2} (\mathbf{1}^T \mathbf{K} \mathbf{1}) \mathbf{1} - \frac{1}{N} \mathbf{K} \mathbf{1} + \mathbf{k}_{\varphi_i} - \frac{1}{N} \mathbf{1} \mathbf{1}^T \mathbf{k}_{\varphi_i} \right)^T u_n - \omega_n,$$

and where $\mathbf{k}_{\Phi_i} = [k_{\Phi_0}(\psi_i, \psi_1), \dots, k_{\Phi_0}(\psi_i, \Phi_N)]^T$.

For the complete derivation, we refer the reader to the Appendix. It is important to note though that due to the closed-form approximation of the pre-image, we are able to use various kernels (e.g., linear and nonlinear PCA) with minimal changes to overall scheme. Next, we discuss how one can evolve the pose parameters.

3.4 Evolving the Pose Parameters

In this section, we discuss the evolution of the pose parameters. Specifically, with a slight abuse of notation, we let $\xi = \lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6\}^T$. Then we are able to compute the

term $\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle$, where λ_i is a translation or rotation parameter as follows:

- For $i = 1, 2, 3$ (i.e., λ_i is a translation parameter) and

$$\mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix},$$

one has

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R} \mathbf{X}_0 + \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \begin{bmatrix} \frac{\partial \lambda_1}{\partial \lambda_i} \\ \frac{\partial \lambda_2}{\partial \lambda_i} \\ \frac{\partial \lambda_3}{\partial \lambda_i} \end{bmatrix}, \mathbf{N} \right\rangle = \left\langle \begin{bmatrix} \delta_{1,i} \\ \delta_{2,i} \\ \delta_{3,i} \end{bmatrix}, \mathbf{N} \right\rangle = N_i, \quad (26)$$

where the Kronecker symbol $\delta_{i,j}$ was used ($\delta_{i,j} = 1$ if $i = j$ and 0 otherwise).

- For $i = 4, 5, 6$ (i.e., λ_i is a rotation parameter) and using the expression of the rotation matrix written in exponential coordinates,

$$\mathbf{R} = \exp \left(\begin{bmatrix} 0 & -\lambda_6 & \lambda_5 \\ \lambda_6 & 0 & -\lambda_4 \\ -\lambda_5 & \lambda_4 & 0 \end{bmatrix} \right),$$

one has

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R} \mathbf{X}_0}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \mathbf{R} \begin{bmatrix} 0 & -\delta_{3,i} & \delta_{2,j} \\ \delta_{3,i} & 0 & -\delta_{1,i} \\ -\delta_{2,i} & \delta_{1,i} & 0 \end{bmatrix} \mathbf{X}_0, \mathbf{N} \right\rangle. \quad (27)$$

We note that one can also expand the rigid-body transformation to a more general affine transformation, and hence provide increased flexibility in the proposed approach.

3.5 Alternative View of Gradient Flow

We briefly revisit the gradient flow in Section 3.2, and present an alternative viewpoint in relationship to the area of shape derivatives [10], [16].

That is, the gradient in (18) involves the computation of the shape derivative, which describes the directions of deformation of the 2D curve (under projection) with respect to the 3D pose parameters and shape coefficients. The gradient is then the dot product of a typical 2D region-based gradient (i.e., $(r_o - r_b) \cdot \hat{\mathbf{n}}$; see, e.g., the Chan and Vese model [8]) with the shape derivative. We note that for each point on the 2D curve, the deformation

direction is compared to the normal, $\langle \frac{\partial \hat{c}}{\partial \xi_i}, \hat{\mathbf{n}} \rangle$, and weights the statistical comparison term, $r_o - r_b$. Then the average over each point of the curve determines the optimal direction of

variation of the finite parameter ξ_i (i.e., the sign of the derivative $\frac{\partial E}{\partial \xi_i}$). Altogether, the link to shape optimization is presented here such that one can view the proposed methodology in the more broader context of shape-based segmentation and pose estimation. Next, we discuss the numerical and implementations details associated with the algorithm.

4 Numerical Details

In (21), the computation of the gradients involves the explicit determination of the occluding curve C . As previously mentioned, one can compute

$$C = \{\mathbf{X} \in S : \langle \mathbf{X}, \mathbf{N} \rangle = 0 \text{ and } \pi(\mathbf{X}) \in \hat{c}\}. \quad (28)$$

However, in practice, this condition is rarely exactly met due to the sampling of a 3D surface. Moreover, if the shape is nonconvex, as it is with most objects seen in this paper, the condition of (28) will yield points that are self-occluded and hence truly not a part of the 3D occluding curve. Thus, an approximation of ε_1 is first made in order to compute an estimate of the visible and nonvisible regions, $\mathcal{V}_{\varepsilon_1}^+ = \{\mathbf{X} \in S : \langle \mathbf{X}, \mathbf{N} \rangle \geq -\varepsilon_1\}$ and $\mathcal{V}_{\varepsilon_1}^- = \{\mathbf{X} \in S : \langle \mathbf{X}, \mathbf{N} \rangle \leq \varepsilon_1\}$, respectively. Relating the visible and nonvisible regions to the occluding curve for both convex and nonconvex shapes, we have

$$C^+ = \{\mathbf{X} \in \mathcal{V}_{\varepsilon_1}^+ : \exists \mathbf{Y} \in \mathcal{V}_{\varepsilon_1}^-, \|\mathbf{X} - \mathbf{Y}\| \leq \varepsilon_2\} \text{ and}$$

$C^- = \{\mathbf{X} \in \mathcal{V}_{\varepsilon_1}^- : \exists \mathbf{Y} \in \mathcal{V}_{\varepsilon_1}^+, \|\mathbf{X} - \mathbf{Y}\| \leq \varepsilon_2\}$, where ε_2 is a chosen (small) parameter. The occluding curve can then be redefined as the union of these two sets or $C = \{\mathbf{X} \in C^+ \cup C^-\}$. We have seen that this procedure mitigates self-occluded points if the proper choice of ε_1 and ε_2 is chosen with regards to the sampling of the 3D surface and camera calibration parameters. In addition, $\hat{c}$ can be obtained by using morphological operations on R , i.e., $\hat{c} = R - \mathcal{E}(R)$, with $\mathcal{E}$ denoting the erosion operation for a chosen kernel [20].

Second, to save computational time, we approximated the term $\sqrt{\frac{\kappa_x \kappa_t}{K}} \simeq 1$ in (21). We note that the approximation was done in order to save computational time and indeed can be a poor approximation if the viewing direction $\mathbf{X}$ and the tangent to the occluding curve are identical. However, we observed that for the sequences and images presented in this paper, the energy decreased and convergence was met.

Last, in performing KPCA, and as with any other statistical learning technique, one must choose the number of modes of variation. In this paper, depending on the data set, the number of modes was chosen to be $l = 4, 5$, or 6 . Moreover, when working with the exponential kernel, the choice of σ is important. Hence, we found that if we choose

$\sigma^2 = \frac{1}{N} \sum_i \min_{j, i \neq j} (\|\psi_i - \psi_j\|^2)$, the algorithm would, for our particular experiments, converge to the desired shape of interest. If one would like to “mix” the shapes in a more linear fashion, σ should be chosen to be a higher value and vice versa.

5 Experiments

We provide segmentation and tracking results to demonstrate the algorithm’s **domain of convergence** and its ability to handle **noise, deformation, occlusions, or clutter**. Moreover, results are given to illustrate the method’s effectiveness in **shape recovery**, which may include **nonlinearities** in one’s training set. Specifically, we generate three 3D training sets corresponding to the number “4” as well as commonly seen teacups and helicopters.¹ This is shown in Fig. 2. Last, because code was not readily available, it should be noted that we do not claim that the proposed method is superior (practically) to existing techniques. Thus, the experiments were performed to highlight the (dis)advantages of a proposed alternative approach for nonrigid segmentation and pose estimation.

5.1 Domain of Convergence

In the first set of experiments, we illustrate the proposed method’s domain of convergence. Although the algorithm is designed for a variational setting such as tracking, in which movements between frames are assumed to be small, it is still interesting to note the domain of convergence.

Specifically, we generated 18 random pairs of the synthetic number 4. This was done by first setting the number of modes $l = 4$, and randomly drawing the set of nonlinear PCA weights from a uniform distribution $\mathcal{U}(2\sigma_i^{\text{deg}}, 2\sigma_i^{\text{deg}})$, where σ_i^{deg} is the i th primary mode of variance. Fig. 3 shows several sample shapes that were generated, as well their initializations and final result, shown in green and yellow, respectively. In particular, using the validation manner of [41], [45], one pair was fixed at the center of the image while the other shape of a specific pair was initialized at positions sampled on a circle whose radius is approximately half of the fixed shape width. Note, the depth was kept constant when initializing the “moving” shape. Then initial rotations of ± 60 degrees in 30 degree increments were tested about each rotational axis while keeping the other two remaining axes fixed. The bottom row of Fig. 3 shows the sampled circle for each of the rotations and, through user visualization, each arrow was marked red for what we considered a “failure” while green represented a “successful” segmentation.

Interestingly, we see that for rotations about the x and y -axes, the algorithm exhibits a larger convergence region than for the z -axis. That is, of the 90 possible initializations, we successfully segmented 76 when rotation was applied to the x -axis, and successfully segmented 66 when rotation was applied to the y -axis. On the other hand, only 48 successful segmentations were observed when rotating about the z -axis. The resulting behavior can be mostly attributed to the initial overlap when the “moving” shape was given. That is, for specific rotations, a change in perspective is seen in the 2D image domain. Combining this with large translations and deviations from the shape results in initializations seen in Fig. 3, the algorithm is driven to an undesirable result. However, we should note that this is a drawback associated with gradient descent as opposed to another optimization method. Although it is beyond the scope of this work, one could strap a particle filter in order to further widen the domain of convergence at the cost of computational complexity.

¹After the models were generated, the shapes were registered with the methodology proposed in [42] so that any variation in shape was not due to pose alignment error.

Nevertheless, the domain of convergence shown here is ideal for segmentation in which one knows the approximate location of the object or tracking scenarios in which deviations in location are generally not seen between consecutive frames. Last, we also note that we performed this test with the linear PCA kernel described earlier and similar results were obtained.

5.2 Segmentation Experiments

In this section, we focus on segmenting a 2D image using a 3D catalog of shapes that may or may not correspond to the object of interest.

5.2.1 Linear PCA Segmentation with Occlusions and Clutter—For the second set of experiments, we provide experimental validation that compares our segmentation method, which is an optimization over a finite set of parameters, with that of the infinite-dimensional GAC technique. The reasoning for such a comparison is as follows: In either methodology, we seek to minimize a cost functional of the general form $E(t) = \int_{\gamma} \Psi(\mathbf{x}, t) d\mathbf{x}$ over a family of curves. For the GAC framework, this family of curves lives only in the image plane when performing 2D image segmentation. Similarly, in the current framework, we are optimizing over a family of 3D “occluding” curves that correspond to 2D “silhouette” curves. That is, we cast the typical infinite-dimensional problem of segmentation as a finite-dimensional optimization problem.

Thus, we benefit from incorporating shape information which results in being able to deal with not only occlusion, but also in cluttered environments where the original assumption of separable statistics does not hold. This is shown on four different examples, as seen in Fig. 4. Fig. 4a illustrates the initialization, while the middle row shows the unsatisfactory result of using an active contour. The final row highlights the results given by the proposed approach with $l = 6$ using the energy proposed in [8]. Although it is not readily apparent, one can alternatively view the examples in Fig. 4 as 3D reconstruction from a single 2D image view exhibiting partial information, which is a fundamental task in computer vision. Details on the level of accuracy of average reconstruction can be found in Sections 5.2.4 and 5.3.1. Also, we note that these experiments can be found in [42].

5.2.2 Linear PCA Segmentation for Shape Recovery—In this section, we shift the focus of our experiments to segmenting shapes that are not in the original training sets using the linear KPCA kernel as done in [42]. We want to specifically highlight an important advantage of learning 3D shapes as opposed to a large catalog of 2D shapes to perform the task of 2D segmentation. This is done by segmenting different shapes arising from different 3D models as well as from altering the pose of a 3D object.

In Fig. 5, we show the segmentation of different views and shapes that are obtained from a person coloring the number “4” on a white-board. The top row highlights the initialization while the bottom row shows the final result. The same experiment is performed for the differing views of two teacups that were not in the original training set (see Figs. 6 and 7). While one may argue that the results are similar to the 2D shape learning approaches, we should note some of the key differences. First, we not only segment the 2D image, but we are able to return the estimated 3D shape and pose from which the 2D object was derived. Moreover, to account for segmentation of objects presented in Figs. 5, 6, and 7 with a 2D shape prior, one would have to learn every possible projection of the 3D object onto the 2D image plane (if no prior knowledge is given about the aspect of the projection). Note, we set $l = 6$ and used the energy proposed in [32].

5.2.3 Nonlinear PCA Segmentation for Shape Recovery—Although it was shown in the previous section that one can segment an object from various views with linear PCA, we now shift our focus on utilizing the nonlinear PCA kernel for a more complex training set such as a rigid class of helicopters. In particular, we found several images and differing viewpoints of helicopters (available online at <http://www.youtube.com>). Thus, the objects again were not within the training set. In particular, Fig. 8 focuses on segmenting a Sikorsky S76 as well as a Bell 212 simulated helicopter from two different views. We should note that while these objects are simulated, the scene is also cluttered with simulated real-life clutter. Despite relatively difficult initializations, we were able to successfully segment two distinct helicopters used. On the other hand, Fig. 9 shows successful segmentations of the same type of helicopters, but in a real-time environment. Note, we set $l = 5$ and used the energy proposed by [32] for these sets of experiments.

5.2.4 Nonlinear versus Linear PCA Segmentation—The next set of experiments demonstrates the robustness of utilizing a nonlinear kernel as opposed to linear PCA in the context of the proposed algorithm. A major drawback in using linear PCA is that one assumes that the training set can be linearly related and decomposed to form a novel learned shape. This assumption is often invalid. For example, when dealing with a more general catalog of shapes in which two sets of objects from different classes are mixed, linear PCA will likely yield an unsatisfactory result. Thus, in order to deal with this problem, one may employ (again) a nonlinear statistical learning technique such as nonlinear PCA. We note that this problem is widely known in the literature and has been solved for various 2D shape-based schemes [31], [12]. However, for the sake of completeness, we demonstrate the increased performance of using nonlinear PCA as compared to linear PCA in the present framework.

We begin by mixing two of the 3D training sets together. In particular, we “corrupt” the helicopter training set, which consists of 12 models, by adding eight of the commonly used teacups. This “new” set of training models is used to perform 2D segmentation of a toy helicopter which is not in our training set for several different viewing angles, as seen in Fig. 10. The first row highlights the initialization, while the second and third rows yield the results of the proposed algorithm when employing linear PCA and nonlinear PCA, respectively. Moreover, because our algorithm can be alternatively viewed as a reconstruction methodology from a single 2D image [40], we also present the corresponding initialization and results for 3D shapes in Fig. 11.

Interestingly, while it may be apparent that nonlinear PCA outperforms linear PCA from the perspective of shape learning, we also found, on average, a faster convergence rate for nonlinear PCA from both a 2D segmentation point of view and for shape reconstruction. This “distance” is computed via ICP measure between two sampled sets of points that lie on the surface of each respective 3D shape [2]. This is shown in Fig. 12 for two of the four viewing angles of our toy helicopter. Specifically, Figs. 12a and 12b show the normalized 2D image segmentation energy of [8], while Figs. 12c and 12d show the normalized L_2 distance between the known 3D shape with that of its reconstructed 3D shape. One can see from the above plots that although the 2D image energy is always being minimized until it reaches steady state, the shape error may gradually increase before it decreases. This is because we are optimizing with the image segmentation energy, and not with the L_2 distance associated with the shape reconstruction error. Nevertheless, nonlinear PCA exhibits a faster convergence rate for both shape reconstruction and image segmentation. Note, we set $l = 5$ for both linear and nonlinear PCA kernels.

5.3 Tracking Experiments

In this section, we focus on tracking an object in a 2D scene using a 3D catalog of shapes. In particular, our initialization for each frame in the video sequence is the pose and shape result from the previous frame. That is, it can be viewed as an extension of the segmentation algorithm previously discussed.

5.3.1 Linear PCA Tracking of Synthetic Deformations, Occlusions, and Noise Using the Number 4

—In the first set of tracking experiments, we demonstrate the algorithm's robustness to noise in the presence of continuous deformation and occlusion.² First, a tracking sequence was generated consisting of 200 frames that were obtained from projecting the number "4" onto the 2D image plane using a simulated camera. Specifically, the variation in the rotation angle was a complete 360 degree cycle, and the model was varied linearly along its z-axis from 170 to 670 spatial units, resulting in a scale in the viewing aspect of the image projection. Also, we translated the model in its x - y axis with a step of 0.5 spatial units for each frame. More importantly, we vary the first three principal modes so that a deformation can be seen. From this basic sequence, three cases of additive Gaussian noise with a standard deviation of $\sigma_n = 25\%$, $\sigma_n = 75\%$, and $\sigma_n = 100\%$ are formed. We also generated an artificial occlusion resembling the intensity of the background with noise $\sigma_n = 25\%$.³

Fig. 13 shows four frames which exhibit typical tracking results from the $\sigma_n = 75\%$ noise case described, as well as the generated occlusion sequence. Here, we have used the region-based energy of [8], and obtained tracking results by varying the first six principal modes, i.e., $l = 6$. Because several 2D-3D pose estimate techniques [33], [18], [28] rely on a correspondence-based scheme to estimate the pose, their methodologies may be sensitive to noise and outliers, as presented in Fig. 13. Because of the robustness of region-based active contours (as compared to local geometric descriptors), the proposed approach yields satisfying visual results such as those of [15]. However, here we have not constrained ourselves to knowing the actual shape of a prespecified number "4". It is straightforward to see (without comparison) that if we were to assume knowledge of a model, then it would not be possible for us to handle the wide range of deformations seen.

Nevertheless, to further ensure the robustness of the algorithm, we provide quantitative tracking results as seen in Table 1. For each image, percent *absolute* errors with respect to the ground truth were computed for both the translation and rotation as follows:

Error = $\frac{\|\mathbf{v}_{\text{measured}} - \mathbf{v}_{\text{truth}}\|}{\|\mathbf{v}_{\text{truth}}\|}$, with $\mathbf{v}$ a translation or quaternion vector. In dealing with the appropriate shape error, we opted to compute the error as follows:

Error = $\frac{\|\mathbf{X}_{\text{measured}} - \mathbf{X}_{\text{truth}}\|}{\max_{j,i \neq j} (\|\mathbf{X}_i - \mathbf{X}_j\|)}$. Note, $\mathbf{X}$ represents the 3D shape of interest, and overload of notation is employed only for this section. Also i and j are simply indices belonging to a training set. At any rate, this error allows us to see how accurate our shape reconstruction is with regard to the maximal error seen in one's training set. From Table 1, we see that the errors for rotation and translation are small even in the presence of noise or severe occlusion. On the other hand, we do see a slight increase in shape error with respect to rotation and translation. This can be mainly attributed to the fact that pose changes may

²We note that a similar, but not exact, experiment can be found in [42]. Also, performing this experiment with nonlinear PCA would yield similar results due to the nature of the given data set.

³Note: We define σ_n to be a percentage of noise generated from a Gaussian distribution to be applied to its corresponding binary image. For example, given the number 4 and the case of $\sigma_n = 25\%$, the image value of $\mathcal{N}(1, .25)$ and $\mathcal{N}(0, .25)$ is chosen for the object and background, respectively.

account for varying shapes as well as the fact that we are only dealing with a single 2D image. However, it still remains small and visually correct.

Moreover, if it is desirable to track a rigid object that is representative of a certain class (e.g., cars, boats, or planes), then one can learn the different 3D models of a class, and thereby relax the constraint of the prior knowledge needed. This will be demonstrated next.

5.3.2 Nonlinear PCA Tracking Occlusion and Clutter of a Toy Helicopter—In many rigid tracking scenarios, only the general class of the object of interest may be known and given. In this experiment, we track a sequence involving a toy helicopter that is not in our 3D helicopter training set. In particular, the sequence presents not only aspect and view changes, but also occlusions from the helicopter rotor blades and a human hand guiding the object. These occlusions are particularly difficult from a statistical viewpoint. That is, the rotor blades are visually black as opposed to a mostly white helicopter, which may cause many segmentation algorithms to exclude this portion of the object. In contrast, the light appearance of the hand relative to most of the background may result in leaks or capture the hand entirely along with the helicopter. Moreover, the clutter in the scene may cause additional problems as previously discussed in Section 5.2.1.

Using the region-based energy of [8], we are able to obtain tracking results by varying the first five principal modes, i.e., $l = 5$. Fig. 14 shows several frames of the video sequence. In particular, the sequence begins with the helicopter on the ground with its rotor blades moving. This inherent movement of the object causes self-occlusions. As the blades begin to slow, the helicopter is moved and placed on top of the book. We see occlusions not only from the blade, but also from the human hand. Nevertheless, the algorithm successfully maintains track. However, we do note that improved tracking performance could be gained by taking into account the temporal coherency of the sequence and using filtering principles such as those proposed in [3], [35].

5.3.3 Nonlinear PCA Tracking of a Sikorsky S-76 and Bell 212 Helicopter in Real Scenarios—Extending the previous section, we now track two helicopters from videos that were obtained online (<http://www.youtube.com>). In particular, using the region-based energy of [32], we were able to obtain tracking results by varying the first five principal modes, i.e., $l = 5$ for both the Sikorsky S76 and Bell 212 sequences as seen in Fig. 15 and Fig. 16, respectively.

In the sequence shown in Fig. 15, the Sikorsky S76 sequence exhibits a large change in the rotation caused by both the helicopter and the camera itself. This resulted in several aspect changes like that in Section 5.3.2. However, the scene in which the helicopter flies in, while simulated, is realistic given the amount of clutter and buildings surrounding the helicopter. Nevertheless, tracking is maintained throughout the sequence. In Fig. 16, we track a Bell 212 helicopter in a real-life sequence. We should note here that although track is effectively maintained, it can be seen that the segmentation is not as successful as that of Fig. 15. That is, one can see that the contour does not properly segment the tail end of the helicopter. This phenomenon can be attributed to the density of the training set. Although we have mentioned that we are able to reduce the computational complexity with regard to 2D shape-based learning, the algorithm can suffer the classic drawbacks when specific details of an object are left out (e.g., a specialized rotor). At any rate, track is still maintained despite these difficulties associated with shape-based learning techniques.

5.4 Performance Analysis

In this section, we report the execution time, iteration count, and other relevant information for several of the experiments performed in this paper in Table 2. It should be noted that our

implementation of the proposed algorithm was done in both MATLAB v7.1 and C/C++ on an Intel Dual Core 2.66 GHz with 4 GB memory. The two pieces of code were integrated via MEX. Also, in relative terms, our implementation of running just the pose portion of the algorithm averaged nearly 100 iterations per second. When including and optimizing over the shape weights, the main reduction in speed was the volume size of the 3D shape as well as the number of models used in the given training set. The 2D image size also impacts computational speed, but is not significant. Moreover, the algorithm can be seen as “unoptimized” code since several steps can be done to effectively increase computing in a sparse manner (e.g., keeping a list of occluding curve points). However, this is beyond the scope of the present paper, and future work is planned to address this issue in detail.

6 Conclusion and Future Work

In this paper, we derive a geometric and variational approach to perform the task of 2D-3D *nonrigid* pose estimation and 2D image segmentation. This can be seen as an extension of the framework presented in [15] and [42], where we assume that we have only a single 3D shape prior. Instead, in the present work, we infer a 3D shape prior from a catalog of 3D rigid shapes, which may represent a general class of an object or possibly a set of deformations that may occur to the model. In addition, to account for nonlinearities in the given training set, we showed that other statistical learning techniques can be employed with minimal changes to the overall framework. As a result, we fully exploit the task of pose estimation and segmentation in a unified framework.

Moreover, because we have shown that the above framework is ideal for tracking rigid objects of a certain class, a direction for future work would be to employ filtering principles such as particle filtering [35], [41]. This is of particular importance since it has been shown that if one exploits the inherent temporal component in video sequences, even more robust tracking results can be obtained. We believe that this should improve the algorithm’s ability to deal with even more challenging occlusions and environments.

Acknowledgments

This work was supported in part by grants from the US National Science Foundation (NSF), the US Air Force Office of Scientific Research, the US Army Research Office, as well as by a grant from the US National Institutes of Health (NIH) (NAC P41 RR-13218) through Brigham and Women’s Hospital. 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>.

Appendix

Gradient Derivation of Exponential Kernel Pre-Image

If we let

$$\mathcal{J}_i = \tilde{\gamma}_i \left(1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i)) \right) \text{ and } \eta_i = \frac{\partial \mathcal{J}_i}{\partial \omega_n},$$

then the general gradient form is

$$\frac{\partial \hat{\psi}}{\partial \omega_n} = \frac{\sum_i^N \eta_i \cdot \psi_i}{\sum_i^N \mathcal{I}_i} - \frac{\left(\sum_i^N \eta_i\right) \left(\sum_i^N \mathcal{I}_i \cdot \psi_i\right)}{\left(\sum_i^N \mathcal{I}_i\right)^2}.$$

Thus, taking the derivative of $\mathcal{J}_i$ w.r.t. ω_i yields

$$\eta_i = \underbrace{\nabla_{\omega_n} \tilde{\gamma}_i}_{(a)} \cdot \left(1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i)) \right) - \frac{1}{2} \tilde{\gamma}_i \cdot \underbrace{\nabla_{\omega_n} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i))}_{(b)}.$$

Now taking the derivative of (a) above by using (11), we get

$$\tilde{\gamma}_i = \sum_n^l \frac{\omega_n u_{ni}}{\sqrt{\lambda_n}} + \frac{1}{N} \left(1 - \sum_j^N \sum_n^l \frac{\omega_n u_{nj}}{\sqrt{\lambda_n}} \right) \nabla_{\omega_n} \tilde{\gamma}_i = \frac{u_{ni}}{\sqrt{\lambda_n}} - \frac{1}{N} \sum_j^N \frac{u_{nj}}{\sqrt{\lambda_n}}.$$

Next, we compute the gradient of part (b). Recall that

$$d_{\mathcal{H}}^2(P^l\varphi(\psi), \varphi(\psi_i)) = \|P^l\varphi(\psi)\|^2 + \|\varphi(\psi_i)\|^2 - 2\langle P^l\varphi(\psi), \varphi(\psi_i) \rangle.$$

From this, we can begin to express each term’s dependence on the shape weight ω_n . The first term can be expressed as

$$\begin{aligned} & \|P^l \varphi(\psi)\|^2 \\ &= \left(\sum_n^l \omega_n V_n + \bar{\varphi} \right)^T \left(\sum_n^l \omega_n V_n \right. \\ &\quad \left. + \bar{\varphi} \right) = \left(\sum_n^l \sum_n^l \omega_n V_n^T V_n \omega_n \right) \\ &\quad + \bar{\varphi}^T \bar{\varphi} + \left(2 \bar{\varphi}^T \sum_n^l \omega_n V_n \right) = \left(\sum_n^l \omega_n^2 \right) \\ &\quad + \bar{\varphi}^T \bar{\varphi} + \left(2 \bar{\varphi}^T \sum_n^l \omega_n V_n \right) \\ &= \left(\sum_n^l \omega_n^2 \right. \\ &\quad \left. + 2 \frac{\omega_n}{\sqrt{\lambda_n}} \bar{\varphi}^T \tilde{\phi} u_n \right) + \bar{\varphi}^T \bar{\varphi}. \end{aligned}$$

Similarly, the third term can be expressed as

$$\begin{aligned}
& \langle P^l \varphi(\psi), \varphi(\psi_i) \rangle \\
&= \left(\sum_n^l \omega_n V_n + \bar{\varphi} \right)^T \varphi(\psi_i) \\
&= \left(\sum_n^l (\omega_n V_n)^T \varphi(\psi_i) \right) \\
&+ \bar{\varphi}^T \varphi(\psi_i) \\
&= \left(\sum_n^l \omega_n \left(\frac{1}{\sqrt{\lambda_n}} \tilde{\phi} u_n \right)^T \varphi(\psi_i) \right) \\
&+ \bar{\varphi}^T \varphi(\psi_i) \\
&= \left(\sum_n^l \frac{\omega_n}{\sqrt{\lambda_n}} (u_n^T \tilde{\phi}^T \varphi(\psi_i)) \right) \\
&+ \bar{\varphi}^T \varphi(\psi_i).
\end{aligned}$$

Combining these terms then gives us

$$d_{\mathcal{H}}^2 = \left(\sum_n^l \omega_n^2 + 2 \frac{\omega_n}{\sqrt{\lambda_n}} (\tilde{\phi}^T \bar{\varphi})^T u_n - 2 \frac{\omega_n}{\sqrt{\lambda_n}} (\tilde{\phi}^T \varphi(\psi_i))^T u_n \right) - 2 \bar{\varphi}^T \varphi(\psi_i) + \bar{\varphi}^T \bar{\varphi} + \|\varphi(\psi_i)\|^2.$$

Now, taking the derivative w.r.t. ω_n , we arrive at the following:

$$\begin{aligned}
& \frac{d(d_{\mathcal{H}}^2)}{d\omega_n} \\
&= 2 \left(\omega_n \right. \\
&\quad \left. + \frac{1}{\sqrt{\lambda_n}} \cdot (\tilde{\phi}^T \bar{\varphi} - \tilde{\phi}^T \varphi(\psi_i))^T u_n \right) \\
&= 2 \left(\omega_n \right. \\
&\quad \left. + \frac{1}{\sqrt{\lambda_n}} \cdot ((\phi - \bar{\phi})^T \bar{\varphi} - (\phi - \bar{\phi})^T \varphi(\psi_i))^T u_n \right) = 2 \left(\omega_n \right. \\
&\quad \left. + \frac{1}{\sqrt{\lambda_n}} \cdot (\phi^T \bar{\varphi} - \bar{\phi}^T \bar{\varphi} - \phi^T \varphi(\psi_i) + \bar{\phi}^T \varphi(\psi_i))^T u_n \right) \\
&= 2 \left(\frac{1}{\sqrt{\lambda_n}} \left(\frac{1}{N} \mathbf{K} \mathbf{I} - \frac{1}{N^2} (\mathbf{I}^T \mathbf{K} \mathbf{I}) \mathbf{I} - \mathbf{k}_{\varphi_i} + \frac{1}{N} \mathbf{I} \mathbf{I}^T \mathbf{k}_{\varphi_i} \right)^T u_n + \omega_n \right),
\end{aligned}$$

where

$$\mathbf{k}_{\varphi_i} = [k_{\varphi_{\sigma}}(\psi_i, \psi_1), \dots, k_{\varphi_{\sigma}}(\psi_i, \psi_N)]^T \quad \phi = [\varphi(\psi_1), \varphi(\psi_2), \dots, \varphi(\psi_N)] \text{ and } \bar{\phi} = \phi - \tilde{\phi}.$$

Putting all of this together, we arrive at the desired gradient:

$$\eta_i = \frac{u_{ni}}{\sqrt{\lambda_n}} - \frac{1}{N} \sum_j^N \frac{u_{nj}}{\sqrt{\lambda_n}} \cdot \left(1 - \frac{1}{2} d_{\mathcal{H}}^2(P^l \varphi(\psi), \varphi(\psi_i)) \right) + \dots \frac{\tilde{\gamma}_i}{\sqrt{\lambda_n}} \left(\frac{1}{N^2} (\mathbf{I}^T \mathbf{K} \mathbf{I}) \mathbf{I} - \frac{1}{N} \mathbf{K} \mathbf{I} + \mathbf{k}_{\varphi_i} - \frac{1}{N} \mathbf{I} \mathbf{I}^T \mathbf{k}_{\varphi_i} \right)^T u_n - \omega_n.$$

We now have all of the components necessary to compute the overall shape gradient used for this work.

References

1. Balan A, Sigal L, Black M, Davis J, Haussecker H. Detailed Human Shape and Pose from Images. Proc. IEEE Conf. Computer Vision and Pattern Recognition. 2007
2. Besl PJ, McKay ND. A Method for Registration of 3D Shapes. IEEE Trans. Pattern Analysis and Machine Intelligence. 1992 Feb.vol. 14(no. 2):239–256.
3. Blake, A.; Isard, M., editors. Active Contours. Springer; 1998.
4. Blanz V, Vetter T. A Morphable Model for the Synthesis of 3D Faces. Proc. ACM SIGGRAPH. 1999:187–194.
5. Bowden R, Mitchell T, Sarhadi M. Reconstructing 3D Pose and Motion from a Single Camera View. Proc. British Machine Vision Conf. 1998; vol. 2:904–913.
6. Bray M, Kohli P, Torr P. Posecut: Simultaneous Segmentation and 3D Pose Estimation of Humans Using Dynamic Graph-Cuts. Proc. European Conf. Computer Vision. 2006:642–655.
7. Caselles V, Kimmel R, Sapiro G. Geodesic Active Contours. Proc. Int'l J. Computer Vision. 1997; vol. 22:61–79.
8. Chan T, Vese L. Active Contours without Edges. IEEE Trans. Image Processing. 2001 Feb.vol. 10(no. 2):266–277.
9. Charpiat G, Faugeras O, Keriven R. Approximations of Shape Metrics and Application to Shape Warping and Empirical Shape Statistics. Foundations of Computational Math. 2005; vol. 5(no. 1): 1–58.
10. Charpiat G, Faugeras O, Keriven R. Shape Statistics for Image Segmentation with Prior. Proc. IEEE Conf. Computer Vision and Pattern Recognition. 2007:1–6.
11. Charpiat G, Maurel P, Pons J, Keriven R, Faugeras O. Generalized Gradients: Priors on Minimization Flows. Int'l J. Computer Vision. 2007; vol. 73(no. 3):325–344.
12. Cremers D, Kohlberger T, Schnoerr C. Shape Statistics in Kernel Space for Variational Image Segmentation. Pattern Recognition. 2003; vol. 36(no. 9):1929–1943.
13. Dambreville, S. doctoral dissertation, Georgia Inst. of Technology. 2008. Statistical and Geometric Methods for Shape-Driven Segmentation and Tracking.
14. Dambreville S, Rathi Y, Tannenbaum A. A Framework for Image Segmentation Using Shape Models and Kernel Space Shape Priors. IEEE Trans. Pattern Analysis and Machine Intelligence. 2008 Aug.vol. 30(no. 8):1385–1399.
15. Dambreville S, Sandhu R, Yezzi A, Tannenbaum A. Robust 3D Pose Estimation and Efficient 2D Region-Based Segmentation from a 3D Shape Prior. Proc. European Conf. Computer Vision. 2008
16. Debreuve E, Gastaud M, Barlaud M, Aubert G. Using the Shape Gradient for Active Contour Segmentation: From the Continuous to the Discrete Formulation. J. Math. Imaging and Vision. 2007; vol. 28(no. 1):47–66.
17. Deutscher J, Reid I. Articulated Body Motion Capture by Stochastic Search. Int'l J. Computer Vision. 2004; vol. 61(no. 2):185–205.
18. Dhome M, Richetin M, Lapreste JT. Determination of the Attitude of 3D Objects from a Single Perspective View. IEEE Trans. Pattern Analysis and Machine Intelligence. 1989 Dec.vol. 11(no. 12):1265–1278.
19. Drummond T, Cipolla R. Real-Time Tracking of Multiple Articulated Structures in Multiple Views. Proc. European Conf. Computer Vision. 2000
20. Gonzalez, RC.; Woods, RE. Digital Image Processing. Addison-Wesley Longman; 2001.
21. Han F, Zhu S. Bayesian Reconstruction of 3D Shapes and Scenes from a Single Image. Proc. Int'l Workshop High Level Knowledge in 3D Modeling and Motion. 2003; vol. 2
22. Hartley, R.; Zisserman, A. Multiple View Geometry in Computer Vision. Cambridge Univ. Press; 2000.

23. Kichenassamy S, Kumar S, Olver P, Tannenbaum A, Yezzi A. Conformal Curvature Flow: From Phase Transitions to Active Vision. *Archives for Rational Mechanics and Analysis*. 1996; vol. 134:275–301.
24. Kohli P, Rihan J, Bray M, Torr P. Simultaneous Segmentation and 3D Pose Estimation of Humans Using Dynamic Graph Cuts. *Int'l J. Computer Vision*. 2008; vol. 79(no. 3):285–298.
25. Kwok J, Tsang I. The Pre-Image Problem in Kernel Methods. *IEEE Trans. Neural Networks*. 2004 Nov.vol. 15(no. 6):1517–1525.
26. Leventon M, Grimson E, Faugeras O. Statistical Shape Influence in Geodesic Active Contours. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2000:1316–1324.
27. Ma, Y.; Soatto, S.; Kosecka, J.; Sastry, S. *An Invitation to 3D Vision*. Springer; 2001.
28. Marchand E, Bouthemy P, Chaumette F. A 2D-3D Model-Based Approach to Real-Time Visual Tracking. *Image and Vision Computing*. 2001 Nov.vol. 19(no. 13):941–955.
29. Mercer J. Functions of Positive and Negative Type and Their Connection with the Theory of Integral Equations. *Philosophical Trans. Royal Soc. London*. 1909; vol. 209:415–446.
30. Michailovich O, Rathi Y, Tannenbaum A. Image Segmentation Using Active Contours Driven by the Bhattacharyya Gradient Flow. *IEEE Trans. Image Processing*. 2007 Nov.vol. 16:2787–2801.
31. Mika, S.; Scholkopf, B.; Smola, AJ.; Muller, KR.; Scholz, M.; Ratsch, G. *Advances in Neural Information Processing Systems*. Vol. vol. 11. MIT Press; 1999. Kernel PCA and De-Noising in Feature Spaces.
32. Paragios N, Deriche R. Geodesic Active Regions: A New Paradigm to Deal with Frame Partition Problems in Computer Vision. *J. Visual Comm. and Image Representation*. 2002; vol. 13:249–268.
33. Quan L, Lan ZD. Linear N-Point Camera Pose Determination. *IEEE Trans. Pattern Analysis and Machine Intelligence*. 1999 Aug.vol. 21(no. 8):774–780.
34. Rathi Y, Dambreville S, Tannenbaum A. Statistical Shape Analysis Using Kernel PCA. *Proc. SPIE*. 2006:425–432.
35. Rathi Y, Vaswani N, Tannenbaum A, Yezzi T. Particle Filtering for Geometric Active Contours with Application to Tracking Moving and Deforming Objects. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2005
36. Riklin-Raviv T, Kiryati N, Sochen N. Prior-Based Segmentation by Projective Registration and Level Sets. *Proc. Int'l Conf. Computer Vision*. 2005:204–211.
37. Rosenhahn B, Brox T, Weickert J. Three-Dimensional Shape Knowledge for Joint Image Segmentation and Pose Tracking. *Int'l J. Computer Vision*. 2007; vol. 73(no. 3):243–262.
38. Rosenhahn B, Kersting U, Powel K, Seidel HP. Cloth X-Ray: Mocap of People Wearing Textiles. *Proc. Ann. Symp. German Assoc. for Pattern Recognition*. 2006:495–504.
39. Rosenhahn B, Perwass C, Sommer G. Pose Estimation of Free-Form Contours. *Int'l J. Computer Vision*. 2005; vol. 62(no. 3):267–289.
40. Rother D, Sapiro G. Seeing 3D Objects in a Single 2D Image. *Proc. Int'l Conf. Computer Vision*. 2009
41. Sandhu R, Dambreville S, Tannenbaum A. Particle Filtering for Registration of 2D and 3D Point Sets with Stochastic Dynamics. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2009
42. Sandhu R, Dambreville S, Yezzi A, Tannenbaum A. Nonrigid 2D-3D Pose Estimation and 2D Image Segmentation. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2009
43. Schmaltz C, Rosenhahn B, Brox T, Cremers D, Weickert J, Wietzke L, Sommer G. Region-Based Pose Tracking. *Pattern Recognition and Image Analysis*. 2007:56–63.
44. Snow D, Viola P, Zabih R. Exact Voxel Occupancy with Graph Cuts. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2000; vol. 1:345–352.
45. Sofka M, Yang G, Stewart C. Simultaneous Covariance Driven Correspondence (CDC) and Transformation Estimation in the Expectation Maximization Framework. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2007:1–8.
46. Taron M, Paragios N, Jolly M. Registration with Uncertainties and Statistical Modeling of Shapes with Variable Metric Kernels. *IEEE Trans. Pattern Analysis and Machine Intelligence*. 2009 Jan.vol. 31(no. 1):99–113.

47. Tsai A, Yezzi T, Wells W, Tempany C, Tucker D, Fan A, Grimson E, Willsky A. A Shape-Based Approach to the Segmentation of Medical Imagery Using Level Sets. *IEEE Trans. Image Processing*. 2003 Feb.vol. 22(no. 2):137–153.
48. Unal G, Yezzi A, Soatto S, Slabaugh G. A Variational Approach to Problems in Calibration of Multiple Cameras. *IEEE Trans. Pattern Analysis and Machine Intelligence*. 2007 Aug.vol. 29(no. 8):1322–1338.
49. Yezzi A, Soatto S. Stereoscopic Segmentation. *Int'l J. Computer Vision*. 2003; vol. 53(no. 3):31–43.
50. Yezzi A, Soatto S. Structure from Motion for Scenes without Features. *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*. 2003 Jun.vol. 1:171–178.

Biographies

Romeil Sandhu received the bachelor's and master's degrees in electrical engineering from the Georgia Institute of Technology in 2006 and 2009, respectively, and received the PhD degree under the supervision of Allen Tannenbaum in 2010. His research interests include numerical optimization, segmentation, and registration, with applications to visual tracking and medical imaging. He is a member of the IEEE.

Samuel Dambreville received the master of science and PhD degrees in electrical engineering from the Georgia Institute of Technology in 2003 and 2008, respectively, and a master's degree in business administration from the same institute in 2006. His research interests include shape-driven techniques for segmentation. He is a member of the IEEE.

Anthony Yezzi is currently a faculty member in the School of Electrical and Computer at the Georgia Institute of Technology. His research interests include the usage of curve/surface evolution theory from differential geometry and partial differential equations in computer vision. He is a member of the IEEE.

Allen Tannenbaum is currently a faculty member in the Schools of Electrical and Computer and Biomedical Engineering, Georgia Institute of Technology. He is also a faculty member in the School of Electrical Engineering, Technion-Israel Institute of Technology. His research interests include systems and control, computer vision, and image processing. He is a fellow of the IEEE.

Fig. 1. Our nonrigid approach to 2D image segmentation and 3D pose estimation through the use of multiple 3D shapes.

Fig. 2.

Three 3D training sets used in this work. (a) Different 3D models of commonly used teacups (6 of the 12 models used for the training are shown). (b) Different 3D models of the number 4 (6 of the 16 models used for the training are shown). (c) Different 3D models of commonly seen helicopters (6 of the 12 models used for the training are shown).

Fig. 3.

Domain of convergence. (a) Sample visual results (top row) with convergence results (bottom row) when rotation is applied to the x-axis. (b) Sample visual results (top row) with convergence results (bottom row) when rotation is applied to the y-axis. (c) Sample visual results (top row) with convergence results (bottom row) when rotation is applied to the z-axis. Note: Arrows are positioned at varying 30 degree increments and red arrows and green arrows denote “failure” and “success,” respectively.

Fig. 4. Linear PCA segmentation with occlusion and clutter. (a) Initialization. (b) Unsatisfactory results obtained from using an active contour. (c) Results obtained from the proposed approach.

Fig. 5.

Linear PCA segmentation for shape recovery: segmentation of different views and shapes of the number “4” which are not present in the training set. (a) Initialization. (b) Final results obtained for the running proposed.

Fig. 6.

Linear PCA segmentation for shape recovery: segmentation of different views and shapes of a beige teacup which is not present in the training set. (a) Initialization. (b) Final results obtained for the running proposed.

Fig. 7.

Linear PCA segmentation for shape recovery: segmentation of different views and shapes of a black teacup which is not present in the training set. (a) Initialization. (b) Final results obtained for the running proposed.

Fig. 8. Nonlinear PCA segmentation for shape recovery: segmentation of different views and shapes of simulated helicopters which are not present in the training set. (a) Initialization. (b) Final results obtained for the running proposed.

Fig. 9.

Nonlinear PCA segmentation for shape recovery: segmentation of different views and shapes of real helicopters which are not present in the training set. (a) Initialization. (b) Final results obtained for the running proposed.

Fig. 10.

Nonlinear versus linear PCA segmentation: image segmentation results when two different classes of shapes (teacups and helicopters) are “mixed” to generate a new training set. (a) Initialization. (b) Unsatisfactory results obtained using the proposed algorithm with PCA. (c) Satisfactory results obtained using the proposed algorithm with KPCA.

Fig. 11.

Nonlinear versus linear PCA segmentation: 3D shape reconstruction results when two different classes of shapes (teacups and helicopters) are “mixed” to generate a new training set. (a) Initialization. (b) Unsatisfactory results obtained using the proposed algorithm with PCA. (c) Satisfactory results obtained using the proposed algorithm with KPCA.

Fig. 12.

Nonlinear versus linear PCA segmentation: PCA and KPCA comparison convergence plots of image and shape energy. (a) and (b) Segmentation energy convergence plot of two different examples of segmenting a toy helicopter. (c) and (d) Corresponding shape energy convergence plots. Note: Black color denotes the KPCA result while red color denotes the PCA result.

Fig. 13.

Linear PCA tracking of synthetic deformations, occlusions, and noise using the number 4. Visual tracking results for the sequence involving the number 4. (a) Tracked sequence with Gaussian noise of standard deviation $\sigma = 75\%$. (b) Tracked sequence for $\sigma = 25\%$ with severe occlusion.

Fig. 14.
Tracking with occlusion and clutter (toy helicopter). Several tracking frames are shown.

Fig. 15. Tracking a Sikorsky S-76 through a simulated environment. Several tracking frames are shown.

Fig. 16.

Tracking a Bell 212 through a real environment. Several tracking frames are shown.

TABLE 1

Quantitative Tracking Results Demonstrating Robustness to Deformation (and Noise)

Noise Level	mean error. (in %)	std. dev. error (in %)	max error (in %)
25%	T: 1.39 R: 1.08 ω : 3.46	T: .44 R: 0.36 ω : 0.84	T: 2.63 R: 2.15 ω : 5.87
75%	T: 1.67 R: 1.46 ω : 4.20	T: .61 R: 0.48 ω : 0.88	T: 3.41 R: 2.95 ω : 6.65
100%	T: 1.85 R: 1.41 ω : 4.24	T: 0.69 R: 0.39 ω : 0.97	T: 3.62 R: 2.14 ω : 7.02
25% (Occ.)	T: 2.51 R: 2.75 ω : 5.27	T: 0.85 R: 1.38 ω : 0.91	T: 4.94 R: 6.93 ω : 7.89

Mean, Standard Deviation, and Max Errors with respect to translation, rotation, and shape recovery are reported for several levels of noise.

TABLE 2
Performance Analysis of the Proposed Nonrigid 2D-3D Pose Estimation and Segmentation Algorithm

Experiment	2D Image Size	3D Model Size	Number of Modes Used	Number of Iterations	Time (in Sec)
Linear PCA Occlusion and Clutter (Black Teacup)	[640×480×3]	[128×128×128]	6	1000	18.14
Linear PCA Shape Recovery (Number 4)	[640×480×3]	[100×100×50]	6	500	7.31
Linear PCA Shape Recovery (Beige Teacup)	[640×480×3]	[128×128×128]	6	1000	18.29
Nonlinear Training Set (Toy Helicopter)	[640×480×3]	[128×128×128]	4	1200	22.64
Tracking Deformation and Noise (Number 4)	[300×300×1] × 200	[100×100×50]	6	30/Frame	155
Tracking Occlusion and Clutter (Toy Helicopter)	[512×288×3] × 1319	[128×128×128]	5	40/Frame	1022

Note that image sizes of $[N \times K]$ and $[N \times K \times 3]$ represent grayscale and color images.