2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

# A Learning Algorithm for Visual Pose Estimation of Continuum Robots

Austin Reiter, Roger E. Goldman, Andrea Bajo, Konstantinos Iliopoulos, Nabil Simaan, and Peter K. Allen

***Abstract*— Continuum robots offer signicant advantages for surgical intervention due to their down-scalability, dexterity, and structural flexibility. While structural compliance offers a passive way to guard against trauma, it necessitates robust methods for online estimation of the robot configuration in order to enable precise position and manipulation control. In this paper, we address the pose estimation problem by applying a novel mapping of the robot configuration to a feature descriptor space using stereo vision. We generate a mapping of known features through a supervised learning algorithm that relates the feature descriptor to known ground truth. Features are represented in a reduced sub-space, which we call *eigen-features*. The descriptor provides some robustness to occlusions, which are inherent to surgical environments, and the methodology that we describe can be applied to multi-segment continuum robots for closed-loop control. Experimental validation on a single-segment continuum robot demonstrates the robustness and efficacy of the algorithm for configuration estimation. Results show that the errors are in the range of $1^\circ$.**

## I. Introduction

Medical robotics and computer-assisted surgery have become integral to the delivery of care due to the improved performance granted by introducing computer processing power and related control hardware into the workflow of surgical intervention. Robotic technology has the potential to improve performance over manual intervention [1] by offering improved precision, increased dexterity, decreased instrument volumes, motion scaling and the integration of sensory information. The introduction of advanced instrumentation has allowed significant progress by the surgical community toward new surgical paradigms that are less invasive than conventional Minimally Invasive Surgery (MIS). These techniques are Single Port Access Surgery (SPAS) [2], Laparo-Endoscopic Single-Site surgery [3], and Natural Orifice Transluminal Endoscopic Surgery (NOTES) [4].

Continuum robots have seen increased interest by the research community because they are dexterous, innately compliant, and down-scalable as surgical effectors for MIS [5], [6], SPAS [7], and NOTES. These robots differ from traditional industrial articulated robots in that trajectories are generated by deformation of internal structures of the robotic mechanism as opposed to the relative motion of individual rigid links. In the past two decades several different designs and actuation modalities have been proposed [8]–[14]. However, they all suffer from lack of accuracy due to friction, extension and torsion of their actuation lines, shape discrepancy from nominal kinematics, and actuation coupling between segments.

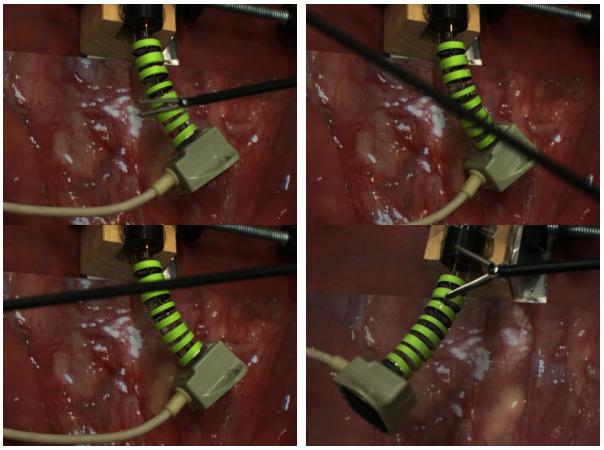


Fig. 1. A stereo camera system views a single segment of a continuum snake arm in order to learn a mapping of the configuration angles from visual feature descriptors. The algorithm was tested in the presence of a realistic occluder in the form of a laparoscopic tool being manipulated between the snake and the camera.

Researchers have tried to overcome these problems with off-line model-based methods [6], [15]–[17] or off-line vision-based approaches [10], [18]–[22]. Camarillo et. al. [21] used a voxel-carving strategy to extract the position of a flexible manipulator using 3 orthogonal cameras spread about the environment. Hannan and Walker [18] extracted individual vertebrae along a snake arm to fit successive circles to determine the curvature by analyzing the change in length of the segment due to curving. In [10], Gravagne and Walker examined different stiffness/compliance ellipsoids in order to explore compliance characteristics. Croom et. al. [22] used Self-Organizing Maps in a stereo vision framework to detect the shape of a continuum robot without the use of fiducials for positional accuracy. More recently, a tiered real-time controller that uses both extrinsic and intrinsic sensory information for improved performance of multi-segment continuum robots has been proposed [23]. The higher tier of this controller uses configuration space feedback while the lower tier uses joint space feedback and a feed-forward term obtained with actuation compensation

A. Reiter and P. K. Allen are with the Dept. of Computer Science, Columbia University, New York, NY 10027, USA areiter, allen@cs.columbia.edu

R. E. Goldman is with the Dept. of Biomedical Engineering, Columbia University, New York, NY 10027, USA reg2117@columbia.edu

A. Bajo and N. Simaan are with the Dept. of Mechanical Engineering, Vanderbilt University, Nashville, TN 37212 andrea.bajo, nabil.simaan@vanderbilt.edu

K. Iliopoulos is with the Dept. of Computer Engineering, Columbia University, New York, NY 10027, USA ki2176@columbia.edu

This work was funded by NIH grant 5R21EB007779-02

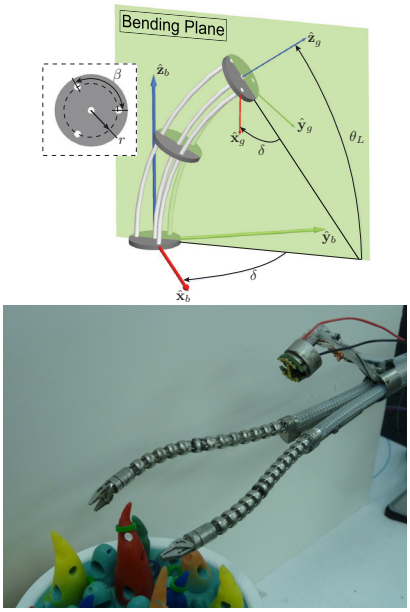



Fig. 2. [**Top**] Structure and kinematic nomenclature for a single-segment continuum robot. [**Bottom**] IREP with two continuum arms constructed along with the stereo camera actuated unit.

techniques. This work requires extrinsic feedback on the segment configuration from a magnetic tracker or from a vision system, such as the proposed algorithm of this work.

This paper proposes a method to estimate the configuration of a single-segment continuum robot using a visual feature descriptor that is extracted from a stereo camera system and mapped to the robot's configuration angles. Our image segmentation and descriptor extraction methods are shown to be robust to partial occlusions. We present these results by manipulating a standard laparoscopic tool in the viewing frustrum, providing different levels of partial occlusions in a realistic fashion. Although the algorithm extends to any continuum robot design, we have in mind the IREP surgical robot [24], shown on the bottom of Fig. 2. Our method uses training samples to interpolate a manifold, which is parameterized by the configuration angles of the continuum segment. This compact representation of the appearance of the robot's configuration allows us to estimate unknown configurations by extracting the feature descriptor and indexing into the manifold to determine the best angles which may have produced that descriptor. The proposed algorithm uses a feature descriptor to provide the sensitivity needed to accurately capture small changes in the configuration of a continuum robot. Although the segment bends in a circular shape, camera perspective effects make measuring the image of the circle (viewed as an ellipse) difficult when the movement is out-of-plane, and the descriptor encodes this information in an alternative and robust fashion. The algorithm relies on the assumption that consecutive configurations are strongly correlated and nearby in the feature descriptor's space. We tested on robot movements in all 3-dimensions and are able to recover configuration angles in the range of $1^\circ$ of accuracy.

## II. Modeling of the Continuum Segment

Several designs of continuum robots that bend in a circular shape have been proposed [25]. This section briefly presents the kinematics of the particular design [11] used to validate the work proposed in this paper. The multi-backbone single-segment robot shown on the top of Fig. 2 is constructed of one centrally located passive primary backbone, and three radially actuated secondary backbones with pitch radius $r$ and separation angle $\beta$. By controlling the lengths of the secondary backbones, the segment can be moved throughout the workspace defined by the kinematics. The pose of the end disk of the continuum robot can be completely described by the generalized coordinates, termed configuration space, by

$$\psi = [\theta_L, \delta]^{\mathrm{T}} \tag{1}$$

where $\theta_L$ and $\delta$ define respectively the angle tangent to the central backbone at the end disk, and the plane in which the segment bends. The orientation of the end disk is given by the following sequence of rotations:

$$\mathbf{R} = \mathbf{R_z}\mathbf{R_y}\mathbf{R_z^T} \tag{2}$$

where $\mathbf{R_z} = \mathrm{e}^{-\delta[\mathbf{e_3}\times]}$, $\mathbf{R_y} = \mathrm{e}^{(\theta_0-\theta_L)[\mathbf{e_2}\times]}$, denote the exponential forms for these rotations, $\mathbf{e}_j$ denote the canonical basis unit vectors for $\mathbb{R}^3$, $[\mathbf{n}\times]$ designates the skew-symmetric cross product matrix of vector $\mathbf{n}$, and $\theta_0 = \pi/2$. The configuration variables $\theta_L$ and $\delta$ can be obtained from (2) as:

$$\theta_L = \theta_0 - \mathrm{atan2}\left(\sqrt{R_{13}^2 + R_{23}^2}, R_{33}\right) \tag{3}$$

$$\delta = -\mathrm{atan2}\left(R_{23}, R_{13}\right) \tag{4}$$

where $R_{ij}$ are the entries of rotation matrix $\mathbf{R}$[1].

## III. Learning Method Overview

The problem of estimating the pose of an object by learning the appearance has been studied previously. Murase and Nayar [26] collected a set of images by sampling the workspace of an object's configuration and compressing to a low-dimensional eigen-subspace. This builds a continuous appearance manifold for which queries can be interpolated for unknown poses. This particular approach is extremely sensitive because the appearance is represented at the pixel level, and spatial variations within the image may present a challenge. Occlusions present an issue as well, which is extremely common in surgical environments. This parametric eigenspace representation is the main motivation for the contributions of our ideas, however our approach is novel in that the manifold is constructed using feature descriptors rather than the images themselves.

Vision offers a low-cost and safe solution to physical measurements in a surgical environment. Because the configuration of a single-segment continuum robot can be completely described by configuration angles, it would be a powerful argument to do so accurately with cameras alone. Fig. 3 shows the full algorithm flow of our method

[1]We use the atan2 notation such that: $\theta = \mathrm{atan2}(\sin(\theta), \cos(\theta))$.

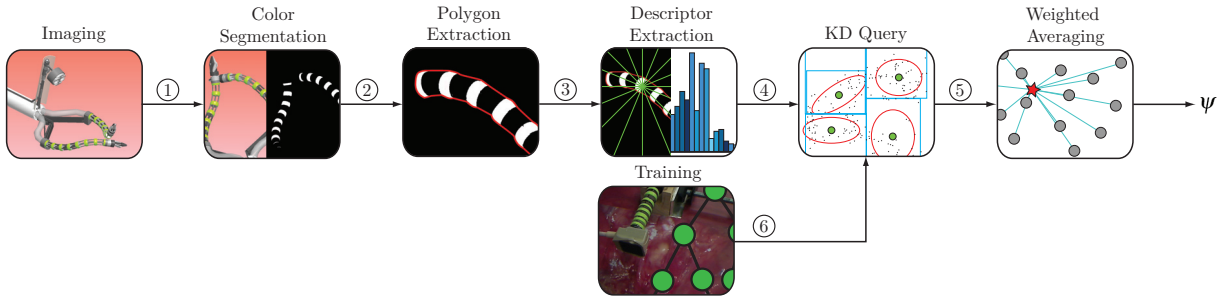



Fig. 3. Algorithm flow for the visual pose estimation algorithm. ① Stereo Images, ② Segmented Components, ③ Polygonal Points, ④ *eigen-features*, ⑤ $b$ closest interpolated poses, ⑥ Initial training set $\mathbf{A}$. Further details in text.

for describing the configuration of a bendable arm using vision alone. The method relies on *learning* mappings of configurations to feature descriptors. The descriptors must be stable enough so that nearby points in feature space represent similar configurations. By discretely sampling a continuous space of configurations, we can interpolate a continuous feature descriptor manifold, which is parameterized by the configuration angles, and then accurately and efficiently estimate the unknown configuration of the arm by indexing into the manifold and matching to the nearest neighbors of the descriptors in the manifold and performing a weighted average from the known configuration angles nearby.

In addition to learning, another positive aspect to our algorithm lies in its ability to estimate the configuration using only partial data due to visual occlusions. Often during surgery, overlapping tools or excess liquids may temporarily occlude parts of the continuum arm. Our algorithm robustly and successfully persists despite partial occlusions.

Note that several of these individual segments are often combined together to form a full robotic snake arm for maximal dexterity, and this algorithm proposes a solution to estimate the configuration of each single segment at a time, which can then be combined in the end to capture the full pose of the robot arm. We first need to train our system to map known poses to feature descriptors.

### A. Ground Truth Collection

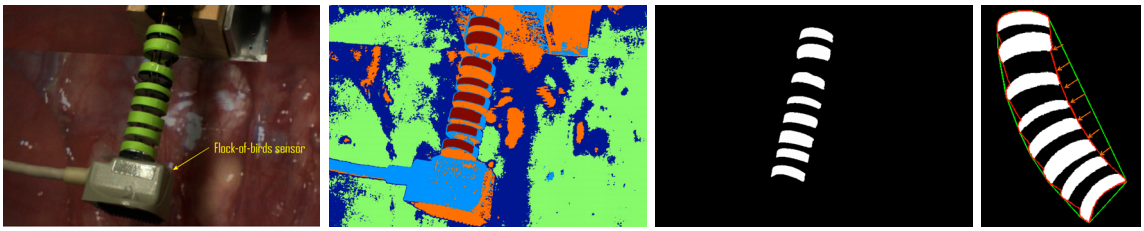
We fixed an *Ascension Technology Flock-of-Birds* 3D tracking device to the distal end of the snake segment (see Fig. 4(a)). This is a magnetic tracker capable of providing 3D positions and orientations at approximately $144Hz$. Positional and orientation accuracy are 1.8mm and 0.5° RMS, respectively. The sensor provides a 3D position $\mathbf{p}$ and a 3D orthonormal rotation matrix $\mathbf{R}$, which we convert to $\psi$ angles according to (3) and (4). This ground truth is sufficient to describe the moving configurations of the snake segment.

### B. Snake Segmentation

*1) Color Segmentation:* A snake arm with 8 vertebrae is color coded with lime markers. The length of the segment is 61mm, and each vertebrae disk has a height of 3.5mm with a diameter of 14mm. This color was chosen to stand out from typical medical imagery, which is more red by nature. It is not unrealistic to place these kinds of fiducial markers on surgical robots to simplify these types of detection tasks [27]. We use a single frame to perform a *k-means clustering* of colors. We choose the CIELAB color space, which consists of a luminosity layer $\mathbf{L}$, chromanicity layer-$a$ (which indicates where colors fall along the red-green axis), and chromanicity layer-$b$ (which indicates where colors fall along the blue-yellow axis). The color information is actually in the $a$ and $b$ layers, and so the clustering is performed using only those 2 components. We found this to be a more robust representation to cluster color pixels than the typical RGB or HSV color spaces which are commonly used.

We make the assumption that in surgical imagery, a finite number of representative colors are present at any given time. By using a single frame, we specify the number of color clusters we expect to show up. For purposes of our experiments, we use 5 clusters according to our environment. An example is shown in Fig. 4, using sample surgical imagery as the background. The original image from the right camera is shown in 4(a), with a superimposed arrow showing where the flock of birds is located. In Fig. 4(b) we show the result of the k-means clustering using the $ab$ components. Here, pixels are labeled according to the closest of the 5 learned clusters. The colored markers stand out quite cleanly, and are labeled as red pixels. Note that no other pixels in the image are red except for on the markers. This learning stage is performed only once, in the beginning, and we select the cluster label corresponding to the markers to label subsequent images. Then, for any given image, we first convert from RGB to CIELAB. Next, for each pixel, we find the closest cluster according to the k-means result and label as 255 if it's the label we previously selected and zero otherwise. A sample result is shown in Fig. 4(c).

*2) Contour Extraction:* Now that we have a binary image, we wish to extract the best encompassing contour about the segmented region. The challenge here lies in the gaps between the separated vertebrae. So as to not be specific to our hardware, we want our algorithm to be applicable in the case of a continuous bendable segment without gaps. We begin by computing the convex hull around the binary pixel locations, shown in green in Fig. 4(d). However, as the segment bends the boundary actually forms a *concave*




(a) Color Image (b) Cluster Labels (c) Segmented Image (d) Contour

Fig. 4. Color segmentation is performed by using a single color frame [**A**] and a known set of color clusters (5 in our experiments) to label pixels in an image according to the closest color cluster [**B**] in CIELAB color space. The marker pixels fall out cleanly from the learning procedure, labeled as red in **B**. This results in a binary labeling [**C**] according to this cluster of pixels on the markers we wish to analyze. The contour being extracted [**D**] represents a concave polygon. We achieve this by starting with the convex hull, shown in green, and then moving in the points along the contour towards the closest binary-labeled pixel location. The result is shown in red, and this gives a good approximation to the best encompassing contour around the segment.

*polygon*. One half of the segment will be convex, and the hull will be correct there, but the concave portion will be off. The convex hull represents a set of vertices of line segments which form the boundary of the polygon, however we want a denser sampling of this boundary. To achieve this, we iteratively interpolate linearly between successive points in the hull, producing control points along the boundary. Then we find the nearest pixel locations to these control points that originally created the hull (the binary labeled pixels), thereby *moving* the contour locations of the concave areas *inwards* towards the true boundary of the object. The result is shown as the red contour in Fig. 4(d). The orange arrows show the need for this, as contour locations in the concave area need significant adjustment. The closest points can be found either using a linear search or a kd-tree. Both the dimensionality and the number of points is small, and so these searches are not computationally intensive.

As a final step, to get an even and dense sampling of the contour, we take the points that make up the contour and apply the Bresenham line algorithm [28] to consecutive points, assuming local linearity between close points. To augment this contour, we also compute a binary edge map from the segmented image (Fig. 4(c)) using a Sobel operator. This yields points that are interior to the polygon on the boundaries of the markers. We add in these points with the contour locations to build the descriptor, described next. This can be helpful in the case of occlusions, where the contour alone may get deformed, and the edges help diminish the effect of the deformation. Conversely, the edges are not sufficient to fill-in the gaps between the separated markers.

### C. Descriptor Extraction

Next we seek to build a descriptor to represent the 3D pose of the object. The scheme is to compute a feature descriptor on each of the stereo images separately and then combine them together to form a single, composite stereo feature vector. Although we are not explicitly performing 3D reconstruction, 3D shape information is being encoded because we have two separate views of the object in a stereo setup, and ambiguous movements due to out-of-plane perspective effects in one camera can be captured by the other camera. We take the location of the center of the segmented region, and build a 1D histogram of the angles of each point along the contour with respect to the center of the object. Fig. 5 describes this conceptually where the left image shows the angular bins radiating from the center location, displayed as green lines. The bins count the number of points in each angular range and build a histogram, as shown on the right. The histogram should be densely binned so that small changes in shape are captured and the descriptor is sufficiently sensitive, yet not overly noisy. We experimented with bins of size 5° (72 bins), 3° (120 bins), and 1° (360 bins), ultimately choosing the 120-bin case.

The intuition behind this representation is that as the segment bends, the shape redistributes the points along the contour in unique ways. By counting the number of points in each bin, we can analyze the redistribution of these points as the shape changes. It can be thought of as the same total number of points in the bins across the frames, yet redistributed into different distributions within the histogram to capture the shape changes. The example shown in Fig 5 has 16 bins for drawing purposes, however in practice we extract much denser bin sizes.

### D. Training

*1) Pre-Processing:* The training phase consists of mapping known ground truth configuration angles with feature descriptors. First we must collect the raw data from the sensors. Using the magnetic sensor and a stereo camera

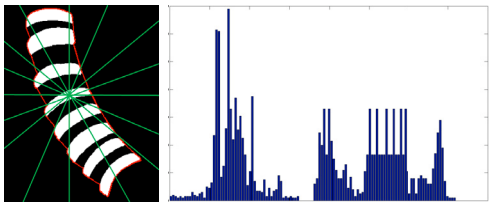


Fig. 5. [**Left**] The descriptor (for a single camera) is constructed by taking points along the contour (red) of the snake segment and computing the angle of each point with respect to the center of the object. The angular bins are depicted as green lines emanating from the center of the segmented region. [**Right**] A densely-binned histogram counts the number of points falling within each angular bin. The drawing on the left is an example showing 16 bins for space considerations, however in our experiments we looked at 72 (5°), 120 (3°), and 360 (1°) bins per image.

system, we collect pairs of stereo images $S_i = \{I_{Li}, I_{Ri}\}$ and associated $\psi_i = \{\theta_{Li}, \delta_i\}$ measurements, where $i = 1, ..., N$ for $N$ discrete training samples.

Each stereo pair $S_i$ is mapped to a feature descriptor $\tilde{H}_i = [H_{Li}^T, H_{Ri}^T]$ to represent the shape of the segment on that frame. Here, $H_{Li}$ is the feature descriptor extracted from the $i^{th}$ left image, and similarly $H_{Ri}$ from the right frame, represented as a single composite feature vector $\tilde{H}_i \in \mathbb{R}^{m \times 1}$.

This gives us an initial training set $\mathbf{A} = \{(\psi_1, \tilde{H}_1), \ldots, (\psi_N, \tilde{H}_N)\}$. However, each $\tilde{H}_i$ is quite high-dimensional, and also may be sparse. Murase and Nayar [26] show that a compact representation of an object's appearance is sufficient for accurate pose estimation by creating a *parametric eigenspace* to represent the appearance. In their case, the images themselves are projected to a lower-dimensional eigen-subspace. For our algorithm, we similarly compute the principal components of the full training set of $\tilde{H}_i$ samples, noting that this is done in feature descriptor space rather than on the original images. In our experiments, we found that we can reduce the dimensionality quite significantly while still recovering a large percentage of the variance.

The principal component analysis (PCA) over $\tilde{H}_i$ yields a set of orthonormal basis vectors $\{e_1, \ldots, e_m\} \in \mathbb{R}^{m \times 1}$. We choose $k < m$ of these eigenvectors to capture a sufficient percentage of the variance of the original feature descriptor training dataset, giving a linear transformation matrix $E = [e_1, \ldots, e_k] \in \mathbb{R}^{m \times k}$, where the columns of $E$ are each of the $k$ basis vectors $e$ representing the top $k$ eigenvalues of the PCA. We project the original training feature descriptor samples in $\mathbf{A}$ to the eigen-subspace:

$$L_i = E^T(\tilde{H}_i - c) \tag{5}$$

where $c$ is the mean feature descriptor over all $\tilde{H}_i$, and $L_i \in \mathbb{R}^{k \times 1}$. We will call the $L_i$ samples *eigen-features*. This gives a final training dataset $\hat{\mathbf{A}} = \{(\psi_1, L_1), \ldots, (\psi_N, L_N)\}$.

*2) Parametric Manifold Interpolation:* We assume that consecutive features are very highly correlated, and so their projections into the eigen-subspace are close together. Our experiments are consistent with this assumption (see section IV). The discrete points $L_i$ describe a smooth parametric manifold represented in eigen-subspace as:

$$G(\psi) = L \tag{6}$$

Depending on the number of degrees-of-freedom (DOFs) represented by $\psi$, the shape of $G$ will vary. For example, if $\psi \in \mathbb{R}^1$, then $G$ represents a curve in $k$-dimensional space. Similarly, if $\psi \in \mathbb{R}^2$, $G$ is a surface, and so on. Using our representation, we obtain a *parametric manifold* which is a surface. The discrete samples are used to interpolate this manifold by performing spline interpolations of $\psi$ to $L$.

A spline is a piecewise polynomial function which can be defined on an $N$-dimensional space to produce a single function value at each $N$-dimensional point. In order to fit points in $\mathbb{R}^2$ to points in $\mathbb{R}^k$, as our manifold describes, we must perform $k$ 2-dimensional spline fits to each of the eigen-feature's output dimensions separately. We use a thin-plate smoothing spline interpolation of the configuration angles $\psi_{i=1,\ldots,N}$ to the eigen-features $L_{i=1,\ldots,N} \in \mathbb{R}^{k \times 1}$. This gives us $k$ sets of spline interpolant coefficients $G_j(\psi_i) = L_{ij}$, where $j = 1, \ldots, k$ corresponds to the $j^{th}$ eigenvector's projection dimension. We then use these interpolant coefficients to *re-sample* the manifold at a higher density over $\psi$'s workspace. This yields interpolated descriptors which smoothly describe these estimated configuration angles.

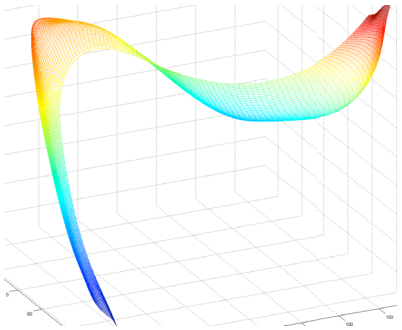


Fig. 6. The interpolated parametric manifold over the 2-DOF pose angles $\psi$, yielding a surface in the feature descriptors eigen-subspace, called *eigen-features*. For purposes of drawing, we only show the first 3 dimensions of the eigen projections.

An example of this manifold is shown in Fig. 6 for the first 3 dimensions of the eigen-features, corresponding to the 3 largest eigenvalues. The discrete training samples that were originally collected (and then projected via PCA) exist as points on (or close to) this manifold. Finally, the resampled manifold points are stored in a *kd-tree* for the querying stage.

### *E. Querying*

Now that we have constructed a densely-sampled manifold and stored it within an efficient look-up data structure, we can query unknown configurations in a very straightforward manner. For any test frame $T$, we extract the feature $\tilde{H}_T$ and using the eigenvector projection matrix $E$ we project $\tilde{H}_T$ using (5) to obtain the test eigen-feature $L_T$. We then use the kd-tree from the interpolated manifold to find the $b$ closest points on the manifold to $L_T$. In our experiments $b = 3$, and each of the $b$ matches provides Euclidean distances $d_l$ in feature space, where $l = 1, \ldots, b$. We use these distances to compute weights, representing the contributions each will make to the final pose estimate, based on proximity in feature space. The weights $w_l$ are computed as:

$$w_l = \frac{1/d_l}{\sum_p \frac{1}{d_p}} \tag{7}$$

The weights contribute as the inverse of the distances in feature space, so that closer points contribute more than further points. The denominator in (7) is provided so that the weights sum to 1. Then a weighted average of the pose angles that created those interpolated eigen-feature matches provides the final configuration estimate for $\tilde{H}_T$:

$$\psi_T = \sum_{l=1}^{b} w_l \psi_l \tag{8}$$

## IV. Experiments & Results

### A. Accuracy of Pose Estimation

First we evaluate the main part of the algorithm, which is to estimate the configuration angles $\psi$ using feature descriptors. We used a stereo camera system composed of two 1024x768 resolution color *Point Grey Research Dragonfly2* cameras mounted on a tripod. During our experiments, the cameras and the snake base remained static. The snake segment was color coded with lime-green markers and we created a background using printouts of photos from a laparoscopic procedure. The segment was manually actuated and the ground truth positions and rotations were collected by interfacing with the Flock-of-Birds sensor.

*1) Training Data:* First, we collected stereo image pairs and associated ground truth rotation measurements. We choose the first frame of the image stream to construct the CIELAB color clusters. Then we manually select the label which corresponds to our marker color. For each subsequent image, we classify each pixel as described in section III-B.1. At the same time, we convert all rotation matrices to $\psi$ configuration angles according to (3) and (4).

These training samples are used to form the initial training dataset $\mathbf{A}$ and then the modified training dataset $\hat{\mathbf{A}}$ according to the method described in III-D. The manifold shown in Fig. 6 is the result of this training procedure, again only showing the first 3-dimensions according to the top 3 eigenvalues from the PCA projections. For our experiments, we collected 2296 training samples.

We experimented with different dimensionalities of the feature descriptors extracted from the individual images. We analyzed bin sizes of 1, 3, and 5 degrees, corresponding to histogram dimensionalities of 360, 120, and 72, respectively. Note that in these cases, the composite stereo descriptor $\tilde{H}$ is twice as long, specifically 720, 240, and 144, respectively. We also experimented with different degrees of dimensionality-reduction in the PCA step in order to test the effect of the loss of dimensions to the overall accuracy. In our experiments, for each of the bin sizes, we looked at different percentages of variance recovery: 65%, 85%, 90% and 95%.

*2) Testing Data:* Testing data is collected in the same way as the training data. For each test measurement $T$, we create the feature descriptor $\tilde{H}_T$ and then the eigen-feature $L_T$ using the projection matrix $E$ obtained from the training data. No test data was used in the creation of the manifold.

*3) Pose Accuracy:* Table I shows results of some of the dimensionality reductions for each of the bin sizes of our feature descriptors. Because we use a kd-tree for feature matching on the eigen-features, we want to reduce this dimensionality for faster matching. In our experiments, we found the best combination of accuracy and run-time efficiency occurring with 120-bin feature descriptors reduced down to 90% variance, resulting in stereo eigen-features $L$ which reside $\in \mathbb{R}^{16}$. This yielded a configuration accuracy with errors of $[\varepsilon_\delta = 0.98^\circ, \varepsilon_{\theta_L} = 1.28^\circ]$ over 806 test samples. With both angles combined together, the overall median pose error was $1.16^\circ$.

TABLE I

Dimensionality Reduction

| N-bins | % Var | Dims | % Var | Dims | % Var | Dims |
|---|---|---|---|---|---|---|
| **72** | 65 | 2 | 85 | 5 | 95 | 19 |
| **120** | 65 | 2 | 85 | 8 | 95 | 39 |
| **360** | 65 | 3 | 85 | 39 | 95 | 200 |

In our experiments, we could not sample a full $360^\circ$ workspace because the snake segment was fixed to a table so that the base could not move. The workspace of our experiments consisted of about 1/2 the entire workspace of the segment, cycling the $\delta$ angle $180^\circ$ through its range and $\theta$ approximately $70^\circ$ for each within-plane rotation. We wanted to ensure that out-of-plane rotations from the imaging plane are sufficiently captured by means of the stereo system. Even though the segment bends in a circular arc, due to perspective effects of camera imaging systems, out-of-plane rotations are viewed as conic sections rather than circles. Often these can be difficult to recover by ellipse-fitting methods, and so our descriptor mapping approach is ideal to avoid these difficult shape-fitting problems.

*4) Occlusions:* We also tested our algorithm against occlusions by manipulating a common laparoscopic tool near the segment, occluding the view from the cameras in a realistic fashion. Although accuracy degrades slightly, we are able to achieve errors of $[\varepsilon_\delta = 1.04^\circ, \varepsilon_{\theta_L} = 2.06^\circ]$, for a combined accuracy of $1.46^\circ$. Fig. 1 shows sample images displaying the kinds of occlusions that were dealt with.

## V. Discussions

**Training Set Size**: One important aspect is in the number of training samples required. Depending on how densely the workspace is sampled, the accuracy of the manifold will vary. If we collect a very dense set of configuration measurements, the problem is reduced to a nearest neighbor matching problem. A strength of the manifold representation

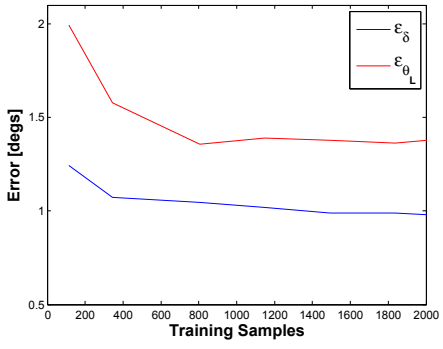



Fig. 7. The results of randomly permuting different percentages of the training data to interpolate the parametric manifold and the effect of this on accuracy. To avoid outliers, the trial for each percentage was done 3 times and the average error was taken as the result.

is that if a small number of training samples is used, the underlying system may still be captured. Fig. 7 shows this observation where we randomly permuted the training samples and selected different percentages to interpolate the manifold and determined the effect on the accuracy. Each trial was done 3 times to avoid outliers, and the average error is shown on the y-axis in degrees. We show the error in $\delta$ and $\theta_L$ separately as the blue and red lines, respectively. Note that even when we only use 15% of the training data (345/2296), we still obtain reasonable results, proving the strengths of the manifold method. In this case, a nearest neighbor approach would be insufficient and the interpolation is required.

**Generalizing The Method**: It's important to note that this method can be extended to interpolate the manifold parametrically using any DOFs which apply to a robot. To describe this idea further, suppose that instead of estimating $\psi$, we wanted to track the 3D position of the endpoint. In this case the manifold would be parametric in 3-DOFs ($x$, $y$, $z$) and this would result in a manifold volume rather than a surface. We ran this experiment using the same data as described in section IV, but using positions from the Flock-of-Birds rather than rotations, and we obtained a median positional accuracy of 0.97mm. In other words, our method can be used to accurately measure different aspects of the robot, depending on the application.

## VI. Conclusions & Future Work

Closed-loop control of surgical robotic systems require a feedback loop with high accuracy in order to perform fine-scaled automated manipulations. Vision is attractive as a low-cost and safe solution, provided that the measurements can be accurate enough to achieve these tasks. In this paper, we have shown a novel method which uses learning to encode visual features that can accurately represent the configuration of a continuum robot robustly. We constructed a parametric manifold which can be indexed in a straightforward fashion to look-up the configuration angles given a descriptor. The manifold is accurate even with a small number of training samples, and is generic enough to represent arbitrary DOFs of a continuum robot. Future work consists of scaling this algorithm to multiple segments to replace the magnetic sensor feedback in [23].

## References


[1] G. Dogangil, B. L. Davies, and F. Rodriguez Y Baena, "A review of medical robotics for minimally invasive soft tissue surgery," *Proc. of the Institution of Mechanical Engineers*, vol. 224, no. 5, pp. 653–679, May 2010.

[2] P. Allemann, M. Schafer, and N. Demartines, "Critical appraisal of single port access cholecystectomy." *The British J. of Surgery*, vol. 97, no. 10, pp. 1476–1480, Jul. 2010.

[3] A. Gumbs, L. Milone, P. Sinha, and M. Bessler, "Totally transumbilical laparoscopic cholecystectomy." *J. of Gastrointestinal Surgery*, vol. 13, no. 3, pp. 533–4, Mar. 2009.

[4] A. N. Kalloo, V. K. Singh, S. B. Jagannath, H. Niiyama, S. L. Hill, C. A. Vaughn, C. A. Magee, and S. V. Kantsevoy, "Flexible transgastric peritoneoscopy: a novel approach to diagnostic and therapeutic interventions in the peritoneal cavity," *Gastrointestinal Endoscopy*, vol. 60, no. 1, pp. 114–117, 2004.

[5] M. Aron, G.-P. Haber, M. M. Desai, and I. S. Gill, "Flexible robotics: a new paradigm." *Current Opinion in Urology*, vol. 17, no. 3, pp. 151–5, May 2007.

[6] N. Simaan, K. Xu, A. Kapoor, W. Wei, P. Kazanzides, P. Flint, and R. Taylor, "Design and Integration of a Telerobotic System for Minimally Invasive Surgery of the Throat." *The Int. J. of Robotics Research*, vol. 28, no. 9, pp. 1134–1153, Sep. 2009.

[7] J. Ding, K. Xu, R. E. Goldman, P. K. Allen, D. L. Fowler, and N. Simaan, "Design, Simulation and Evaluation of Kinematic Alternatives for Insertable Robotic Effectors Platforms in Single Port Access Surgery," in *IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 1053–1058.

[8] S. Hirose, *Biologically Inspired Robots: Snake-like Locomotors and Manipulators*. Oxford University Press, USA, 1993.

[9] G. Robinson and J. Davies, "Continuum robots - a state of the art," in *IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 2849–2854.

[10] I. A. Gravagne and I. D. Walker, "Manipulability, force, and compliance analysis for planar continuum manipulators," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 3, jun 2002.

[11] N. Simaan, R. H. Taylor, and P. Flint, "A Dexterous System for Laryngeal Surgery," in *IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 351–357.

[12] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury, "Mechanics Modeling of Tendon-Driven Continuum Manipulators," *IEEE Trans. on Robotics*, vol. 24, no. 6, pp. 1262–1273, 2008.

[13] R. J. Webster III, J. M. Romano, and N. J. Cowan, "Mechanics of Precurved-Tube Continuum Robots," *IEEE Trans. on Robotics*, vol. 25, no. 1, pp. 67–78, 2009.

[14] P. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and Control of Concentric-Tube Robots," *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 209–225, 2010.

[15] K. Xu and N. Simaan, "Actuation Compensation for Flexible Surgical Snake-like Robots with Redundant Remote Actuation," in *IEEE Int. Conf. on Robotics and Automation*, no. May, 2006, pp. 4148–4154.

[16] V. Agrawal, W. J. Peine, B. Yao, and S. Choi, "Control of Cable Actuated Devices using Smooth Backlash Inverse," in *IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 1074–1079.

[17] S. B. Kesner and R. D. Howe, "Design and Control of Motion Compensation Cardiac Catheters," in *IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 1059–1065.

[18] M. Hannan and I. Walker, "Vision based shape estimation for continuum robots," in *IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 3449–3454.

[19] B. A. Jones and I. D. Walker, "Practical Kinematics for Real-Time Implementation of Continuum Robots," *IEEE Trans. on Robotics*, vol. 22, no. 6, pp. 1087–1099, Dec. 2006.

[20] I. D. Walker, C. Carreras, R. McDOnnell, and G. Grimes, "Extension versus bending for continuum robots," *Int. J. of Advanced Robotic Systems*, vol. 3, no. 2, pp. 171–178, 2006.

[21] D. B. Camarillo, K. E. Loewke, C. R. Carlson, and J. K. Salisbury, "Vision based 3-D shape sensing of flexible manipulators," *IEEE Int. Conf. on Robotics and Automation*, pp. 2940–2947, May 2008.

[22] J. M. Croom, D. C. Rucker, J. M. Romano, and R. J. Webster III, "Visual Sensing of Continuum Robot Shape Using Self-Organizing Maps," in *IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 4591–4596.

[23] A. Bajo, R. E. Goldman, and N. Simaan, "Joint and Configuration Feedback for Enhanced Performance of Multi-Segment Continuum Robots," in *IEEE Int. Conf. on Robotics and Automation*, 2011, p. Accepted.

[24] K. Xu, R. Goldman, D. Jienan, P. Allen, D. Fowler, and N. Simaan, "System design of an Insertable Robotic Effector Platform for Single Port Access (SPA) Surgery," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 5546–5552.

[25] R. J. Webster III and B. A. Jones, "Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review," *The Int. J. of Robotics Research*, Jun. 2010.

[26] H. Murase and S. Nayar, "Visual learning and recognition of 3d objects from appearance," *Int. J. on Computer Vision*, vol. 14, no. 1, pp. 5–24, Jan 1995.

[27] G. Q. Wei, K. Arbter, and G. Hirzinger, "Automatic tracking of laparoscopic instruments by color coding," in *CVRMed-MRCAS'97*, ser. Lecture Notes in Computer Science, 1997, vol. 1205, pp. 357–366.

[28] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, January 1965.