

# Learning-Based Configuration Estimation of a Multi-Segment Continuum Robot

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

**Abstract**—In this paper, we present a visual learning algorithm for estimating the configuration of a multisegment continuum robot designed for surgery. Our algorithm interpolates a stereo visual feature descriptor manifold using Radial Basis Functions (RBFs) to estimate configuration pose angles. Results are shown on a 3-segment snake robot, where rotational accuracy in the range of  $1^\circ - 2^\circ$  is achieved.

## I. INTRODUCTION

Continuum robots are continuously bending elastic structures [1] that promise great advantages for surgical intervention. Their inherent compliance increases safety during surgery and allows for deeper reach inside the anatomy. This capability is essential for new demanding surgical paradigms such as Single Port Access Surgery (SPAS) [2] and Natural Orifice Translumenal Endoscopic Surgery (NOTES) [3].

Continuum robots have been investigated recently by the medical robotics research community for Minimally-Invasive Surgery (MIS) [4], SPAS [5], and NOTES [6]. Different designs and actuation modalities have been proposed [7]–[12], however all suffer from lack of accuracy due to friction, extension and torsion of their actuation lines, kinematic approximations, and actuation coupling between segments.

Methods have been proposed to overcome these issues with model-based [6], [13]–[15] and vision-based approaches [16]–[18]. In [16], the authors 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 [17], the authors used a voxel-carving strategy to extract the position of a flexible manipulator using three orthogonal cameras. In [18], the authors used Self-Organizing Maps and stereo vision to detect the shape of a concentric-tube robot without the use of fiducials for positional accuracy. Finally, a tiered real-time controller that uses both extrinsic and intrinsic sensory information for improved performance of multisegment continuum robots was presented in [19].

In this paper we extend a previous algorithm [20] to estimate the configuration of a single-segment continuum robot. This method interpolates a smooth functional mapping from robot configurations to features using a look-up table. The extension to multiple segments is time-consuming

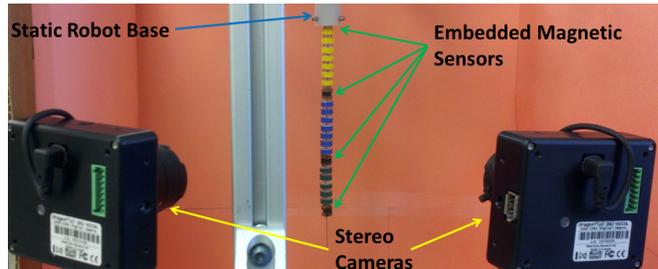


Fig. 1. The experimental setup of our 3-segment robot, where each segment is individually color coded to facilitate segmentation of the vertebrae.

because a high-dimensional nearest-neighbor lookup (with *many* samples) is required.

The contribution in this paper is a method for estimating the configuration of a multisegment continuum robot arm using a visual learning algorithm. Off-line ground truth robot configuration angles are mapped to stereo visual feature descriptors through a smoothly-varying parameterized feature manifold. Radial Basis Functions are used to interpolate this high-dimensional feature manifold to directly estimate the configuration angles from the features, avoiding any memory-intensive look-up tables in the reverse direction. The coupling effects of the independently bending segments are accounted for by expressing the orientation of each segment with respect to a static base coordinate frame. The application of a smoothing operation to the shape descriptor increases the stability and repeatability of the pose estimation through the manifold.

## II. MODELING OF THE MULTI-SEGMENT CONTINUUM ROBOT

A single-segment snake arm is constructed of one centrally located passive primary backbone, and three radially placed secondary backbones with pitch radius  $r$  and separation angle  $\beta$ , as shown on the left of Fig. 2. The segment is moved through the workspace by controlling the lengths of the secondary backbones. The pose of the end disk of segment  $k \geq 1$  can be completely described by the generalized coordinates:

$$\psi_k = [\theta_{L_k}, \delta_k]^T \quad (1)$$

where  $\theta_{L_k}$  and  $\delta_k$  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:

$$\mathbf{R}_k^{k-1} = \mathbf{R}_Z \mathbf{R}_Y \mathbf{R}_Z^T \quad (2)$$

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

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

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

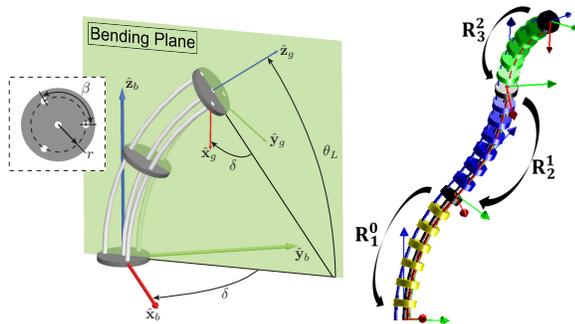


Fig. 2. [Left] Structure and kinematic nomenclature for a single segment continuum robot. [Right] Drawing of our 3-segment continuum robot, displaying coordinate systems and the rotations which represent the configurations.

where  $\mathbf{R}_z = \text{Rot}(-\delta_k, \hat{\mathbf{z}})$ ,  $\mathbf{R}_y = \text{Rot}(\theta_0 - \theta_{L_k}, \hat{\mathbf{y}})$ , and operator  $\text{Rot}(\phi, \mathbf{u})$  returns a rotation matrix of angle  $\phi$  about axis  $\mathbf{u}$  and  $\theta_0 = \pi/2$ . The configuration variables  $\theta_{L_k}$  and  $\delta_k$  can be obtained from (2) as<sup>1</sup>:

$$\theta_{L_k} = \theta_0 - \text{atan2}\left(\sqrt{R_{13}^2 + R_{23}^2}, R_{33}\right) \quad (3)$$

$$\delta_k = -\text{atan2}(R_{23}, R_{13}) \quad (4)$$

where  $R_{ij}$  are the entries of rotation matrix  $\mathbf{R}_k^{k-1}$ . To augment our approach to multiple segments (Fig. 2, right), we use a global representation of each segment, regardless of its kinematic chain. As such, we seek a rotation  $\mathbf{R}_k^0$  which maps the end disk of any segment  $k$  to the static base  $\{0\}$  of the robot. To achieve this, we chain together rotations of segments, proximally, all the way up to the base coordinate frame. For example, given sequentially-ordered segments, it's assumed that the rotation matrix  $\mathbf{R}_k^0$  is formed by composition:

$$\mathbf{R}_k^0 = \mathbf{R}_1^0 \mathbf{R}_2^1 \dots \mathbf{R}_k^{k-1} \quad (5)$$

In this way, we can express every segment of the snake arm in a common coordinate frame and train with these mappings all together. A 3D rotation matrix represents 3 independent DOFs, and so we want a 3-vector to represent this transform. We chose to use **ZYX Euler angles**,  $\mathbf{e}_k = [\alpha_{e_k}, \beta_{e_k}, \gamma_{e_k}]$ , s.t.:

$$\alpha_{e_k} = \arcsin\left(\frac{R_{21}}{\eta}\right) \quad (6)$$

$$\beta_{e_k} = \arcsin(-R_{31}) \quad (7)$$

$$\gamma_{e_k} = \arcsin\left(\frac{R_{32}}{\eta}\right) \quad (8)$$

where  $\eta$  is a normalization factor:

$$\eta = \sqrt{R_{11}^2 + R_{21}^2} \quad (9)$$

In this formulation,  $\alpha_{e_k}$  is a rotation about the z-axis,  $\beta_{e_k}$  is a rotation about the y-axis, and  $\gamma_{e_k}$  is a rotation about the x-axis. Note that other rotation representations may also be used in place of this and easily substituted into our approach. We use  $\mathbf{e}_k$  to represent the 3-vector Euler angles.

<sup>1</sup>We use the atan2 notation such that:  $\theta = \text{atan2}(\sin(\theta), \cos(\theta))$ .

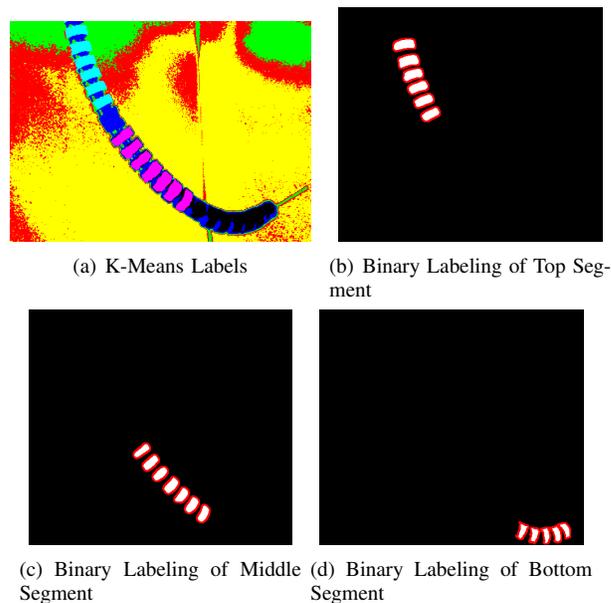


Fig. 3. The binary labelings [white pixels] resulting from the clustering labeling shown in Fig. 3(a), and the associated edges of these binary masks [red pixels], obtained by applying the Sobel operator. The edges are used to build the feature descriptors, described in Sec. IV-B

### III. PARAMETERIZED MANIFOLD

We begin with a description of how we map the feature descriptors to rotation angles. Our method relies on *learning* mappings of features to physical configurations. Our descriptors must be carefully chosen so that nearby points in feature space represent similar configurations. By discretely sampling a continuous space of configurations, we can interpolate a feature descriptor manifold, which is parameterized by the configuration angles  $\mathbf{e}_k$ . In [20] we described the following forward mapping:

$$\mathbf{G}(\boldsymbol{\psi}_k) = L_k \quad (10)$$

where  $\boldsymbol{\psi}_k$  is as described in (1),  $L_k$  is a compressed feature descriptor (through applying Principal Components Analysis (PCA) to a set of training descriptors to reduce the dimensionality, Sec. IV-C), and  $\mathbf{G}$  is the manifold mapping. We showed that we can use training samples to learn  $\mathbf{G}$  and interpolate estimated features  $L_k$  based on arbitrary DOFs  $\boldsymbol{\psi}_k$ . We constructed a look-up table of these interpolated features and estimated unknown configurations by finding the nearest neighbors of an observed descriptor. We used a weighted-average of the configurations that produced the nearest feature matches to recover the configuration angles. Because we were operating with 2 DOFs, the look-up table was reasonably sized in terms of memory. However, as more DOFs are added, this becomes intractable quite quickly.

Ideally, we would like to learn the inverse mapping  $\mathbf{G}^{-1}$  mapping features directly to the estimated rotation angles corresponding to that feature, thereby bypassing any need for a lookup table. Because these features are normally high-dimensional, the interpolation becomes quite complex. We found that using Radial Basis Functions (RBFs) [21]

provided a good approximation to this interpolated mapping if the underlying assumptions of smoothness are sufficiently maintained. As we will show, the descriptor choice becomes incredibly important as features which don't correspond smoothly to DOFs cannot be interpolated accurately.

#### IV. STEREO FEATURE DESCRIPTORS

In order to build a feature descriptor which describes the physical configuration of our continuum segment, we first must select which pixels we will use to build the feature.

##### A. Image Segmentation

The pixels corresponding to each segment must be extracted, and so we use an unsupervised K-Means clustering of pixels in CIELAB color space. The color information exists mainly in the chromaticity-layers  $a$  and  $b$ , and so a single frame is used to learn the most representative colors in the image frame. After training we must manually select which clusters correspond to which snake segments, but once that is done we can automatically extract the segment pixels in each frame independently. A sample clustering result is shown in Fig. 3(a), where different labels are automatically assigned to the pixels in the image according to their color separation. This facilitates segmenting the continuum segments from the image, individually, and results in 3 individual binary labeling, shown as white pixels in Figs. 3(b)-3(d).

Next, we compute edges from these binary images using the Sobel edge detector. This gives us points along the outer-edges of the object, shown as red pixels in Figs. 3(b)-3(d). The intuition behind using these points to build the feature descriptor is that as a particular segment bends, the shape changes redistribute the points along the outer edges in unique ways. Our descriptor seeks to capture and describe these changes smoothly, as described next.

##### B. Descriptor Extraction

Our feature descriptor must be used to represent the 3D configuration of the snake segment. Although we do not explicitly perform stereo reconstruction, we combine feature descriptors from stereo cameras into a single, composite feature. In this way, 3D shape information will be 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 by means of the descriptor.

To build the feature descriptor of a single segment in a single camera, we take the location of the center of the edges and build a 1D histogram of the angles of each point of the edges with respect to the center of the object. A pictorial representation of this descriptor is shown in Fig. 4, whereby we show 14 bins for purposes of describing the idea, however we used 120-bins (every  $3^\circ$ ) in our experiments. The bins radiate outwards from the center of the edge pixels (e.g., the *median* of the points). The bins count the number of edge pixels in each angular range. The histogram should be densely binned so that small changes in shape are captured and the descriptor is sensitive, yet not overly noisy.

Because this descriptor is sensitive to the location of the center point which we use to compute the radial bins, we perform a smoothing operation to provide some invariance to the location of the center that we use. We apply a Gaussian weighting around a bin location when a vote is added to the histogram, performing wrap-around at the boundaries of the histogram to account for the cyclic nature of rotations. We show results on the overall descriptor stability in Sec. V-C.

Finally, to encode the shape in 3D, we collect the feature descriptor from the left and right camera separately and concatenate them together into a single stereo feature vector. These are used to map directly to rotation angles to describe the configuration of a snake segment.

##### C. Feature Pre-Processing

Before constructing the mapping  $\mathbf{G}^{-1}$  of features to rotation angles, we pre-process the features to reduce the complexity using PCA. We use a magnetic sensor to provide ground truth pose angles for training purposes. To collect training data, we collect pairs of stereo images  $S_i = \{I_{Li}, I_{Ri}\}$  and associated  $\mathbf{e}_{k_i} = \{\alpha_{e_{k_i}}, \beta_{e_{k_i}}, \gamma_{e_{k_i}}\}$  measurements, where  $i = 1, \dots, N$  for  $N$  discrete training samples.

The stereo pairs  $S_i$  are mapped to 1-D stereo feature descriptors (for each segment  $k$ )  $\tilde{H}_{k_i} = [H_{k_{Li}}^T, H_{k_{Ri}}^T]$  to represent the shape of the segment on that frame. Our training set then becomes  $\mathbf{A} = \{(\mathbf{e}_{k_1}, \tilde{H}_{k_1}), \dots, (\mathbf{e}_{k_N}, \tilde{H}_{k_N})\}$ . We use PCA to reduce each  $\tilde{H}_{k_i}$  because each is very high-dimensional but also quite sparse. It is shown in [22] that a compact representation of an object's appearance sufficiently captures the pose by creating a parametric eigenspace to represent this appearance. Therefore, we compute the principal components of the training set of  $\tilde{H}_{k_i}$  samples. We observed that we can substantially reduce the dimensionality while still recovering a large percentage of the variance.

We then project the original training feature descriptors in  $\mathbf{A}$  to the eigen-subspace:

$$L_{k_i} = E^T (\tilde{H}_{k_i} - c) \quad (11)$$

where  $c$  is the mean feature descriptor over all  $\tilde{H}_{k_i}$ ,  $E$  is a matrix whose columns consist of the  $b$  eigenvectors corresponding to the top  $b$  eigenvalues we wish to preserve, and  $L_{k_i} \in \mathbb{R}^{b \times 1}$ . We choose  $b$  to capture a sufficient percentage of the variance of the original feature training set. We call these  $L_{k_i}$  samples *eigen-features* and these are used to learn

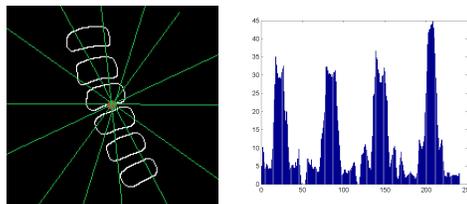


Fig. 4. A pictorial representation of our feature descriptor, for a single segment of the robot from the left camera. Although we portray 14 bins, for purposes of describing the idea, in our experiments we used 120 bins.

the mappings in the parametric manifold. Next we describe the RBF method used to learn this mapping.

#### D. RBF Interpolation

A Radial Basis Function (RBF) is a function based on a scalar radius:

$$\phi(r) = \phi(|x - x_i|) \quad (12)$$

where  $\phi(r)$  can be different types of functions, such as Gaussian, Multiquadric, Linear, Cubic, or Thinplate Splines. In order to use RBFs to interpolate a function mapping  $f(x)$ , an approximation of the function is made by choosing coefficients  $C_0$ ,  $C_1$ , and  $\lambda_i$  to match values of the function at interpolation nodes, such as:

$$f(x) = C_0 + C_1x + \sum_{i=1}^n \lambda_i \phi(|x - x_i|) \quad (13)$$

Once coefficients  $C_0$ ,  $C_1$ , and  $\lambda_i$  are found, we can use this to approximate the function at any point. As mentioned before, we seek the mapping

$$\mathbf{e}_k = \mathbf{G}^{-1}(L_k) \quad (14)$$

so that the estimate of the configuration of a particular segment can be directly computed from the associated stereo feature descriptor extracted from a pair of images. We use RBFs to interpolate this mapping  $\mathbf{G}^{-1}$  from the training data described in section IV-C. This produces a multi-dimensional interpolation procedure mapping descriptor measurements in  $\mathbb{R}^b$  to configurations in  $\mathbb{R}^3$ . Therefore we need to compute 3 different sets of RBF coefficients corresponding to each of the 3 rotation DOFs in  $\mathbf{e}_k$ . More specifically, one set of coefficients will interpolate the mapping of  $L_k$  to  $\alpha_{e_k}$ :

$$\alpha_{e_k} = \tilde{\mathbf{G}}_{\alpha}^{-1}(L_k) \quad (15)$$

such that  $\tilde{\mathbf{G}}_{\alpha}^{-1}$  is an approximation of the true mapping,  $\mathbf{G}_{\alpha}^{-1}$ , which estimates the angles  $\alpha_{e_k}$  from  $L_k$ . We do the same for  $\tilde{\mathbf{G}}_{\beta}^{-1}$  and  $\tilde{\mathbf{G}}_{\gamma}^{-1}$ , which approximate  $\mathbf{G}_{\beta}^{-1}$  and  $\mathbf{G}_{\gamma}^{-1}$ .

In the end, we have coefficients which can efficiently produce estimates of rotations from any snake segment to the static base coordinate frame. This method is in place of our previous approach, which required a look-up table to be constructed from the forward mapping  $\mathbf{G}$  and is sensitive to the density of the training samples. This means isolated training measurements may provide larger errors when interpolating from the nearest-neighbors because of the weighted averaging of nearby samples. However, interpolations using RBFs perform very poorly with descriptors that don't vary smoothly (see Sec. V-D).

## V. EXPERIMENTS

We experimented with the 3-segment continuum robot of Fig. 1, where each segment was coded with individual color markers to facilitate image segmentation. This type of marker-based extraction is common for surgery [23]. In our experiments we used 2 PointGrey Dragonfly cameras for

our stereo imaging. Four magnetic sensors were embedded at points known along the robot arm to produce ground truth of *relative* 3D rotations from each segment to its most proximal segment. We used the Ascension Technology 3D Guidance trakSTAR for ground truth angles, containing 4 Model 90 6-DOF sensors. Position static accuracy of the sensor is 1.4 mm RMS with an orientation static accuracy of  $0.5^\circ$ .

We call the most proximal segment, attached to the static robot base, segment 1. The next distal segment is called segment 2, and the final, distal-most segment is segment 3. Therefore, we seek rotations  $\mathbf{R}_1^0$ ,  $\mathbf{R}_2^0$ , and  $\mathbf{R}_3^0$ , as described in (5). For ground truth,  $\mathbf{R}_1^0$  is given directly by the magnetic trackers. To get  $\mathbf{R}_2^0$ , we are given  $\mathbf{R}_2^1$  from the magnetic trackers and then compute

$$\mathbf{R}_2^0 = \mathbf{R}_1^0 \mathbf{R}_2^1 \quad (16)$$

and similarly for  $\mathbf{R}_3^0$ , we compute

$$\mathbf{R}_3^0 = \mathbf{R}_1^0 \mathbf{R}_2^1 \mathbf{R}_3^2 = \mathbf{R}_2^0 \mathbf{R}_3^2 \quad (17)$$

Then, using the Euler angle conversion expressed in (6), (7), and (8), we collect  $\mathbf{e}_1^0$ ,  $\mathbf{e}_2^0$ , and  $\mathbf{e}_3^0$  for use in training.

The color codings help extract the segments of the snake, individually from each other, using the image processing method described in Sec. IV-A. To begin, we train an RBF to interpolate the mapping  $\mathbf{G}^{-1}$  by collecting descriptors from the stereo images, as described in Sec. IV-B, and then pre-processing them as in Sec. IV-C. For each frame, we get **three** stereo descriptors, each with their respective rotation angles to the static base frame from the magnetic trackers. Because the segments are treated independently, as a freely-rotating body in 3D space, for each stereo pair we get three completely separate training samples, corresponding to each of the 3 segments of the continuum arm. Because a descriptor is mapped to the rotation of the end disk of that segment to the static base (regardless of the segments above it), we can collect these samples and train an RBF to interpolate the function mapping, as described in Sec IV-D. In this way, we remove the coupling effects that would be present with relative rotations. One advantage of treating the segments individually is that if one or more is occluded, we can still compute the configuration of the visible segments because the coupling of the proximal segments above it is accounted for in our formulation.

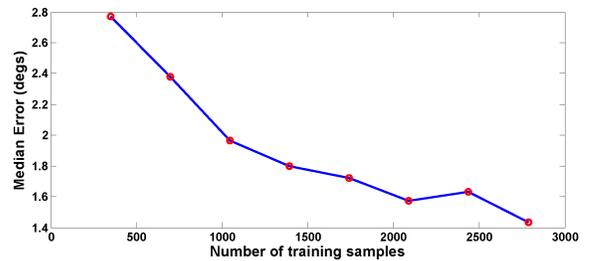


Fig. 5. The effect of number of training samples on overall rotational accuracy.

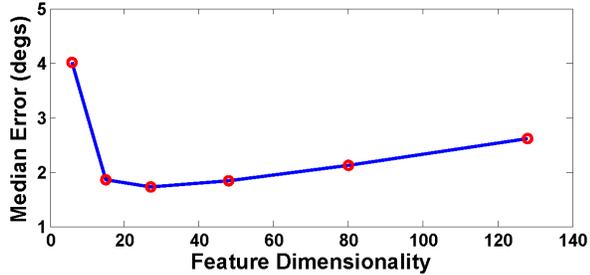


Fig. 6. The effect of PCA feature dimensionality reduction on overall rotational accuracy.

Our experiment consisted of randomly sampling the workspace of the continuum robot and collecting measurements for off-line training and testing. Overall, we collected 3483 total samples. We experimented with different amounts of training data to analyze the effect on the interpolation quality. In each of the cameras we extracted feature descriptors using 120 bins (every  $3^\circ$ ), and so the composite stereo features were 240-dimensional, before applying PCA.

#### A. Training Set Size

Fig. 5 shows our analysis of the effect of the number of training samples to interpolate  $\mathbf{G}^{-1}$  using an RBF with respect to overall median rotational error (in all 3 dimensions of rotation combined). As you can see, even training with only 10% of the data (amounting to 348 samples), we can achieve an overall median accuracy of  $2.77^\circ$ . Our best result comes with more training data, not surprisingly, at  $\sim 1.43^\circ$  of rotational accuracy.

To be sure these experiments are fair, for each trial we randomly permute the data samples and randomly select some for training and the rest for testing, according to the split required by the experiment. We then take the average result over this random sampling of the 3 trials to better capture the behavior of that experiment.

#### B. Dimensionality Reduction

We also experimented with the effect of the PCA dimensionality reduction on the 240-dimensional stereo feature descriptors. We chose to train with 50% of the data and test with the other 50% of the data, making sure the testing data was not included in the interpolation of the training data. Fig. 6 shows the effect of different levels of variance recovery on the overall median rotational accuracy. The recovered feature dimensionality increases as more variance is included in the PCA projections to produce the eigen-features  $L_k$ .

In these experiments, we notice an interesting result. As expected, using the same amount of training data in each experiment we get an increase in accuracy with more dimensions, but only up to a point, where at  $\sim 30$  dimensions (corresponding to 80–85% of the variance recovered) accuracy decreases. We attribute this effect to over-fitting. To investigate this further, we take the higher-dimensional cases and train with more data, and accuracy increases again

to about the same level. This study gives a nice intuition about the trade-off between number of training samples and dimensionality reduction, and how they interact with each other. Again, as previously we randomly permute the training and testing samples.

#### C. Descriptor Stability

The descriptor histogram is built-up by the angles of the segmented pixel locations from the edge maps with respect to the center of the segment in the image. Typically we compute this center location by taking the *median* of the edge pixel locations, to approximate the center of the segment. As we noted earlier, the repeatability of this histogram to describe a particular configuration of a snake segment is sensitive to this center location. Therefore we added a smoothing operation to add some translational invariance to this center location.

To analyze the stability of the descriptor, we devised an experiment as follows. Given a configuration of a snake segment as a test, we compute the descriptor from a single camera as we already described. This is our reference histogram  $H_{ref} \in \mathbb{R}^{120}$ . We then add varying amounts of Gaussian noise to the median center location, using noise levels of 0.1 up to 8 pixels, and recompute the histogram with the noisy center  $H_n$ . Because the units of the descriptor are hard to interpret, we chose to use a percent difference to analyze how the descriptor changes with noisy center locations. The results are shown in Fig. 7. It is observed that the percent difference of the histograms computed with the noisy centers changes linearly with pixel noise. The slope of the best-fit line to this data is 5.2, meaning that for every pixel offset from the true center we get about an extra 5% difference in the histogram.

The percent difference (PD) is an average of the percent difference of every bin in  $H_n$  (for noise level  $n$ ) to the reference histogram  $H_{ref}$ :

$$PD = \frac{\sum_{i=1}^{120} PD_i}{120} \quad (18)$$

where  $PD_i$  is the percent difference of a particular bin  $i \in [1, \dots, 120]$  of the histograms:

$$PD_i = \frac{|H_n(i) - H_{ref}(i)|}{H_{ref}(i)} \times 100 \quad (19)$$

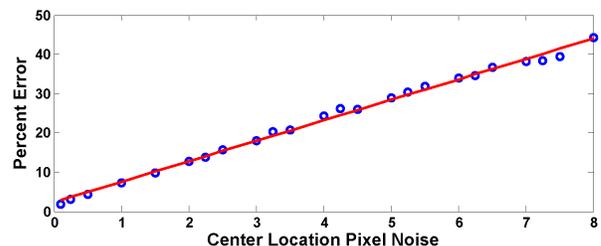


Fig. 7. The stability of our feature descriptor with respect to the noise of the center pixel location.

For each noise parameter, we perform 400 tests with that same noise parameter and take the average PD, again to reduce randomness. Using 0.5 pixels of noise, the PD from the reference is 4.3% and when we apply 2 pixels of noise, the PD grows to 12.8%. Although this still produces a larger percent difference in the descriptor than we would like, the smoothing operation reduces the tendency of the histograms to change significantly when the segment center is noisy. Further increasing this stability is a topic of future research.

#### D. Alternative Feature Descriptors

The choice of our descriptor is not arbitrary. To prove this, we chose a popular descriptor in the computer vision literature called *Histogram of Oriented Gradients* (HoG) [24]. In order for the interpolation mapping to work, the descriptors must vary smoothly with changes in 3D rotation.

We performed the same tests as previously described using HoG descriptors in place of ours. To contrast our approach, we update the **gradient** orientation bins using the gradient magnitude as the vote for that bin using the gray values of the image pixels. Again we combine the HoG's from the two cameras to construct a stereo feature descriptor, ensuring fair comparison. We found that the RBF cannot accurately interpolate the function mapping, and gives rotational errors on the order of  $\sim 100^\circ$  and more. We believe this is the case because the HoG descriptor more accurately describes texture rather than shape, and this provides no smoothness guarantees in the parametric rotation space. This means that nearby HoG's in feature descriptor space do not necessarily describe similar physical configurations in terms of 3D rotation, but rather in appearance, which is not relevant to the problem being addressed in this paper. It's important then that the feature descriptor is more of a *shape* descriptor rather than a *texture* descriptor.

## VI. CONCLUSIONS

In this paper we have described an algorithm to compute the configuration of a multi-segment continuum robot using vision. We constructed a function mapping which trains off of ground truth data to produce 3D rotation angles directly from a stereo feature descriptor. In the future, we plan to run closed-loop live experiments with a snake robot to perform fine-scale manipulations, as in [19], where the algorithm described in this paper can directly feed into the controller. To do this accurately over long periods of time, we will need to develop a tracking algorithm, noting that during a smooth motion of the continuum arm, the configuration angles will change slowly and the estimates may be filtered to avoid jumps and outliers.

## REFERENCES

- [1] 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.
- [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. N. Kalloo, V. K. Singh, S. B. Jagannath, H. Niiyama, S. L. Hill, C. A. Vaughn, C. A. Magee, and S. V. Kantsevov, "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.
- [4] 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.
- [5] A. Bajo, R. Goldman, L. Wang, D. Fowler, and N. Simaan, "Integration and preliminary evaluation of an insertable robotic effectors platform for single port access surgery," in *IEEE Int. Conf. on Robotics and Automation*, 2012.
- [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] S. Hirose, *Biologically Inspired Robots: Snake-like Locomotors and Manipulators*. Oxford University Press, USA, 1993.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] S. Kesner and R. Howe, "Position control of motion compensation cardiac catheters," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1045–1055, 2011.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] M. Hannan and I. Walker, "Vision based shape estimation for continuum robots," in *IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 3449–3454.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] A. Reiter, R. E. Goldman, A. Bajo, K. Iliopoulos, N. Simaan, and P. K. Allen, "A learning algorithm for visual pose estimation of continuum robots," in *IEEE Intl. Conf. on Intelligent Robots and Systems*, 2011.
- [21] M. D. Buhmann, *Radial Basis Functions*. Cambridge University Press, 2003.
- [22] 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.
- [23] G. Q. Wei, K. Arbter, and G. Hirzinger, "Automatic tracking of laparoscopic instruments by color coding," in *First Joint Conf. on Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery*, 1997, pp. 357–366.
- [24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.