2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Vision Based 3-D Shape Sensing of Flexible Manipulators

David B. Camarillo, Kevin E. Loewke, Christopher R. Carlson, and J. Kenneth Salisbury

Abstract— **Rigid robotic manipulators employ traditional sensors such as encoders or potentiometers to measure joint angles and determine end-effector position. Manipulators that are flexible, however, introduce motions that are much more difficult to measure. This is especially true for continuum manipulators that articulate by means of material compliance. In this paper, we present a vision based system for quantifying the 3-D shape of a flexible manipulator in real-time. The sensor system is validated for accuracy with known point measurements and for precision by estimating a known 3-D shape. We present two applications of the validated system relating to the open-loop control of a tendon driven continuum manipulator. In the first application, we present a new continuum manipulator model and use the sensor to quantify 3-D performance. In the second application, we use the shape sensor system for model parameter estimation in the absence of tendon tension information.**

I. Introduction and Background

Flexible manipulators are found in a variety of fields from medical to aerospace where there is a requirement for long thin instruments. In the medical field, the growing popularity of minimally invasive treatments has spurred the development of many new and increasingly complex flexible catheter like instruments. This complexity can be effectively managed with robotic control which provides more dexterous and stable manipulation at reduced scales [6], [4], [24], [21], [23].

Traditional robotic control is based on precise and well defined sensor-actuator systems. The sensors are often encoders or potentiometers located at each joint used to calculate the end-effector position. For a flexible device, however, there are not well defined joints nor an obvious method for instrumentation. This difficulty is most evident for continuum manipulators that articulate by means of material compliance (Fig. 1). These manipulators are often controlled in an open-loop sense but even in that case the model must be calibrated and its performance assessed according to some physical measurement. This need for quantification motivates the development of sensing for flexible manipulators.

Previous researchers have used strain measurements to quantify the articulation of flexible devices. In [16] and [17], piezoelectric actuators and sensors were placed on flexible structures to develop dynamical models and control systems.

D. Camarillo and K. Loewke are with the Department of Mechanical Engineering, Stanford University, Stanford, CA, USA, {dbcamarillo,kloewke}@stanford.edu

C. Carlson is with Hansen Medical Inc., Mountain View, CA, USA, ccarlson@hansenmedical.com

J. Salisbury is with the Departments of Computer Science and Surgery, Stanford University, Stanford, CA, USA, jks@robotics.stanford.edu

This work was supported in part by the Stanford Bio-X Graduate Fellowship and by Hansen Medical, Inc.

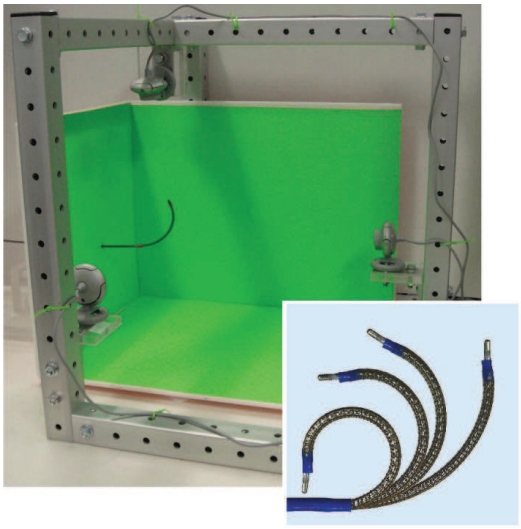


Fig. 1. Vision-based 3-D shape sensing system. The manipulator shown is an intracardiac catheter produced by Hansen Medical [1].

Applying these types of sensors to flexible manipulators can be difficult, however, since they are not easily scalable and the additional instrumentation can be cumbersome. Fiber optic sensors have also been developed for measuring strain on flexible objects [3]. In general, shape sensing based on strain measurement can be inaccurate as it often relies on models that can be incomplete or over-simplified. As a result, other researchers have moved toward using vision to quantify the articulation of flexible devices.

In [12], a single high-speed camera was used to track distinct bands on a continuum manipulator and estimate constant curvatures for each section. While this work showed that the vision-based measurements were much more accurate than the manipulator's internal measurements, the system was limited to 2-D planar articulations. In [11], a single camera was used to track coplanar points on a continuum manipulator and determine the pose of that plane by exploiting its projective homography relative to a reference plane. This approach enabled 3-D sensing with a single camera, but occlusions made it difficult to track all visual markers and for practical implementation it was concluded that multiple cameras were necessary. In [18], a three-camera system was developed for 3-D tracking and parameter estimation of a flexible rope manipulator. This system, however, was designed to track only the endpoint position of the rope, and therefore relied on a model of the rope rather than a measurement its actual shape.

Prior vision-based sensing systems have demonstrated clear benefits, but have also been limited to restricted motions and/or measurements. In this paper we present a new vision based system for estimating and registering the full 3-D shape of a continuum manipulator in real-time. Our sensor system builds upon a technique called shape-from-silhouette, which is a marker-less tracking algorithm that has been traditionally applied by the computer-vision community for human body pose estimation [9]. In this method, multiple cameras are placed at different viewing angles around the object of interest, and each camera extracts a silhouette of the moving object. By projecting these silhouettes and finding their mutual intersection, the 3-D shape of the object can be obtained. Our system, as shown in Fig. 1, uses 3 cameras placed at orthogonal angles around a $12 \times 12 \times 12$ inch working space.

In the following sections, we describe the algorithms and software development for the sensor system. We validate the system for accuracy with known point measurements and for precision by estimating a known 3-D shape. We then present two applications of the validated system relating to the open-loop control of a tendon driven continuum manipulator. In the first application, we present a new continuum manipulator model and use the sensor to quantify its 3-D performance for the first time. Secondly, we develop a model parameter estimation method for the 3-D model. This calibration method uses only the sensor output and tendon displacements whereas the model is normally calibrated using tendon tension measurements.

II. Shape Sensing Algorithm

Our shape sensing algorithm builds upon a technique called shape-from-silhouette, and consists of three main steps. The first step is camera calibration, which is used to determine the intrinsic parameters (focal length, principle point, and lens distortion coefficients) as well as the extrinsic parameters (rotation and translation of each camera relative to the reference frame). The second step is silhouette generation, where each camera extracts a silhouette of the moving object using background subtraction and/or color information. The third step is silhouette projection and intersection, where the silhouettes are projected into a volumetric space in order to find their mutual intersection. This resulting volume represents the 3-D shape of the object, and is often referred to as its visual hull. Fig. 2 shows a screen shot of our system while tracking a flexible manipulator in real-time.

A. Silhouette Extraction

There are several techniques that can be used to generate the silhouette image. One of the most common techniques, known as background subtraction, compares the difference in pixel intensities between the run-time image and a previously acquired background image. If the intensity difference is sufficiently large, the pixel is classified as foreground (silhouette). To remove false-positives due to shadows from the object, additional tests can be performed using the angle between the pixel-vectors in RGB color space [10]. While in practice we have found this technique to work well, our controlled environment allows for a simpler and faster approach based on color-matting [22]. In this approach we subtract the green channel of the background image from the green channel of the run-time image and threshold the result to obtain a silhouette. To fill in missing pixels, we apply a morphological filter (gray-scale dilation) to the silhouette image. Finally, we segment the foreground pixels into a single region using connected-component labeling [14].

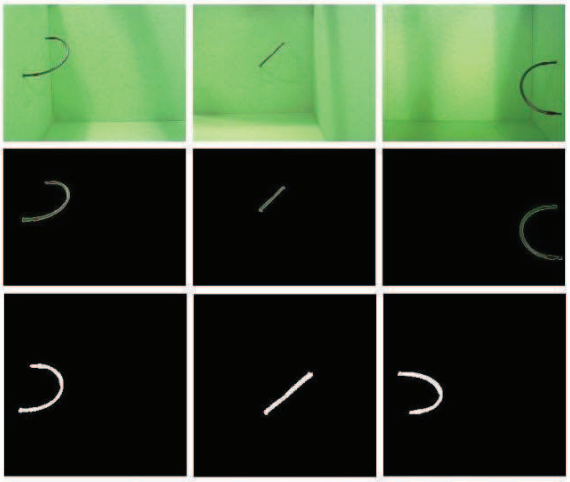


Fig. 2. Screen shot of tracking a continuum manipulator. The top 3 images show the current camera views, the middle 3 images show the silhouettes, and the bottom 3 images show orthogonal views of the 3-D point cloud.

B. Point Cloud from Silhouette Intersections

Once the silhouettes are obtained, they can be projected into a volumetric space to find their mutual intersection. As with silhouette extraction, several techniques are available for finding the visual hull [8], [19]. Here we use a common technique known as voxel carving, where the volume of interest is divided into a discrete three-dimensional grid of volume elements (voxels). Each voxel is back-projected onto the source images, and is classified as part of the 3-D shape if it lies within all of the silhouette images. To enable real-time implementation, a lookup table is created for beforehand that maps every voxel to its respective pixel location in each image. This is described in Algorithm 1.

The function *backProject* in Algorithm 1 takes a 3-D point and back-projects it to a pixel location using the camera's extrinsic and intrinsic parameters. In practice, our tracking system has values of $N_x = N_y = N_z = 256$ and $voxelSize = 1$ mm, and acquires images of size 640×480 pixels. With the lookup table defined, the voxel space can be searched during run-time with Algorithm 2. If the back-projection of a 3-D point lies within all 3 silhouette images, then that 3-D point is considered part of the shape.

Although this method is straightforward to implement, it can be quite time-consuming for large voxel spaces. We therefore use a hierarchical voxel searching method similar to [5], where the voxel space is initially searched at a down-sampled resolution, and then searched at higher resolutions

Algorithm 1 Make Lookup Table

$i \leftarrow 0$
for $x \leftarrow 1 : N_x$ **do**
  for $y \leftarrow 1 : N_y$ **do**
    for $z \leftarrow 1 : N_z$ **do**
      $X \leftarrow origin_x + (x \times voxelSize_x)$
      $Y \leftarrow origin_y + (y \times voxelSize_y)$
      $Z \leftarrow origin_z + (z \times voxelSize_z)$
      for $k \leftarrow 1 : N_{cams}$ **do**
        $pixel \leftarrow backProject(X, Y, Z, k)$
        $lookupTable(i, k) \leftarrow pixel$
      end for
      $i \leftarrow i + 1$
    end for
  end for
end for

Algorithm 2 Shape from Silhouette

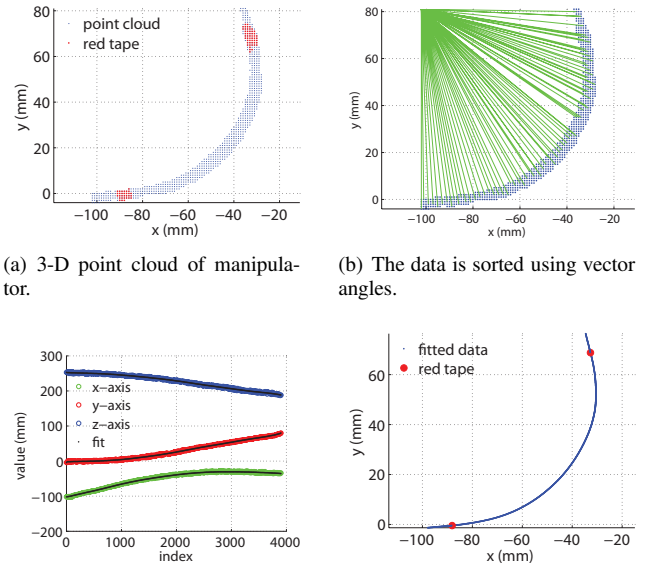
$i \leftarrow 0$
for $x \leftarrow 1 : N_x$ **do**
  for $y \leftarrow 1 : N_y$ **do**
    for $z \leftarrow 1 : N_z$ **do**
      for $k \leftarrow 1 : N_{cams}$ **do**
        $pixel \leftarrow lookupTable(i, k)$
        if $silhouettes(k, pixel) == 0$ **then**
          break
        end if
      end for
      if $k == N_{cams}$ **then**
        $volume(x, y, z) \leftarrow 1$
      end if
      $i \leftarrow i + 1$
    end for
  end for
end for

in a recursive manner. Our tracking system currently runs in real-time at 3-4 Hz on a laptop with a Pentium M 2.13 GHz processor using the Intel OpenCV library.

C. Shape Estimation and Modeling

Up until this point we have established a generic tracking system for estimating the 3-D shape of an object. For our particular application of tracking flexible manipulators, further processing is needed to calculate parameters such as the arc-length and curvature. We begin by modeling our manipulator as a curved cylinder. Our goal is to turn the 3-D point cloud, as shown in Fig. 3(a), into a single spline that runs through the centroidal axis. We achieve this using three processing steps.

The first step is to re-order the point cloud in a meaningful way, since the order of voxel scanning does not necessarily correspond to the shape of the object. Ideally, we want to have the data ordered based on its distance along the length of the manipulator. To do this, we calculate a center value of the point cloud, and define vectors from this average



(a) 3-D point cloud of manipulator.

(b) The data is sorted using vector angles.

(c) A moving average is used to smooth each axis separately.

(d) Final model with single points for end regions.

Fig. 3. After the point cloud is obtained, a three-step process is used to generate the final model.

value to each point. The data is then ordered based on the angle between each vector and a reference vector (the first vector). This is illustrated in Fig. 3(b). An alternative (although slower) sorting algorithm is to check the distance between all pairs of points and sort the data based on shortest distances.

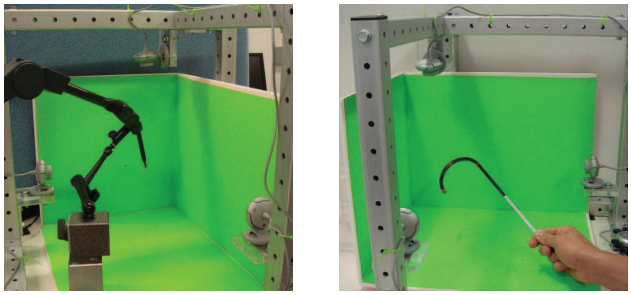
The second step is to smooth the data to create a single spline. For this, we separate the 3-D point cloud into three bins (or vectors) corresponding to the x-axis, y-axis, and z-axis data. The data is then smoothed using a moving average. In practice, we have found that a polynomial fit also works well, and can significantly reduce the effects of noise when calculating curvature values using derivatives. This step is illustrated in Fig. 3(c).

The cylindrical model has a 2 degree-of-freedom null-space in that it does not distinguish torsion and axial compression. We assume that there is no torsion, but compression is significant. Therefore, the third step is to track distinct regions on both ends of the catheter to measure compression. This was achieved by placing small pieces of red tape around the circumference of the manipulator at the two ends. These pieces of tape were identified during voxel carving by looking at the color of the silhouette. For the voxels marked as 'red', the average value was taken and its location was found on the corresponding spline. The final spline of the catheter, along with single points corresponding to the end regions, is shown in Fig. 3(d).

III. Sensor Validation

Evaluating the accuracy and precision of tracking systems is critical for medical applications [20]. This was done for our system through two experiments. The first experiment was designed to evaluate accuracy, and used an Immersion

MicroScribe G2 digitizer [2], as shown in Fig. 4(a). The second experiment was designed to evaluate precision, and used a CNC-machined catheter-shaped object, as shown in Fig. 4(b). These experiments are described in the following sections.

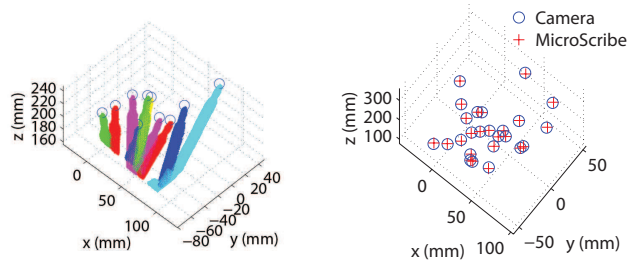


(a) Microscribe digitizer. (b) Machined catheter object.

Fig. 4. Two experiments were performed for validating our tracking system. The Microscribe was used to validate accuracy, and the machined object was used to validate precision.

A. Point Measurements and Registration

To validate the accuracy of our tracking system, we placed the tip of the MicroScribe in the workspace and obtained its 3-D point cloud representation at different poses. We then extracted the tip locations of the MicroScribe's stylus from the point clouds. This data is shown in Fig. 5(a) for 10 different poses. In order to compare the extracted tip locations to the MicroScribe's own position information, we needed to perform a registration between the two data sets. This was achieved using a non-linear least squares routine that solved for a single rigid rotation and translation between the two data sets while enforcing the rotation matrix to be orthonormal. This allowed us to transform the MicroScribe's data points into the tracking system's coordinate system for comparing the measurements. The results are shown for 25 different poses in Fig. 5(b). For this data set, the root-mean-square (rms) errors between the tracking system's points and the MicroScribe's points are $rms_x = .72$ mm, $rms_y = .61$ mm, and $rms_z = .64$ mm. Since the voxel-space is sampled at 1 mm resolution, this establishes that the error in the sensor does not exceed the discretization.



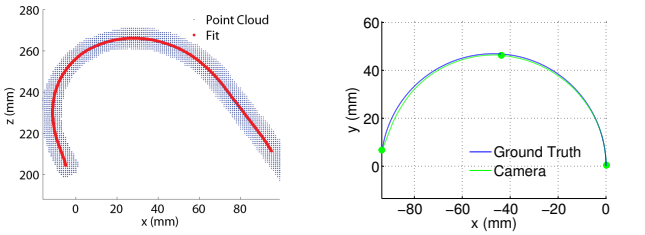
(a) Point cloud and extracted tip location for 10 poses. (b) Comparison of camera sensor to Microscribe for 25 poses.

Fig. 5. Validation of accuracy with the Microscribe.

B. Estimation of a Known Shape

To validate the precision of our tracking system, we manually moved the catheter-shaped object around and obtained its 3-D point cloud representation at different poses. We also placed pieces of red tape on 3 parts of the object. After obtaining the point cloud, we processed the data using the methods discussed in Section II-C. Fig. 6(a) compares the point cloud data to the final fitted model. Fig. 6(b) compares this fitted model to the known shape of the object (obtained from its CAD file). The fitted model was registered to the known object shape by paring data points along the lengths and using the registration scheme described in Section III-A.

This procedure was repeated for 4 different poses of the object to generate RMS error values. First, we computed the difference between the fitted model and the known model along their entire lengths. This was achieved by sampling the known model at discrete points such that the vector-elements of the two data sets were paired. The rms errors between the fitted model and the known model are $rms_x = .69$ mm, $rms_y = .59$ mm, and $rms_z = .24$ mm. For the 3 marked points representing the centers of the pieces of red tape, the rms errors between the tracking system's points and the known points are $rms_x = .49$ mm, $rms_y = .77$ mm, and $rms_z = .47$ mm.

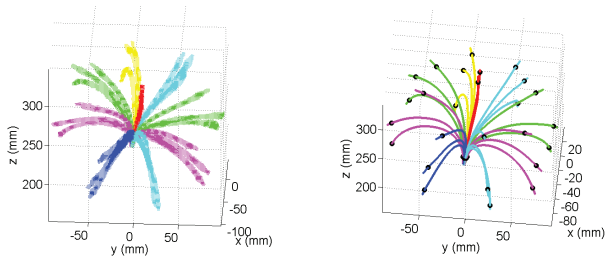


(a) Point cloud of aluminum object and fitted model. (b) Comparison of fitted model to known shape.

Fig. 6. Validation of precision with a machined aluminum object.

IV. Shape Sensing of a Continuum Manipulator

With the sensing system validated for accuracy and precision, we can proceed to the application of shape sensing of a continuum manipulator. For the two applications presented in the following sections, measurements are needed for several different articulations. We took a set of data composed of 26 different static poses that filled the workspace. As discussed earlier, our tracking system obtains the 3-D point cloud representation of the object in real-time. However, since our initial applications are applied to open-loop control, the point cloud data was exported at discrete times. This data was then post-processed using the methods discussed in Section II-C. The results are shown in Fig. 7. The raw data acquired by the sensing system in real-time is shown in Fig. 7(a), and the fitted models for this data that were post-processed are shown in Fig. 7(b). This data is the first quantitative evaluation that we have been able to obtain for 3-D shape of the manipulator.



(a) Point clouds of object. (b) Final splines and endpoints.

Fig. 7. Shape sensing of a continuum manipulator for 26 different poses.

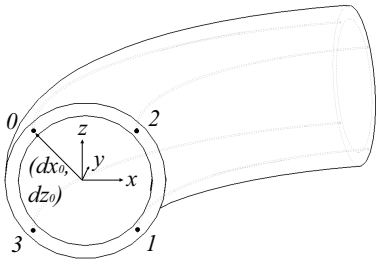


Fig. 8. Illustration of our 3-D continuum manipulator.

V. Evaluation and Estimation of a Continuum Manipulator Model

In this section, we present a new model for a tendon-driven continuum manipulator relating tendon displacement to manipulator configuration. We use the validated sensor system to compare model configuration results to the measured shape in Fig. 7 (and its circular approximation) for two different calibration methods. The manipulator model is presented in only so much detail as required to see its relation to the sensor system. An in depth study of the manipulator model including its relation to previous models (i.e. [24], [15], [13]) is in [7]. In summary, it is a linear model based on the compliant system mechanics that can be used in the forward or inverse directions for n-redundant tendons.

A. Mechanics Based Model of a Tendon Driven Continuum Manipulator

Fig. 8 is an illustration of our continuum manipulator with four co-axial tendons terminated at the distal tip. The manipulator is operated by displacing the tendons by $\mathbf{y} = [\Delta l_0, \Delta l_1, \ldots]^T$ which applies tendon tension $\boldsymbol{\tau} = [\tau_0, \tau_1, \ldots]$ and results in the manipulator bending and compressing with curvature and axial strain specified in the configuration variable $\mathbf{q}$:

$$\mathbf{q} = [\kappa_x,\ \kappa_z,\ \epsilon_0,\ \epsilon_1]^T. \tag{1}$$

Our manipulator as seen in Fig. 1 is composed of a long proximal section that is constrained to prevent bending, followed by the articulating distal section. The distal articulation is characterized by circular arcs (ignoring disturbances) with curvature about two axes κ_x, κ_z. The proximal and distal axial compressive strains are ϵ_0 and ϵ_1.

The goal of the model is to relate tendon displacements $\mathbf{y}$ to configuration shape $\mathbf{q}$. To do so, we begin with the constitutive equation describing linear elastic mechanical response

$$\mathbf{K}_m\,\mathbf{q} = \mathbf{D}\,\boldsymbol{\tau} \tag{2}$$

where

$$\mathbf{D} = \begin{bmatrix} dz_0 & dz_1 & dz_2 & dz_3 \\ -dx_0 & -dx_1 & -dx_2 & -dx_3 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

$\mathbf{D}$ represents the tendon moment arms resulting from a cross-product calculation and when multiplied by $\boldsymbol{\tau}$ yields the general beam moments and compressive force. On the left side of (2), the beam loading is balanced by the general beam strain $\mathbf{q}$ multiplied by the beam stiffness matrix $\mathbf{K}_m = \text{diag}(K_x, K_z, K_{a0}, K_{a1})$. The physical justification for (2) is beyond the scope of this paper but it provides the important relationship between tendon tension and manipulator configuration.

We must now introduce tendon displacement in order to complete the model. When a manipulator is in a loaded state, there are three principle deformations: bending strain, axial strain, and tendon strain. The three of these sum to the total tendon displacement

$$\mathbf{y} = \mathbf{D}^T\,\mathbf{L}_0\,\mathbf{q} + \mathbf{L}_t\,\mathbf{K}_t^{-1}\,\boldsymbol{\tau}. \tag{3}$$

The tendon tension $\boldsymbol{\tau}$ is multiplied by its compliance $\mathbf{K}_t^{-1}$ and scaled by the tendon length $\mathbf{L}_t$ (both diagonal matrices) for the tendon stretch contribution in total the displacement. The manipulator axial and bending strain are contained in $\mathbf{q}$, scaled by the section lengths $\mathbf{L}_0 = \text{diag}(L_1, L_1, L_0, L_1)$ and then translated to lengths along the tendon path by $\mathbf{D}^T$. The net result is the total tendon displacement $\mathbf{y}$ which can be our control input.

With the conservation of strain (3) relating tendon tension and displacement, we can now combine with the constitutive equation (2) to solve for the 3-D forward kinematics as follows:

$$\mathbf{y} = \mathbf{C}_m\,\boldsymbol{\tau} \tag{4}$$

where the compliance matrix is

$$\mathbf{C}_m = \mathbf{D}^T\,\mathbf{L}_0\,\mathbf{K}_m^{-1}\,\mathbf{D} + \mathbf{L}_t\,\mathbf{K}_t^{-1}$$

leading to

$$\mathbf{q} = \mathbf{A}\,\mathbf{y} \tag{5}$$

where the forward kinematics transformation matrix is

$$\mathbf{A} = \mathbf{K}_m^{-1}\,\mathbf{D}\,\mathbf{C}_m^{-1}.$$

Equation (4) describes the input-output relationship from tendon tension to displacement. $\mathbf{C}_m$ is square and invertible since any set of tendon displacements always has an associated set of tensions (ignoring slack for now). Therefore, in (5), $\mathbf{y}$ is multiplied by $\mathbf{C}_m^{-1}$ yielding tendon tension, then multiplied by $\mathbf{D}$ yielding the general beam loading, and

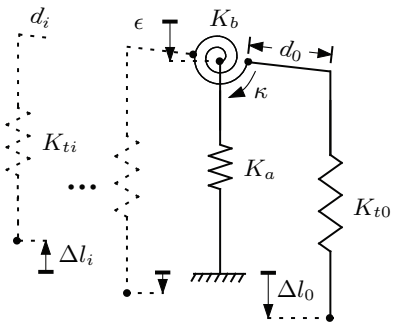


Fig. 9. Spring model for tendon-manipulator system. Shown with a single tendon (solid on right), but can be used to model n tendons (dashed on left).

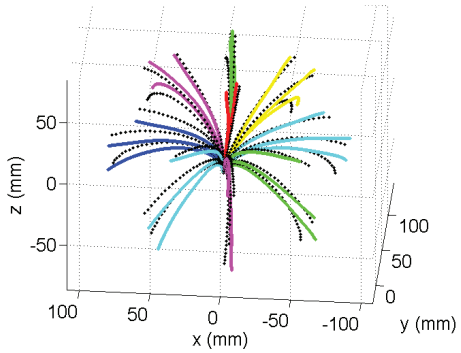


Fig. 10. Results of the sensor output spline (shown by solid color lines) compared to the model prediction (shown by dashed black lines).

TABLE I

COMPARISON OF PARAMETER ESTIMATION METHODS

	Tension Est.	Displacement Est.
Bending stiffness, K_b	457 N-mm	K_{bx}=352; K_{bz}=611 N-mm
Axial stiffness, K_a	2,257 N/ϵ	2,475 N/ϵ
Tip position error	4.9%	4.8%

finally multiplied by $\mathbf{K}_m^{-1}$ to arrive at the manipulator strain or configuration.

A diagrammatic representation of this system is shown in Fig. 9. The rotational spring represent the bending mode, the grounded center spring the manipulator's axial mode, and the springs on the periphery are the tendons. This planar spring system helps to understand intuitively the relationships among the three principle quantities $\boldsymbol{\tau}$, $\mathbf{y}$ and $\mathbf{q}$.

With the complete 3-D kinematics, we can now articulate the manipulator and compare the model prediction to the output of the vision sensor system. The only remaining issues are registering the sensor coordinate frame to that of the model and choosing the model parameters. As for the model parameters, all of the matrix elements in the model are lengths (relatively easy to measure) except $\mathbf{K}_t$ and $\mathbf{K}_m$ that are normalized stiffnesses. The tendon stiffnesses in $\mathbf{K}_t$ are measured externally to the system. The manipulator stiffnesses in $\mathbf{K}_m$ are taken from the tendon tension experiments in [7] that satisfy (2). In those experiments, the two axial modes were lumped together since their independent measurement was much more difficult without the present vision system. The axial stiffness is now broken up into two serial elements which are divided based on the observation of 2:5 distal to proximal compression under equal load. The serial combination of these axial stiffness parameters is specified in the first column of Table I as K_a.

As a mere convenience, the manipulator is articulated by inverting the kinematics (5) to give a reasonable range of tendon displacements $\mathbf{y}$. This is done using the minimum-norm solution after pruning the last row of $\mathbf{A}$ as to not specify the redundant proximal compression and avoid rank deficiency. Configurations $\mathbf{q}$ chosen are arbitrary except that we attempt to fill the workspace (Fig. 7) as well as maintain sufficient axial compression to avoid slack tendons.

Fig. 10 shows the results of the sensor output spline compared to the model prediction. The registration of these two data sets was achieved as a two step process of first identifying the position offset then the orientation. The reference frame for the catheter is embedded at the proximal end of the bend. A band of red tape was placed at the proximal end of the bending section allowing us to locate the position of the model coordinate frame. The orientation of this coordinate frame is defined according to Fig. 8. Since this coordinate frame is not rigidly fixed in space, in each frame we must locate the proximal reference band for a new registration. Presently, the band does not have sufficient structure to specify the full orientation of the frame. However, we assume the orientation does not change from frame to frame due to the kinematics of the proximal section. Therefore, we do a single initial registration by articulating the manipulator in each of its known tendon directions and registering to the model by means of the non-linear least squares method described in Section III-A. We then keep this rotation matrix for all subsequent poses but update the position at each frame.

A relevant metric of performance for many applications is the tip position. Therefore, in Table I we quantify the model performance using the average distance error as a percentage of the $160mm$.

B. Tensionless Model Estimation

As mentioned above, we assume accurate knowledge of all matrices in our kinematic model except the manipulator stiffness matrix $\mathbf{K}_m$. In [7], the stiffnesses are estimated by displacing a single tendon and measuring its force as compared to the curvature and axial compression of the manipulator measured by eye-sight. This is a tedious manual process not well suited for 3-D calibration. Therefore, we develop an alternative method for model estimation that uses only the 3-D sensor system combined with tendon displacements to estimate the manipulator stiffnesses. This allows for calibration without tension sensing hardware.

In order to incorporate the 3-D shape information into the model calibration, it must be translated into a common form. The model is parameterized by circular arcs, so the first step is to approximate the sensor data as circular arcs.

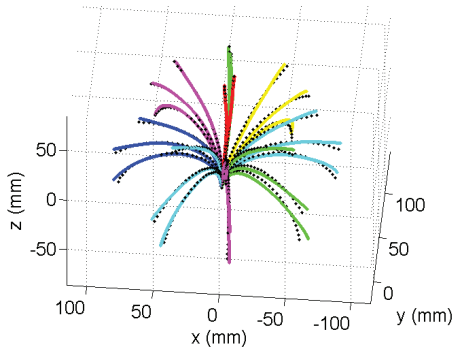


Fig. 11. Results of the sensor output spline (shown by solid color lines) compared to the circle approximation (shown by dashed black lines).

For each pose of the catheter we perform a nonlinear least squares routine to find the curvature and rotation plane that best approximate the data (using the spline length for arc length). The cost function for this routine is the distance error between the data and the circular arc. This circular approximation is easily converted to the curvature and strain quantities found in $\mathbf{q}$. Fig. 11 shows the results for these circular approximations. Another reasonable method that could have been used would align the tip orientations. For the present method, however, we quantify the position fit by the average distance error over each arc as a ratio of the work space for the entire data set. The resulting error is 1.9% and is reasonable enough to train the model with this circular approximation.

The key in achieving tensionless model estimation is to isolate $\mathbf{K}_m$ as a function of only beam configuration $\mathbf{q}$ and tendon displacement $\mathbf{y}$. In order to eliminate $\boldsymbol{\tau}$, we first solve for it in (3) and then insert it into (2) as follows:

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{K}_t \mathbf{L}_t^{-1}(\mathbf{y} - \mathbf{D}^T \mathbf{L}_0 \mathbf{q}), \\ \mathbf{K}_m \mathbf{q} &= \mathbf{D} \mathbf{K}_t \mathbf{L}_t^{-1}(\mathbf{y} - \mathbf{D}^T \mathbf{L}_0 \mathbf{q}). \end{aligned} \tag{6}$$

The right side of (6) can be solved entirely given a data point that consists of the circularly approximated sensor output $\mathbf{q}$ and the associated tendon displacement $\mathbf{y}$. Given 26 data points, we formulated a least-squares solution for the diagonal elements of $\mathbf{K}_m$ listed in the second column of Table I.

The axial stiffness parameter K_a in Table I is a combination of the proximal and distal series spring elements for ease of comparison to the tension based calibration. The value is very close for both the tension based and displacement based calibrations. For the displacement based calibration, we estimate two bend stiffness parameters listed under K_b, the first about the x-axis and the second about z. The nominal values of the 3-D bend stiffnesses are close to the single stiffness value measured in the tension calibration. The directional difference suggests that perhaps the manipulator is slightly more compliant in one direction than the other but is likely a result of estimation error.

To quantify the model performance with this new calibration, we run the tendon displacement data set through the forward kinematics model (5). We then calculate the model predicted tip position to compare against the sensor data. Table I lists the average tip-to-tip distance error as a percentage of work-space.

The performance of the displacement calibrated model is similar to the tension calibrated model. Therefore, we could use either method depending on the availability of multiple tension instrumented axes. However, both of these techniques are fairly basic in that they only solve for a few stiffness parameters. The model is only moderately sensitive to these parameters and no doubt the other matrices have some error and could use an intelligent parameter estimation. In any case, the continuum manipulator is not a completely linear system due to effects such as friction based hysteresis [7], [24]. Therefore, the mechanical construction of a manipulator can directly effect the performance of this linear model even with sound calibration.

VI. Conclusions and Future Work

In this paper we have presented a vision based system for estimating the 3-D shape of a flexible manipulator in real-time. Our system draws on a technique called shape-from silhouette to find a point cloud representation of the object. The sensor system was shown to have millimeter-level accuracy and precision by comparing its output to known point measurements and a known 3-D shape. We subsequently presented two applications of the validated system relating to the open-loop control of a continuum manipulator. In the first application, we presented a new continuum manipulator model and evaluated its performance of tip position estimation to have 4.9% error from the sensed position. In the second application, we explored an alternative calibration method using only the 3-D shape measurements and tendon displacement to estimate the manipulator stiffnesses. Our experimental results showed that the performance of the alternative calibration model was similar to the tension calibrated model. This confirmed that our sensor system can provide full 3-D calibration without the need for multiple axis tension sensing hardware.

Future work on the sensor side could begin with establishing a full registration to the manipulator's reference frame. This might require further markings to specify the axial rotation of the manipulator as well as an improved method for calculating the pointing direction of the red tape markers. This complete registration will be essential in measuring manipulators with multiple serial sections used for higher degree-of-freedom manipulation. The registration could also be reversed to estimate the locations of the tendons when their placement is not well known and thereby improve model performance. We also plan to optimize the current algorithms to increase the speed and improve robustness.

On the manipulation side, there are many other potential applications of this sensor system. The sensor could be used for other model calibration schemes that estimate multiple matrix parameters and potentially integrate tension sensing with multiple tension controlled axes. In addition, the real-time performance of the sensor system allows for the opportunity to pursue closed-loop feedback control or model

improvement in real-time.

VII. Acknowledgments

The authors would like to thank the staff and management team at Hansen Medical, Inc. for their support and cooperation. In particular, we would like to thank Alex Goldenberg, Dr. Neal Tanner, Tom Bourne, and Aditya Koolwal.

References

[1] Hansen medical website. http://www.hansenmedical.com/.

[2] Immersion corporation website. http://www.immersion.com/.

[3] Fiber optic position and shape sensing device and method relating thereto, 2005. U.S. Patent 20060013523.

[4] A. Al-Ahmad. Utilizing robotic catheter control technology for EP procedures. *EP Lab Digest*, 7(8):20–22, 2007.

[5] F. Caillette and T. Howard. Real-time markerless human body tracking with multi-view 3-d voxel reconstruction. *Proceedings of British Machine Vision Conference (BMVC'04)*, September 2004.

[6] D. Camarillo, T. Krummel, and K. Salisbury. Robotic technology in surgery: past, present, and future. *American Journal of Surgery*, 188(4A):2S–15S, 2004.

[7] D. Camarillo, C. Milne, and K. Salisbury. Mechanics modeling of tendon driven continuum manipulators. *Under review for IEEE Transactions on Robotics*, 2008.

[8] K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part 1: Theory and algorithms. *International Journal of Computer Vision*, 62(3):221– 247, May 2005.

[9] K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part 2: Applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, 63(3):225–245, August 2005.

[10] K. M. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2:714–720, June 2000.

[11] V. Chitrakaran, A. Behal, D. Dawson, and I. Walker. Setpoint regulation of continuum robots using a fixed camera. *Robotica*, 2007.

[12] M. Hannan and I. Walker. Real-time shape estimation for continuum robots using vision. *Robotica*, 23:645–651, 2005.

[13] S. Hirose. *Biologically Inspired Robots, Snake-Like Locomotors and Manipulators*. Oxford University Press, Oxford, 1st edition, 1993.

[14] A. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.

[15] B. A. Jones and I. D. Walker. Kinematics for multisection continuum robots. *IEEE Transactions on Robotics*, 22(1):43–55, 2006.

[16] S. Leleu, H. Abou-Kandil, and Y. Bonnassieux. Piezoelectric actuators and sensors location for active control of flexible structures. *Transactions on Instrumentation and Measurement*, 50(6):1577–1582, 2001.

[17] S. Lyshevski. Data-intensive analysis and control of flexible pointing systems with pzt actuators. *Proceedings of International Frequency Control Symposium and PDA Exhibition*, pages 948–956, 2003.

[18] T. Matsuno, T. Fukuda, and F. Arai. Flexible rope manipulation by dual manipulator system using vision sensor. *Proceedings of IEEE International Conference on Advanced Intelligent Mechatronics*, pages 677–682, July 2001.

[19] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. *Eurographics Workshop on Rendering*, 2001.

[20] C. Nafis, V. Jensen, L. Beauregard, and P. Anderson. Method for estimating dynamic EM tracking accuracy of surgical navigation tools. *Proceedings of SPIE Medical Imaging: Visualization, Image-Guided Procedures, and Display*, 6141, 2006.

[21] R. Satava. Emerging technologies for surgery in the 21st century. *Archives of Surgery*, 134(11):1197–1202, 1999.

[22] A. Smith and J. Blinn. Blue screen matting. *Proceedings of the 23rd annual conference on computer graphics and interactive techniques*, pages 256–268, 1996.

[23] R. Taylor. Robots as surgical assistants: Where we are, wither we are tending, and how to get there. *Proc. AIME*, pages 3–11, 1997.

[24] K. Xu and N. Simaan. Actuation compensation for flexible surgical snake-like robots with redundant remote actuation. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4148–4154, May 2006.