String Motif-Based Description of Tool Motion for Detecting Skill and Gestures in Robotic Surgery

Narges Ahmidi[1], Yixin Gao[1], Benjamín Béjar[2], S. Swaroop Vedula[1],
Sanjeev Khudanpur[1,3], René Vidal[1,2,3], and Gregory D. Hager[1]

[1] Department of Computer Science
[2] Department of Biomedical Engineering
[3] Department of Electrical and Computer Engineering,
Johns Hopkins University, Baltimore, MD 21218, USA

Abstract. The growing availability of data from robotic and laparoscopic surgery has created new opportunities to investigate the modeling and assessment of surgical technical performance and skill. However, previously published methods for modeling and assessment have not proven to scale well to large and diverse data sets. In this paper, we describe a new approach for simultaneous detection of gestures and skill that can be generalized to different surgical tasks. It consists of two parts: (1) descriptive curve coding (DCC), which transforms the surgical tool motion trajectory into a coded string using accumulated Frenet frames, and (2) common string model (CSM), a classification model using a similarity metric computed from longest common string motifs. We apply DCC-CSM method to detect surgical gestures and skill levels in two kinematic datasets (collected from the da Vinci surgical robot). DCC-CSM method classifies gestures and skill with 87.81% and 91.12% accuracy, respectively.

Keywords: surgical motion, descriptive models, gesture and skill classification, geometry, descriptive curve coding, robotic surgery.

1 Introduction

Methods that are currently used to assess acquisition and maintenance of surgical skill in the training laboratory and operating room suffer from significant shortcomings [1]. Existing methods are focused on either subjective global evaluation of performance or unstructured, descriptive feedback [1,2]. Some evaluation metrics such as total task completion time and path length reasonably correlate with surgical skill but are not instructive, i.e. they provide limited information to the trainee on whether and how to improve their performance in different stages of the task. On the other hand, unstructured, descriptive feedback typically requires the presence of a senior surgeon and is inefficient [1,2].

The advent of robotic surgery has created new opportunities to automate objective assessment of skill acquisition by surgical trainees. Because surgeons may exhibit different levels of skill at various stages of the task, automated skill assessment requires the detection of gestures that are being performed. Prior approaches to

K. Mori et al. (Eds.): MICCAI 2013, Part I, LNCS 8149, pp. 26–33, 2013.

automatically detect surgical skill and gestures have significant performance and utility limitations (Table 1) [3-9]. For example, Hidden Markov Models (HMMs) and other statistical methods such as Linear Dynamical Systems (LDS) were used to detect surgical gestures with reasonable accuracy (around 85%). Our objective in this paper is to present and evaluate a general approach for signal representation, called Descriptive Curve Coding (DCC) using accumulated Frenet frames (AFF) followed by analysis of string motifs, which can be used to simultaneously identify both surgical skill and gestures using kinematic data describing surgical motion.

2 Experiment Setup

We used two datasets to develop and validate our methods. The first dataset (DS-I) has been described in detail elsewhere [5,9]. It contains 39 trials of a four-throw continuous suturing task (performed by 8 surgeons in multiple sessions) on a bench-top model using the da Vinci surgical robot (Intuitive Surgical, Inc., Sunnyvale, California). The second dataset (DS-II), with 110 trials, was collected from 18 surgeons performing interrupted suturing followed by either a square knot or a surgeon's knot using the da Vinci surgical robot [10]. The operators performed multiple sessions over several days, repeating the suture/knot-tying task three times in each session. The surgical task in DS-II is more complex compared to that in DS-I.

The data from each task was manually segmented, i.e., the start and the end of every gesture in a task were annotated, by watching the endoscopic-video recordings. The gesture labels were specified by an experienced surgical educator, and manually assigned by two researchers in our lab (88% average chance-corrected-agreement between annotators). We used 10 gestures from DS-I (total of 787 sample gestures) under the same experiment setup explained in [5]. We used seven gestures from DS-II: grab needle, grab suture, grab suture-tail, pull needle, pull suture, rotate suture once, rotate suture twice. Because kinematic data does not contain information about the surgical environment (e.g., object being held), we combined the gestures into three context-free groups: grab (722 samples; average duration of 2 sec), pull (431 samples; average duration of 1.3 sec), and rotate (137 samples; duration of 3.8 sec).

Our ground truth for skill assessment consisted of Global Rating Scores (GRS) assigned based on the Objective Structured Assessment of Technical Skills (OSATS) approach [11]. An experienced surgical educator, who was masked to the identity of the operator, assigned the scores by watching video recordings of operators performing the tasks. The OSATS approach is comprised of six elements; each one scored using a Likert scale ranging from 1 to 5. In practice, a single GRS is assigned for the entire task, whereas automated assessment can be continuous over the task (i.e., assigns a skill level to each gesture in a task).

We considered trials for which an operator was assigned a score of 3 on at most two items and a score 4 or 5 on the other items on the GRS as being at "expert" skill level; trials with a score less than 3 on all items on the GRS as being at "novice" skill level; and trials that fell in between the expert and novice categories as being at "intermediate" skill level. DS-II contains 30 novice-level trials, 37 intermediate-level trials, and 43 expert-level trials.

3 Methodology

Our approach is comprised of three steps (Fig. 1): feature extraction (signal representation through AFF and DCC), training (string-motif-based model, metrics, and classification), and finally evaluation of the classification on a test dataset.

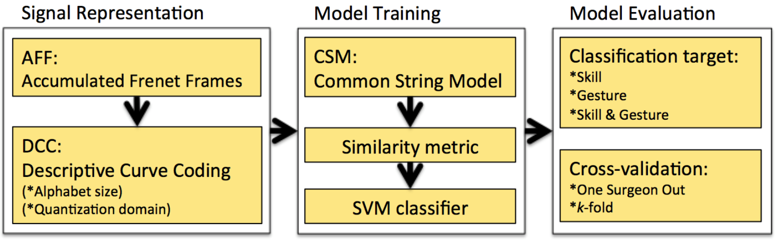

Fig. 1. Three steps comprising our approach for gesture/skill classification

3.1 Accumulated Frenet Frames (AFF)

In [4], we introduced the idea of DCC using Frenet Frames (FF), which assign a local coordinate system to each point of a trajectory, based on the local curvature. The coding system defined on FFs is coordinate-independent and thus independent of the surgical setup. However, FFs alone do not adequately represent the curvature of some smooth trajectories, as explained below. Therefore, we introduce AFF, which accumulates changes in direction of the motion trajectory over short spatial or temporal windows and thus is more sensitive to gradual changes.

The tool tip movement is represented by a sequence of local frames (Fig. 2a). Each frame is comprised of three orthogonal unit vectors: $\vec{v_i}$ follows the tangent of the curve ($\overrightarrow{w_i}$), $\vec{u_i}$ is the normal vector following the concavity of the curve, and $\vec{n_i}$ is the binormal vector formed as the cross-product of $\vec{v_i}$ and $\vec{u_i}$. In the original FF, $\vec{v_i}$ directly followed the tangent vector, whereas in AFF, it follows the last considerable change of direction (defined in 3.2). The AFF accumulates small changes of direction until they are large enough to update the frame orientation.

3.2 Descriptive Curve Coding (DCC)

DCC transforms the time series of AFF into a coded string representation of tool motion by mapping each motion to a small set of canonical directions. Let S ={0,1,…, n} be the index set of vectors comprising the coding alphabet. The following equation is used to generate an alphabet representing direction changes of $\pi/2^p$:

$$[\vec{x}]_p = [\vec{x}]_{p-1} \cup \left[\left(\overrightarrow{x_i + x_j} \right) / \left\| \overrightarrow{x_i + x_j} \right\| \mid \left| \vec{x_i} \otimes \vec{x_j} \right| \neq 0;\ \vec{x_i}, \vec{x_j} \in [\vec{x}]_{p-1} \right]$$
$$[\vec{x}]_1 = \left[\vec{0},\ \ \vec{v_i},\ \ \vec{v_i} \otimes \vec{u_i},\ \ \vec{u_i} \otimes \vec{v_i},\ \ \vec{u_i},\ \ \overrightarrow{-u_i},\ \ \overrightarrow{-v_i} \right]$$

For example, Figure 2a shows a set of vectors encoding direction as cardinal directions (the base case of equation above, p=1). $[x]_1$ contains six orthogonal vectors indicating the possible changes of direction plus no-motion (DCC7): 'x_0: no-motion', 'x_1: forward', 'x_2: left', 'x_3: right', 'x_4: down', 'x_5: up',' x_6: backward'. $[x]_2$ defines a 19-element alphabet (DCC19), which includes all bisectors of DCC7 vectors. Fig. 3 illustrates differences in representation between FF and AFF and between DCC7 and DCC19. The representation by AFF is closer to the original shape than the representation by FF (Fig. 3). In addition, using a larger alphabet size (DCC19 vs. DCC7) results in signal representations that are closer to the original shapes.

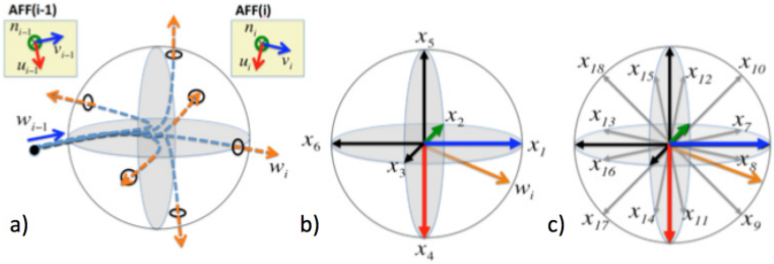

Fig. 2. Possible changes in the direction of the motion trajectory at a given point: a) the change of direction of the motion trajectory between w_i (current window) and w_{i-1} (previous window) is encoded using a set of predefined possible vectors *[x]*. b) seven element alphabet when changes larger than $\pi/4$ are of interest (DCC7), or c) nineteen element alphabet when changes larger than $\pi/8$ are of interest (DCC19).

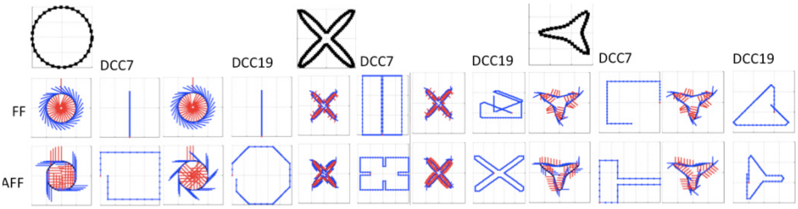

Fig. 3. First row: the original shape. Even columns show the representation of the curve after coding by DCC7 or DCC19, respectively. Odd columns show *u* (red) and *v* (blue) vectors of either FF (second row) or AFF (third row) along the curve. Using AFF with DCC19 clearly provides the most detailed approximation to the original shape.

3.3 Common String Model (CSM)

DCC encodes each motion trajectory j as a string $\mathrm{T_j}$ of length m. We hypothesize that within this string, there are recurring string patterns (string motifs) that correlate with the gesture being performed, and the skill with which it is being performed. Thus, within a training set $\{\mathrm{T_j}\}$, we apply an algorithm to extract the 'Longest Common String' (LCS) for each pair of strings $< T_\mathrm{x}, \mathrm{T_y} >$ using the dynamic time warping approach. LCS returns three values: the longest common motifs (C), number of joint occurrences (N), and the set of motif locations in each training string (O). The collection of triples $< \mathrm{C}, \mathrm{N}, \mathrm{O} >$ computed by applying LCS to each pair of strings in

$\{T_j\}$ forms the dictionary D. For example, if LCS finds a common motif C between T_x and T_y and the motif C occurs at elements X, Z in T_x and element Y in T_y, the entry in the dictionary would be:

$$\mathrm{LCS}(T_x, T_y) = < C, N_{xy} = 3, \qquad O_{xy} = \{X, Y, Z\} >$$

If C is found in other pairs of strings, for example in $< T_h, T_q >$, then we update the dictionary entry for C by summing N_{xy} and N_{hq}, and merging the sets O_{xy} and O_{hq}.

3.4 Similarity Metric and Classifier

We used the following equation to measure the pseudo-similarity value between a given string T_y and a dictionary D:

$$\text{Similarity}\left(T_y, D\right) = \frac{1}{d} * \sum_{C_i \in D \cap T_y} (\log(P_i) + w_1 * \log(|C_i|) + w_2 * \log(A_i))$$

$$A_i = \frac{1}{1 + \sum_{J \in O} |Y - J|}$$

where C_i represents each of the *d* motifs found in both T_y and CSM. P_i is the frequency of motif C_i appearing in the model and $|C_i|$ represents the length of the motif C_i. A_i is a measure of the mis-alignment between the location of C_i in T_y (denoted as Y) and the location of C_i in the training samples (J). The weight factors w_1 and w_2 are learned using a gradient descent algorithm to optimize the performance of the metric for a given classification problem as explained below. The similarity metric tends to assign higher values to test strings with longer matching motifs with the model. However, it also applies a misalignment penalty for motifs that do not occur during corresponding segments of the surgical task.

3.5 Training

For a given classification problem, a dictionary is computed for each class of interest. For a given set of weights the pseudo-similarity provides a value determining the affinity of a given test string to each dictionary. A Support Vector Machine (SVM) classifier is trained using a feature vector S (comprised of pseudo-similarities), and the performance is tabulated. The weights, w_1 and w_2, are optimized to maximize the classifier's performance by tuning the pseudo-similarity to be as discriminative as possible for a given task.

4 Evaluation and Results

We analyzed DS-I for gesture classification and DS-II for both gesture and skill classification. For gesture classification in both DS-I and DS-II, we apply three methods assuming that the boundaries are known (from manual annotation): HMM [9], LDS [5], and DCC-CSM. The HMM method was configured as a 3-state, 3-mixtures (3S3M), and a 3-state, 1-mixture (3S1M), along with 9-dimensional linear discriminant analysis (LDA). The LDS algorithm used Martin and Frobenius distance metrics, and nearest neighbor (NN) and SVM as classifiers. We tried different orders for the dynamical models (range from 3 to 15) and reported the best results obtained. We also examine the performance of DCC-CSM under varying conditions – using two types of "windows" for encoding the motion trajectory (with resolutions of either 0.25mm in the spatial domain or 25ms in the temporal domain), and using alphabets of two sizes (DCC7 and DCC19).

For our evaluations, we used three validation methods – leave-one-session-out ("SO"), leave-one-user-out ("UO"), and *k*-fold cross-validation (leave 20% out). For our analyses using DS-I, we under-sampled gesture classes to include 43 samples (median of class size) to create balanced (B) training datasets. We also report results of our analyses on DS-I using unbalanced (UB) training datasets (except DCC-CSM). Tables 2 to 5 include both macro- and micro- averages of correct classification.

We investigated the following using DCC-CSM on DS-II: gesture classification for a known skill level, skill classification for a known gesture, and simultaneous gesture and skill classification (assuming neither is known). For simultaneous classification of gesture and skill, we trained nine models (3 gestures times 3 skill levels). The model that is most similar to the test sample represents both gesture and skill.

Table 1. Gesture detection performance with unknown boundaries – micro averages of correct frames- reported in the literature for various algorithms using dataset DS-I. *=one-trial-out.

Suturing task	HMM- 3S3M	S-LDS [9]	HMM-HLDA [9]	FA-HMM [9]	SHMM [6]
SO	72%*	80.79%	74.13%	78.27%	81.1%
UO	69%	67.1%	N/A	57.2%	67.8%

Table 2. Gesture classification performance, assuming known boundaries, macro (top number) and micro (bottom number) averages, percentage of correct segments on DS-I (chance=10%)

Suturing task		HMM 3S3M	HMM 3S1M	LDS NN Mar	LDS NN Fro	LDS SVM Mar	LDS SVM Fro	DCC7, spatial	DCC7, temporal	DCC19, spatial	DCC19, temporal
SO	B	67.49 71.69	85.98 87.12	66.97 74.60	74.59 80.00	72.46 80.12	82.53 87.69	-	-	-	-
	UB	68.43 73.68	88.72 91.81	67.10 82.22	74.29 83.65	64.88 81.62	80.23 90.01	79.92 80.10	86.79 85.26	82.92 81.39	77.88 83.23
UO	B	49.54 56.45	59.61 64.04	54.77 63.28	58.41 65.66	62.16 69.35	68.07 75.76	-	-	-	-
	UB	48.67 57.08	63.49 71.60	55.77 69.02	56.98 70.10	59.63 75.41	69.82 81.39	74.70 78.18	75.88 79.92	79.65 78.56	72.28 80.81

On DS-I, previous algorithms for gesture classification, assuming unknown boundaries, achieved a classification accuracy of up to 81% (Table 1). In contrast, assuming known boundaries, most variants of HMM, LDS, and DCC-CSM methods achieved comparable or better gesture classification accuracies on the same dataset (Table 2). Specifically, HMM with the simpler model (3S1M), LDS (SVM Fro), and DCC7 (temporal) appear to perform better than the other methods we evaluated. Since DCC-CSM memorizes the performed pattern, its performance decreases significantly with smaller classes. Those insignificant performances were excluded from Table 2.

On DS-II, DCC-CSM methods (DCC7, temporal) classified gestures more accurately than HMM and LDS methods within nearly all skill levels (Table 3). All methods were sensitive to training with data leaving-one-user-out.

Our findings on gesture classification using DS-I are not directly comparable to those using DS-II. However, the near perfect gesture classification by DCC-CSM methods on DS-II suggests that these methods are sensitive to size of the training sample. DCC-CSM methods using temporal windows were more accurate than those using spatial windows. We believe that the differential performance between temporal and spatial windows is because the former captures change over time, which is a component of velocity of the motion trajectory. DCC19 was not consistently more accurate than DCC7, as we anticipated this maybe because DCC19 yields shorter strings than DCC7 that are then commonly seen for each gesture.

Table 3. Gesture classification performance (with known skill level), macro (top number) and micro (bottom number) averages, percentage of correct segments on DS-II (chance=33%).

Evaluation method	Known skill level	HMM 3S3M	HMM 3S1M	NN Mar	NN Fro	SVM Mar	SVM Fro	DCC7 spatial	DCC7 temporal	DCC19 spatial	DCC19 temporal
k-fold	Novice	59.98 61.35	73.05 75.78	70.97 69.92	70.82 71.18	71.93 75.27	74.78 78.75	90.77 91.83	96.94 98.35	75.77 75.85	79.87 80.89
	Inter	70.31 67.81	81.14 83.46	71.45 72.03	71.80 71.99	71.12 74.38	76.63 78.10	76.62 83.78	98.81 99.25	92.14 91.17	82.74 83.80
	Expert	69.41 69.13	80.26 81.96	66.44 66.76	69.91 68.54	63.64 70.00	67.34 71.74	87.78 91.38	94.81 95.90	86.14 87.46	88.35 92.15
UO	Novice	53.89 55.73	64.17 63.93	64.15 67.08	64.56 65.70	66.53 71.14	75.42 74.78	69.38 67.99	74.18 83.39	47.32 55.40	85.33 89.58
	Inter	75.17 68.63	74.02 78.63	69.28 69.87	66.95 69.48	70.19 73.18	72.90 76.01	63.88 54.91	70.03 74.07	70.94 70.18	74.23 75.76
	Expert	52.95 59.62	71.14 76.05	57.72 60.60	61.29 64.92	53.13 62.08	64.96 72.38	57.06 45.71	88.32 89.73	71.66 78.57	75.52 78.08

DCC-CSM methods achieved nearly 98% skill classification accuracy for the "pull" gesture (Table 4). DCC-CSM methods simultaneously classified both gesture and skill with nearly 97% accuracy (k-fold; Table 5), which was sensitive to training with data leaving out one user.

Table 4. Skill classification performance (with known performed gesture), macro/micro averages on DS-II (chance=33%)

Evaluation method	Known gesture	DCC7 spatial	DCC7 temporal	DCC19 spatial	DCC19 temporal
k-fold	grab	90.85 91.14	84.78 86.16	88.26 89.10	82.74 83.45
	pull	90.20 90.00	97.85 97.86	90.66 90.68	96.85 96.85
	rotate	90.25 91.60	93.13 93.60	88.64 89.00	95.23 95.80
UO	grab	63.17 64.22	91.88 91.45	40.41 42.44	78.60 78.73
	pull	72.52 73.20	86.26 86.71	68.45 70.10	71.29 71.10
	rotate	85.29 85.87	75.39 76.09	72.02 71.74	74.76 75.00

Table 5. Simultaneous gesture and skill classification performance (with no prior knowledge) on DS-II (chance=11%)

Evaluation method	DCC7 spatial	DCC7 temporal	DCC19 spatial	DCC19 temporal
k-fold	94.98 96.62	84.78 96.92	88.48 91.37	92.41 91.37
UO	66.05 67.47	70.91 78.35	47.46 50.30	58.83 68.64

5 Conclusion

We evaluated DCC-CSM methods, and compared them with existing HMM and LDS methods, for detection of surgical gestures and skill on two datasets using two cross-validation approaches. DCC-CSM methods were at least as accurate or were more accurate than HMM and LDS methods under most experimental conditions on both datasets. Whereas HMM and LDS methods relied on a detailed kinematic representation of surgical tool motion, DCC-CSM methods used only position and orientation of the tool-tip. DCC-CSM methods offer a way to seamlessly classify both gesture and skill. The sensitivity of DCC-CSM methods to detect various surgical gestures, in other datasets, using alternate similarity metrics and different methods to assign the ground-truth for skill has yet to be evaluated.

Acknowledgment. This work was funded by NSF CDI-0941362 and CPS CNS-0931805. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Benjamín Béjar was supported in part by the Talentia Fellowships Program of the Andalusian Regional Ministry of Economy, Innovation and Science. The authors would like to thank Dr. Grace Chen for scoring the surgeries and Anand Malpani, Lingling Tao and George Chen for assistance in data annotation.

References

1. Bell, R.H.: Why Johnny cannot operate. Surgery 146, 533–542 (2009)
2. Gearhart, S.L., Wang, M.H., Gilson, M.M., Chen, B., Kern, D.E.: Teaching and assessing technical proficiency in surgical subspecialty fellowships. Journal of Surgical Education 69, 521–528 (2012)
3. Reiley, C.E., Lin, H.C., Yuh, D.D., Hager, G.D.: A Review of Methods for Objective Surgical Skill Evaluation. Surgical Endoscopy 25, 356–366 (2011)
4. Ahmidi, N., Hager, G.D., Ishii, L., Gallia, G.L., Ishii, M.: Robotic Path Planning for Surgeon Skill Evaluation in Minimally-Invasive Sinus Surgery. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part I. LNCS, vol. 7510, pp. 471–478. Springer, Heidelberg (2012)
5. Zappella, L., Béjar, B., Hager, G., Vidal, R.: Surgical gesture classification from video and kinematic data. Medical Image Analysis (2013)
6. Tao, L., Elhamifar, E., Khudanpur, S., Hager, G.D., Vidal, R.: Sparse Hidden Markov Models for Surgical Gesture Classification and Skill Evaluation. In: Abolmaesumi, P., Joskowicz, L., Navab, N., Jannin, P. (eds.) IPCAI 2012. LNCS, vol. 7330, pp. 167–177. Springer, Heidelberg (2012)
7. Rosen, J., Solazzo, M., Hannaford, B., Sinanan, M.: Task decomposition of laparoscopic surgery for objective evaluation of surgical residents' learning curve using hidden Markov model. Computer Aided Surgery 7(1), 49–61 (2002)
8. Dosis, A., Bello, F., Gillies, D., Undre, S., Aggarwal, R., Darzi, A.: Laparoscopic task recognition using hidden Markov models. Studies in Health Technology and Informatics 111, 115–122 (2005)
9. Varadarajan, B.: Learning and inference algorithms for dynamical system models of dexterous motion. PhD thesis, Johns Hopkins University (2011)
10. Kumar, R., Jog, A., Vagvolgyi, B., Nguyen, H., Hager, G.D., Chen, C.C.G.: Objective measures for longitudinal assessment of robotic surgery training. The Journal of Thoracic and Cardiovascular Surgery 143(3), 528–534 (2012)
11. Martin, J.A., Regehr, G., Reznick, R., MacRae, H., Murnaghan, J., Hutchison, C., Brown, M.: Objective Structured Assessment of Technical Skill for Surgical Residents. British Journal of Surgery 84, 273–278 (1997)