Feasibility of Real-Time Workflow Segmentation for Tracked Needle Interventions

Matthew S. Holden, Tamas Ungi, Derek Sargent, Robert C. McGraw, Elvis C. S. Chen, Sugantha Ganapathy, Terry M. Peters, *FIEEE*, Gabor Fichtinger, *Senior Member, IEEE*

***Abstract*—Computer-assisted training systems promote both training efficacy and patient health. An important component for providing automatic feedback in computer-assisted training systems is workflow segmentation: the determination of what task in the workflow is being performed. Our objective was to develop a workflow segmentation algorithm for needle interventions using needle tracking data. Needle tracking data were collected from ultrasound-guided epidural injections and lumbar punctures, performed by medical personnel. The workflow segmentation algorithm was tested in a simulated real-time scenario: the algorithm was only allowed access to data recorded at, or prior to, the time being segmented. Segmentation output was compared to the ground-truth segmentations produced by independent blinded observers. Overall, the algorithm was 93% accurate. It automatically segmented the ultrasound-guided epidural procedures with 81% accuracy and the lumbar punctures with 82% accuracy. Given that the manual segmentation consistency was only 84%, the algorithm's accuracy was 93%. Using Cohen's d statistic, a medium effect size (0.5) was calculated. Because the algorithm segments needle-based procedures with such high accuracy, expert observers can be augmented by this algorithm without a large decrease in ability to follow trainees in a workflow. The proposed algorithm is feasible for use in a computer-assisted needle placement training system.**

***Index Terms*—workflow segmentation, epidural, lumbar puncture**

I. Introduction

CLINICIAN competency is a key factor in patient health and safety. In particular, clinicians gain competency through training methods that include feedback components [1].

We propose a workflow segmentation algorithm that is able to identify a user's motions based on needle tracking data. The proposed algorithm will produce real-time feedback, which improves skill retention in medical trainees over simple efficiency metrics [2].

A. Background

Traditionally, trainees learn and practice needle interventions under the supervision of an expert clinician. However, this approach requires significant time investment from the expert and assessment criteria and feedback can vary significantly between experts [3]. The traditional training protocol is thus neither optimally time efficient nor maximally beneficial for skill acquisition.

We suggest that computer-assisted needle placement training can be introduced to improve the training protocol. If such a training system were able to provide standardized feedback and evaluation [1], it could complement expert supervision in many situations.

One important feature of a computer-assisted training system is the ability to follow the trainee in the procedural workflow in real-time, as the procedure is performed. That is, the system must be able to discretely determine what task the trainee is performing at all times. Here, we refer to this as workflow segmentation. In this paper, we propose an algorithm for workflow segmentation of needle-based procedures based upon only tracking data from the needle.

A workflow segmentation algorithm in a computer-assisted needle placement training system must take tool tracking input and produce task labeling as an output in real-time. Additionally, the algorithm should require manually segmented procedures as the only input for training and use limited procedure-specific information (i.e. instructions to user). The algorithm should not require any particular structure to the procedure (the tasks may be performed in any order, with arbitrary repetition), need no manual preprocessing of the data, and allow the user to perform the procedure as normal. That is, the user may pick any ordering of tasks and the algorithm must determine what task is being performed at all times. In particular, the algorithm must handle the case when the user redoes a task. Moreover, the sequence of tasks in the testing data might never be encountered in the training dataset. We propose a workflow segmentation algorithm which satisfies these criteria for implementation in a computer-assisted training system.

Manuscript received December 13, 2013. M. S. Holden was supported by the NSERC Alexander Graham Bell Canada Graduate Scholarship. T. Ungi was supported by the Queen's University-Ministry of Research and Innovation Ontario Post-Doctoral Fellowship. Funding support for this project in T. M. Peters' laboratory is from the Canadian Institutes for Health Research and the Canadian Foundation for Innovation. G. Fichtinger was funded as a Cancer Care Ontario Research Chair.

M. S. Holden, T. Ungi and G. Fichtinger are with the School of Computing, Queen's University, Kingston, Ontario, Canada (e-mail: mholden8@cs.queensu.ca, ungi@cs.queensu.ca, gabor@cs.queensu.ca).

D. Sargent and R. C. McGraw are with the School of Medicine, Queen's University, Kingston, Ontario, Canada (e-mail: dsargent@qmed.ca, 3rcm4@queensu.ca)

E. C. S. Chen and T. M. Peters are with Robarts Research Institute, Western University, London, Ontario, Canada (e-mail: chene@robarts.ca, tpeters@robarts.ca).

S. Ganapathy is with the London Health Sciences Centre, London, Ontario, Canada (e-mail: Sugantha.Ganapathy@lhsc.on.ca).

B. Previous Work

Several previous reports have proposed workflow segmentation algorithms for tool-based interventions. Their algorithms use techniques that are outlined below.

One important requirement of our application is that it must allow tasks within the workflow to be repeated and to occur in any sequence. Castellani *et al.* [4] achieve 84% online workflow segmentation accuracy using Markov Models and support vector machines. Ahmadi *et al.* [5] achieve 92% workflow segmentation accuracy using an algorithm based on dynamic time warping. Lalys *et al.* [6] report 93% workflow segmentation accuracies using support vector machine techniques in combination with Markov Models. Padoy *et al.* [7] use Markov Models to perform workflow segmentation on pin placement and suturing tasks. Padoy *et al.* [8] achieve 97% workflow segmentation accuracy using a dynamic time warping algorithm and 92% using a Markov Model algorithm. Each of these algorithms performs workflow segmentation on procedures in which the order of tasks is known beforehand (or is defined by the training data), and thus, these algorithms are not applicable to a system for segmenting a procedure with an unknown sequence of tasks.

Real-time feedback and instruction is an important part of our application. Hundtofte *et al.* [9] show workflow segmentation accuracies of 85% using Markov Models. Lin *et al.* [10] report 89% workflow segmentation accuracy by linear discriminant analysis and Bayes classifiers. Tao *et al.* [11] achieve offline workflow segmentation accuracy of 83% using sparse Markov Models. Each of these algorithms requires future observations to identify the workflow segmentation, thus, cannot be applied in real-time.

Identifying the task transition points is often more difficult than identifying what task is being performed between known transition points. For real-time applications, the algorithm must automatically identify transition points as they occur. Reiley *et al.* [12] demonstrated motion classification accuracies of up to 93% using Markov Models, which performed with higher accuracy than linear discriminant analysis and Gaussian mixture models. Varadarajan *et al.* [13] show gesture recognition accuracies as high as 87% using linear discriminant analysis and Markov Models. Ahmidi *et al.* [14] achieve 78% motion recognition accuracy using *k*-means clustering and Markov Models for explicitly delineated tasks. In each of these algorithms, transition points must be manually identified by an observer, and thus, these algorithms are not suitable for real-time workflow segmentations.

There exist several techniques for determining whether a tool trajectory follows a curve or a surface which investigate a similar motion classification problem. Li *et al.* [15] use Markov Models and virtual fixture techniques for robotic curve following procedures, and achieve 94% motion intent accuracy. Aarno *et al.* [16] use a layered Markov Model technique to perform workflow segmentation on curve following tasks. These algorithms only identify whether a predefined curve is followed, and this does not correspond directly to needle-based procedures in which a single task may be performed in various different ways. Video-based techniques provide different challenges than workflow segmentation techniques based on tool tracking data. James *et al.* [17] demonstrate 75% workflow segmentation accuracy with video-based techniques using parallel layer preceptor techniques. Haro *et al.* [18] use linear dynamical system and “Bag of Features” techniques to perform surgical video classification with 89% accuracy. These techniques require a known sequence of tasks and manual segmentation, respectively, so are not usable for our application.

From our literature review, no workflow segmentation algorithms have been found that meet all the criteria for following a trainee in an intervention in real-time. Our proposed workflow segmentation algorithm is novel in that it satisfies all these criteria for implementation in a computer-assisted needle placement training system.

II. Methods

We propose an algorithm that at each timestamp in the tool trajectory produces a discrete task label. Timestamps in the tool trajectories were represented as seven element vectors where the first three elements were position values and the last four elements were the quaternion components associated with the rotation. All components of the vector were treated equivalently in the algorithm. First, the algorithm must be trained using data with known ground-truth workflow segmentation. Then, the trained algorithm can automatically generate a workflow segmentation of a testing procedure.

A. Workflow Segmentation Algorithm

1) Gaussian Filter: While noise is inevitably associated with any modality of tool tracking, its effects can be removed using filtering techniques. Here, noise is reduced by a moving average Gaussian filter of the form described in Eq. 1.

$$s(t) = \frac{\int_0^T x(\tau) N(\tau - t; \sigma^2) d\tau}{\int_0^T N(\tau - t; \sigma^2) d\tau} \tag{1}$$

This filter is applied to each degree of freedom separately. Because this filter requires integration over a finite interval, the divisor must be introduced for normalization. *N* is the normal distribution with variance σ^2.

2) Orthogonal Transformation: To map the continuous tool trajectory in time to a vector space with distinctly represented motion features, an orthogonal transformation, which transforms the data into a higher-dimensional vector space with extracted features, is applied [19]. The transformation takes the form described by Eq. 2.

$$o(t) = \int_{t-\Delta t}^{t} s(\tau) P_i(\tau) d\tau \tag{2}$$

The orthogonal transformation is computed for each degree of freedom separately. The functions P_i refer to Legendre polynomials of i^{th} order. The time interval Δt and order of the transformation are input parameters to the algorithm.

Orthogonal transformation has not been previously used for feature extraction in workflow segmentation. This step is motivated by previous works using orthogonal transformations in the handwriting recognition [19] and speech recognition [20] literature. This step is validated by calculating the accuracy of the algorithm with and without orthogonal transformation.

3) Principal Component Analysis: The orthogonal transformation re-expresses the data in a higher-dimensional vector space. Since this is often problematic for data mining algorithms, principal component analysis is used to project the data into a lower dimensional space. The principal component analysis takes the form described in Eq. 3.

$$\bar{p}(t) = [\bar{o}(t) - mean(\bar{o}(t))]eig[\mathrm{cov}(\bar{o}(t))] \tag{3}$$

By taking only the eigenvectors corresponding to the largest eigenvalues of the covariance matrix [21], the dimensionality of the vector space is reduced. The number of retained eigenvectors is an input parameter to the algorithm.

4) K-Means Clustering: A modified *k*-means clustering algorithm is used to discretize the output of the principal component analysis into discrete cluster labels. This algorithm is preferred over the traditional *k*-means clustering algorithm using random starting points, because of its deterministic nature, and its improved sum of squared errors when the number of clusters approaches the number of data points. The modified algorithm is based on the fast global *k*-means algorithm [22], and is applied as described in Eq. 4.

1) Pick the data point farthest from any centroid, and create a new centroid at that data point.
2) Iteratively recalculate cluster memberships and cluster centroids until cluster memberships do not change.
3) Repeat until *k* centroids have been added.

$$c(t) = kmeans(\bar{p}(t)) \tag{4}$$

The first centroid is initialized as the mean of the entire set of observations. The number of centroids *k* is an input parameter to the algorithm.

5) Markov Modeling: A Markov Model is used to relate the sequence of cluster labels to task labels. The Markov Model represents tasks in the workflow as states and cluster labels as observations. The Markov Model estimation algorithm (Eqs. 5, 6, 7) is employed to determine the model's parameters [23].

$$\pi = \frac{\sum_{t=0}^{T}(Task(t) == i)}{\sum \pi} \tag{5}$$

$$A_{i,j} = \frac{\sum_{t=0}^{T}((Task(t) == i) \wedge (Task(t+1) == j))}{\sum A_i} \tag{6}$$

$$B_{i,j} = \frac{\sum_{t=0}^{T}((Task(t) == i) \wedge (c(t) == j))}{\sum B_i} \tag{7}$$

The values of the function *Task* are determined from the manual segmentation of the data. The most likely task that the user is performing is determined using a modified, real-time Viterbi algorithm [4], where the task label is computed only for the most recent time step. No particular structure to the Markov Model is enforced. Optionally, physically impossible task transitions may be disallowed. Additionally, because the training data does not cover all possible task sequences, all other state transition and observation probabilities are set to small, non-zero values δ.

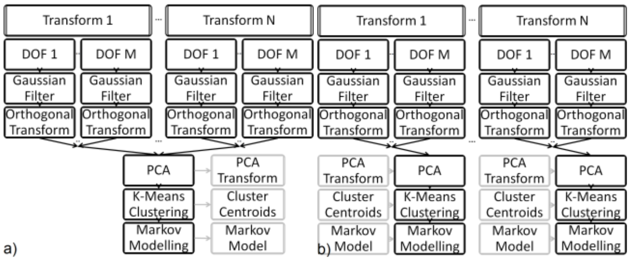

Fig. 1. Block diagram outlining workflow segmentation algorithm a) training and b) testing. Algorithm steps are indicated in black, quantities computed during algorithm training phase are indicated in gray.

B. Algorithm Training

The workflow segmentation algorithm is first trained using tool trajectories from procedures with known ground-truth workflow segmentations through the following steps (Fig. 1a):

1) Apply Gaussian filter.
2) Apply orthogonal transformation.
3) Calculate principal component analysis transformation.
4) Calculate *k*-means cluster centroids and memberships for each task label separately.
5) Train Markov Models using the estimation algorithm with ground-truth workflow segmentations.

C. Algorithm Testing

The trained workflow segmentation algorithm can subsequently be used to automatically identify the workflow segmentation of a test procedure using the following steps (Fig. 1b):

1) Apply Gaussian filter.
2) Apply orthogonal transformation.
3) Apply principal component analysis transformation calculated from training data.
4) Determine cluster membership using centroids from training data.
5) Determine the most likely task based on the sequence of cluster labels, using the modified Viterbi algorithm.

D. Ultrasound-Guided Epidural Procedure

Ultrasound-guided epidural procedures were used to validate the proposed workflow segmentation algorithm. In total, 88 procedures were collected from 16 novice residents/clinicians and 6 expert clinicians. Procedures were performed from both the left and right sides of a poly-vinyl chloride (PVC) plastic spine phantom with printed plastic vertebrae and silicone-rubber skin. The experimental setup is illustrated in Fig. 2.

First, an expert briefed each participant in the ultrasound-guided epidural procedure workflow, using a spine model. The expert demonstrated proper placement of the ultrasound probe and the needle using the paramediam epidural access approach [24] and explained the corresponding workflow:

P1) Probe Translation: Place the probe para-sagitally (parallel and offset spine) to view the facet joints.

P2) Probe Rotation: Angle the probe medially (towards middle) to visualize the lamina and interlaminar space.

N1) Translation: Place the needle-tip on the skin, in-plane and inferior to ultrasound probe.

N2) Rotation: Angle the needle in-plane to the ultrasound probe.

N3) Insertion: Insert the needle into the interlaminar space, entering the epidural space.

N4) Verification: Verify the epidural space (by ultrasound).

N5) Retraction: Remove the needle from tissue.

Each participant was allowed one or more practice procedure on each side of the phantom before their tool trajectories were recorded. Each participant performed at least one and no more than three ultrasound-guided needle placement procedures on each side of the phantom.

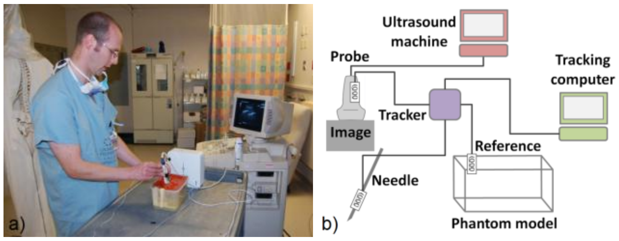

Fig. 2. a) Photograph of ultrasound-guided epidural setup (tracking computer not shown). b) Schematic of ultrasound-guided epidural setup illustrating systems used in the experiment. Coil shapes indicate that the object was tracked electromagnetically.

Although both the needle and the ultrasound probe were tracked, only tasks involving the needle (N1-N5) were used for workflow segmentation (tasks P1-P2 involving the probe only were not included, to demonstrate that our algorithm works for procedures involving one tracked tool). While the above tasks performed in order describes the procedure's optimal workflow, users were allowed to perform the tasks in any order and repeat tasks as necessary to achieve success.

Participants used a 21 gauge, Chiba-tip needle (diameter 0.82mm). Ultrasound imaging was performed using the Aloka SSD-1700 ultrasound machine (Hitachi Aloka Medical Ltd.) with a 30mm curve-linear probe (model UST-9104-5) at 5.0MHz. The needle, ultrasound probe, and phantom were tracked using the NDI Aurora electromagnetic tracking system (Northern Digital Inc.). The needle was tracked by a 5DOF sensor integrated in the stylet (0.7mm root-mean-square accuracy, no information about rotation about the needle axis was available); 6DOF sensors were affixed to the probe and phantom for tracking (0.5mm root-mean-square accuracy).

The ultrasound-guided needle placement procedures were divided into groups based on skill level and whether the procedure was performed from the left or right side of the spine. Group sizes were: Novice Left 35, Novice Right 32, Expert Left 10, Expert Right 12. The leave-one-out cross-validation method was performed separately for each group. Data from all groups were used to calculate accuracy statistics.

The algorithm's parameters (see Appendix for precise values) were optimized to produce the optimal mean segmentation accuracy for the ultrasound-guided epidural procedures. This was performed by optimizing each parameter individually and iterating, using manual supervision to avoid local optima. The parameter set optimizing the workflow segmentation accuracy for the recorded set of ultrasound-guided epidurals is not necessarily optimal for other datasets. We conjecture, however, that the optimal parameter set for other procedures will be similar. Thus, the calculated optimal parameter set for the ultrasound-guided epidurals is an estimate of the optimal parameter set for other datasets.

E. Lumbar Puncture Procedure

Lumbar puncture procedures were used to verify the proposed workflow segmentation algorithm and its applicability to multiple procedures with different tool tracking and ultrasound setups. Procedures from 12 self-reported novices performing the lumbar puncture were tracked and recorded. The procedures were performed on a PVC plastic spine phantom with printed plastic vertebrae, silicone-rubber skin, and a rubber ligamentum flavum. Fig. 3 illustrates the system setup.

First, a demonstrator instructed each participant in the lumbar puncture workflow, and demonstrated the procedure on the spine phantom. The demonstrator explained appropriate needle-guidance and verification techniques. In particular, the demonstrator explained and demonstrated the following procedural workflow:

N1) Translation: Place the needle-tip on the skin's surface, centered between the L3-4 or L4-5 vertebrae.

N2) Rotation: Angle the needle 15° cephalad.

N3) Insertion: Puncture the skin, and insert the needle into the subarachnoid space.

N4) Verification: Verify the subarachnoid space (remove the needle stylet, test for flow of CSF).

N5) Retraction: Remove the needle from the tissue.

Each participant was allowed four practice procedures on the spine phantom: two punctures in the L3-4 space and two in the L4-5 space. Then, each participant performed two recorded trials in each of the L3-4 and L4-5 spaces.

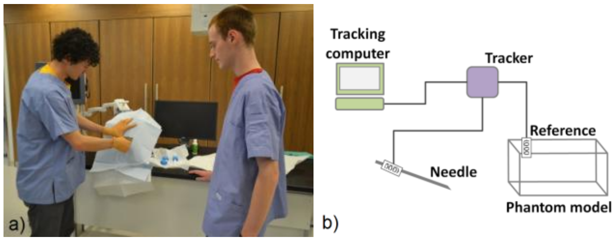

Fig. 3. a) Photograph of lumbar puncture setup. b) Schematic of lumbar puncture setup illustrating systems used in the experiment. Coil shapes indicate that the object was tracked electromagnetically.

Needle motions were involved in all tasks (N1-N5); thus, all tasks were used for workflow segmentation. Again, performing the above tasks in order constitutes the optimal workflow for the lumbar puncture. Users, however, were allowed to perform the tasks in any order and repeat tasks as necessary to achieve success.

Participants used a 19 gauge lumbar puncture needle (diameter 1.07mm). The needle and phantom were tracked using the Ascension TrakStar electromagnetic tracking system (Ascension Technology Corporation). The needle was tracked by a reusable 6DOF sensor inserted in the stylet; the phantom was tracked by an externally affixed 6DOF sensor (1.4mm root-mean-square accuracy for both).

The lumbar puncture procedures were divided into four groups based on amount of practice and whether the procedure was performed in the L3-4 or L4-5 space. Group sizes were: Unpracticed L3-4 12, Unpracticed L4-5 12, Practiced L3-4 12, Practiced L4-5 7. The leave-one-out cross validation method was performed separately for each group. Data from all groups were used to calculate accuracy statistics.

To test whether the optimal algorithm parameters from the ultrasound-guided epidural procedure are indeed optimal for other procedures, they were not adjusted from those calculated for the ultrasound-guided epidural procedure. By not adjusting the parameters we are able to gain a sense of how robust the algorithm is with respect to changing procedures.

F. *Simulated Data*

To further validate the proposed algorithm’s effectiveness for workflow segmentation of procedures with arbitrary order and repetition of tasks, simulated needle placement data was generated. Both the ultrasound-guided epidural and lumbar puncture procedures follow the same workflow when performed optimally: 1) find insertion point; 2) find insertion angle; 3) find insertion depth; 4) verify target; 5) retract needle. Thus, simulated data was generated for each task, based on predefined entry and target points for the procedure. The entry and target points were used to define needle displacements at task transitions, and a spline was calculated between these transition points to achieve a continuous trajectory. Additionally, Gaussian noise of amplitude 1.4mm (root-mean-square error for typical electromagnetic tracking systems) was added to each degree of freedom independently.

Two groups of simulated data were generated: one for which all procedures followed the optimal workflow and one where none of the procedures followed the optimal workflow. The order of tasks for the first group followed the optimal workflow (steps 1-5 described above). The order of tasks for the second group was randomly determined using a Markov process [23], where each physically possible task transition occurred with equal probability. The length of each task was chosen randomly from a normal distribution centered at the task’s average length over all collected ultrasound-guided epidurals and lumbar punctures. The leave-one-out cross validation method was performed separately for each group.

G. *Workflow Segmentation Accuracy Calculation*

The accuracy was defined as the proportion of time stamps for which the workflow segmentation produced the same task label as the ground-truth. The leave-one-out cross-validation method was used to verify the accuracy of the algorithm. Additionally, the training accuracy of the algorithm was calculated by training the algorithm using all data from the group and segmenting each procedure using this trained algorithm. High training accuracy and low testing accuracy indicates that the algorithm is overfitting the training data.

The ground-truth segmentation was determined manually by experts who visualized the recorded tool trajectories in 3D using VCR-style controls to manipulate playback and indicate the task transition times. While this technique is more accurate than real-time manual task segmentation, it is not 100% accurate due to the coupled nature of needle-based tasks. Thus, the ground-truth is not perfect.

To determine the accuracy of the ground-truth, blinded observers performed manual segmentations on a subsample of the recorded ultrasound-guided epidural procedures. Their segmentations were compared to the manual segmentations used as ground-truth, and this was used to calculate the manual segmentation consistency.

The manual segmentation consistency serves as a benchmark accuracy that is a practical upper limit on the accuracy of the automatic segmentation algorithm. Also, lower manual segmentation consistency leads to lower automatic segmentation accuracies. To quantify this, we compute the automatic segmentation accuracy as a proportion of the manual segmentation consistency. This estimates the accuracy the algorithm would have if the ground-truth was perfect.

H. *Task Transition Windows*

The observer is often unable to distinguish which task a user is performing at times near a task transition. The user may even perform two tasks simultaneously (i.e. position and rotate needle at the same time). In such instances, defining sharp task transition points may be inappropriate. In fact, most applications do not require accurate identification of task boundaries. When procedural workflows are modeled, the order of tasks has higher importance, and when real-time workflow instructions are generated for the current task, a short delay may be acceptable to the users.

The variances in the task transition identifications between manual segmentations from all observers were measured and used to calculate a window around each task transition point. Within these transition windows, the workflow segmentation produced by the algorithm was considered correct if it identified the task the user was performing as either of the two tasks involved in the transition. Using this technique, workflow segmentation accuracies were calculated for algorithm and the blinded observers. This accounts for the fact that tasks may be coupled near times of transition, but the algorithm must still identify which tasks are involved.

I. *Temporal Accuracy*

For application in a real-time feedback system, the algorithm’s temporal accuracy was evaluated as the standard deviation of the difference between the ground-truth segmentation and the automatic segmentation transition points. This identifies the time taken for the algorithm to recognize a transition in tasks, which we call the temporal accuracy. This measure was calculated using only task transitions that were correctly identified.

III. Results

A mean automatic segmentation accuracy of 81% was found for the ultrasound-guided epidural, and an accuracy of 82% for the lumbar puncture. The distributions of segmentation accuracies (Fig. 4), however, exhibit large standard deviations (Table 1). The confusion matrices for the ultrasound-guided epidural and lumbar puncture procedures demonstrate the algorithm’s accuracy for each task individually (Tables 2 and 3).

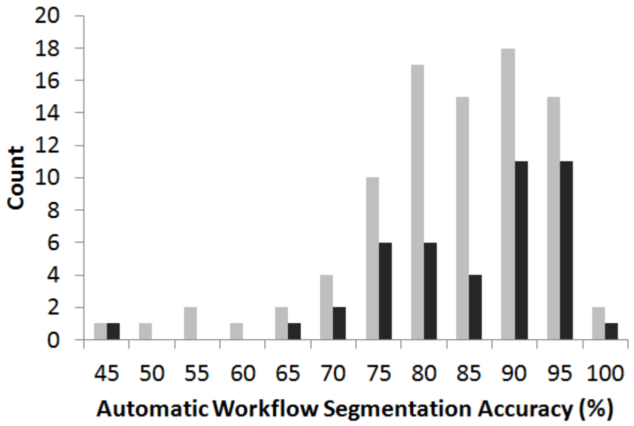

Fig. 4. Automatic workflow segmentation accuracy histogram for the ultrasound-guided epidural (light) and lumbar puncture (dark) procedures. Bins are indicated by their upper bound.

Of the 88 ultrasound-guided epidurals, 45% did not follow the optimal workflow, and in this case the accuracy of the algorithm was 79.9%, compared to 81.1% when the optimal workflow was followed. Of the 43 lumbar punctures, only 19% did not follow the optimal workflow, resulting in a mean mean workflow segmentation accuracy of 71.5%, compared to 84.9% for the procedures for which the workflow was optimal.

Using the proposed algorithm without the orthogonal transformation step yielded accuracies of 77% for the ultrasound-guided epidural and 80% for the lumbar puncture. Both decreases are statistically significant by paired *t*-test ($\alpha = 0.05$), demonstrating that orthogonal transformation improves workflow segmentation.

Statistic	Ultrasound-Guided Epidural	Lumbar Puncture
Count	88	43
Mean (%)	80.6	82.4
Standard Deviation (%)	10.6	10.5
Median (%)	82.4	85.2
Minimum (%)	44.6	42.4
Maximum (%)	98.3	96.6

Table 1. Automatic workflow segmentation accuracy statistics for the ultrasound-guided epidural and lumbar puncture procedures.

The average workflow segmentation accuracy for each group was: Epidural Novice Left (35 procedures) 81.6%, Epidural Novice Right (32 procedures) 80.2%, Epidural Expert Left (10 procedures) 75.3%, Epidural Expert Right (11 procedures) 83.1%, Lumbar Unpracticed L3-4 (12 procedures) 82.9%, Lumbar Unpracticed L4-5 (12 procedures) 82.3%, Lumbar Practiced L3-4 (12 procedures) 80.8%, Lumbar Practiced L4-5 (7 procedures) 84.3%.

		Automatically Segmented Task					Mean
		N1	N2	N3	N4	N5	Length (s)
Ground-Truth Task	N1	88.0	10.6	0.2	0.2	0.9	6.4
	N2	8.2	77.4	12.3	0.0	2.1	1.5
	N3	0.3	7.6	78.3	6.5	7.3	4.4
	N4	0.0	0.0	12.9	83.5	3.6	3.4
	N5	0.1	2.2	19.1	11.1	67.5	1.6

Table 2. Confusion matrix comparing the automatic workflow segmentations with the ground-truth segmentations for the ultrasound-guided epidural procedure. Values indicate the percentage of timestamps the actual task was segmented as belonging to each predicted task.

		Automatically Segmented Task					Mean
		N1	N2	N3	N4	N5	Length (s)
Ground-Truth Task	N1	83.6	5.8	0.3	4.6	5.7	6.3
	N2	24.1	64.9	9.9	0.0	1.1	2.5
	N3	0.3	3.5	86.3	8.2	1.8	9.6
	N4	38.0	0.9	11.8	49.0	0.3	6.2
	N5	42.2	0.6	36.6	14.3	6.3	6.2

Table 3. Confusion matrix comparing the automatic workflow segmentations with the ground-truth segmentations for the lumbar puncture procedure. Values indicate the percentage of timestamps the actual task was segmented as belonging to each predicted task.

The training accuracy associated with the algorithm was 90% for the ultrasound-guided epidural and 93% for the lumbar puncture. This demonstrates that the algorithm is not overfitting the training data. In addition, the algorithm’s accuracy as a function of training set size was measured (Fig. 5). As expected, the accuracy improves as the training set sizes increases, but the algorithm still exhibits mean accuracy over 75% for the smallest training set sizes.

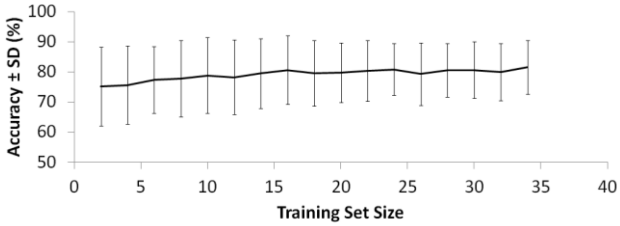

Fig. 5. Automatic workflow segmentation accuracy as a function of training set size for the ultrasound guided-epidural procedure.

Example workflow segmentations for both the ultrasound-guided epidural and lumbar puncture procedures with median accuracy are shown below (Fig. 6). This figure illustrates that most errors are due to misidentification of transition times, rather than incorrect task classification.

Given the manual segmentation consistency, the automatic segmentation algorithm was 93% accurate. In particular, for the subsample manually segmented by the blinded observers, the manual segmentation consistency was 84% and the mean automatic segmentation accuracy was 79% (Table 4). Using Cohen’s d statistic, the effect size between the manually and automatically segmented procedures was medium (0.5).

Choosing larger task transition windows not only increased the mean segmentation accuracies, but also decreased the standard deviations (Table 4), suggesting that much of the variation in accuracies is due to task misidentification near task transitions.

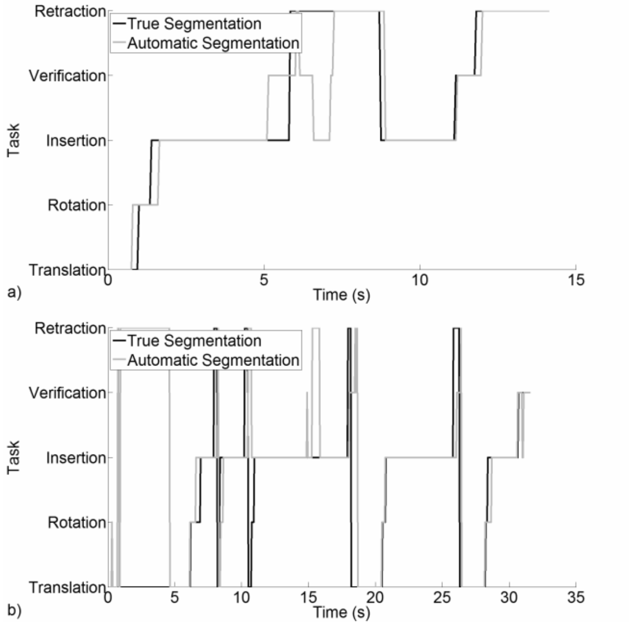

Fig. 6. Automatic workflow segmentation (light) versus ground-truth workflow segmentation (dark) for the non-optimal procedure with the median automatic workflow segmentation accuracy for a) the ultrasound-guided epidural procedure and b) lumbar puncture procedure.

For the both the ultrasound-guided epidural and the lumbar puncture procedures, the temporal accuracy was 1.2s. In constrast, the average task length was 3.4 ± 2.1s for the ultrasound-guided epdiural and 6.2 ± 2.5s for the lumbar puncture. Interestingly, in many instances the algorithm identified a task transition prior to the corresponding transition in the ground-truth segmentation (Fig. 6).

Window Size	Manual Segmentation Consistency (Mean ± SD %)	Algorithm Accuracy (Mean ± SD %)
± 0σ = 0.0s	84.4 (± 7.8)	78.5 (± 12.5)
± 1σ = 0.3s	89.2 (± 6.7)	83.1 (± 11.0)
± 2σ = 0.6s	93.1 (± 5.2)	85.8 (± 10.2)
± 3σ = 0.9s	94.9 (± 4.1)	87.8 (± 9.7)

Table 4. Automatic workflow segmentation accuracy of the proposed algorithm versus manual workflow segmentation accuracies from blinded observers over the subsample of procedures segmented by each blinded observer for varying window sizes.

For the simulated data, the mean segmentation accuracies for procedures following the optimal workflow and those not were 77.5% (standard deviation 15.7%) and 76.2% (standard deviation 11.8%) respectively. Using Cohen's d statistic, the effect size between the optimal and non-optimal groups of simulated data was small (<0.1).

IV. Discussion

The results shown here are not directly comparable to the results in the literature because there are different constraints on the algorithms. The accuracies reported in the literature serve as benchmarks against which to compare our algorithm. Because the proposed algorithm produces similar results and is subject to more constraints (i.e. must segment procedures with task sequences that possibly do not appear in the training data), this demonstrates that it is sufficiently accurate for use in the proposed computer-assisted needle placement training system. Additionally, our ground-truth segmentations are only consistent to within 84%, which is lower than the ground-truth consistency in most other studies. Many results from the literature are from robotic manipulator or laparoscopic procedures, which are not subject to human hand noise. This leads to less coupling between consecutive tasks, and thus, these procedures can be manually segmented with greater precision [4]-[10], [12]-[13].

Our algorithm achieved 93% accuracy relative to the manual segmentation consistency. This is an estimate of the accuracy the algorithm would achieve if the ground-truth was perfect. We suggest that the difference between the manual segmentations and automatic segmentations may be practically insignificant for our application.

The low manual segmentation consistency is partially due to the coupled nature of the tasks and partially to the sparse 3D visualization observers used to manually segment the procedures. Discrete workflow segmentation is required for providing real-time instruction. Thus, to improve manual segmentation consistency, tool tracking information could be augmented with synchronized video to assist observers in identifying task transition points. This multi-stream setup, however, requires temporally calibrated data recording and was not available for our experiments.

The algorithm's temporal accuracy (1.2s) is adequate for application in a computer-assisted needle placement training system because it is significantly shorter than the length of any task in the workflow. Additionally, the temporal accuracy is expected to improve with improved manual segmentation consistency. Many of the less accurate segmentations, which are often due to poor temporal accuracy, will also improve. The algorithm identifies task transitions both too early and too late. This is expected because the algorithm produces a task label at every time at which tracking data is recorded, and the motion characterizing the beginning of a task may appear before or after the ground-truth task transition. The temporal accuracy reported here is unaffected by time delays in the tool tracking system since the analysis was performed offline.

Since a complete validation study of the simulated data was not performed, the workflow segmentation accuracies for the simulated data cannot be compared to experimental data. The relative accuracy of the optimal and non-optimal groups, however, is similar (by the test of effect size). This shows that the algorithm is effective at both segmenting procedures that do follow the optimal workflow, as well as procedures that do not. This further validates the claim that this algorithm works when the order of tasks is not known beforehand.

The analysis of our workflow segmentation algorithm simulates a real-time scenario, but does not actually provide feedback to the user in real-time. In a true real-time scenario, the user would react and adjust according to the provided feedback, and the workflow segmentation algorithm should adjust its feedback accordingly. This may introduce motions which would not be known to the algorithm during the training phases. The analysis performed here does not show whether the proposed algorithm is robust to user adjustment. The algorithm must be implemented in a real-time scenario to test its validity when the user adjusts to feedback.

Finally, the ultrasound-guided epidural procedure and the lumbar puncture procedure are both needle-based spinal procedures that follow the same paradigm: 1) find insertion point; 2) find insertion angle; 3) find insertion depth; 4) verify target; 5) retract needle. Since the analyzed procedures are so similar, the algorithm must be tested using a paradigmatically different procedure to further verify its general validity.

V. Conclusion

The proposed workflow segmentation algorithm performs with accuracy similar to accuracies reported in the literature. The algorithm is 93% as accurate as manual segmentations, and the effect size is medium (0.5). Additionally, it was validated on both the ultrasound-guided epidural and lumbar puncture datasets using different hardware setups. Its temporal accuracy was 1.2s, which is acceptable for our application.

Our findings demonstrate that the algorithm is applicable to computer-assisted needle placement training systems for following a user in a procedural workflow in real-time to provide instructions. Using such training systems to complement expert supervision has potential to improve medical training and clinician competency.

Acknowledgment

We thank Dr. J. Brookes of LHSC for his assistance with the ultrasound-guided epidural data collection.

References

[1] D. A. Rogers, G. Regehr, T. R. Howdieshell, K. A. Yeh, E. Palm, “The impact of external feedback on computer-assisted learning for surgical technical skill training,” American Journal of Surgery, vol. 179, no. 4, pp 341-343, Apr. 2000.

[2] M. C. Porte, G. Xeroulis, R. K. Reznick, A. Dubrowski, “Verbal feedback from an expert is more effective than self-accessed feedback about motion efficiency in learning new surgical skills,” American Journal of Surgery, vol. 193, no. 1, pp 105-110, Jan. 2007.

[3] R. Aggarwal, T. P. Grantcharov, A. Darzi, “ Framework for systematic training and assessment of technical skills,” Journal of the American College of Surgeons, vol. 204, no. 4, pp 697-705, Apr. 2007.

[4] A. Castellani, D. Botturi, M. Bicego, P. Fiorini, “Hybrid HMM/SVM model for the analysis and segmentation of teleoperation tasks,” Robotics and Automation, vol. 3, pp. 2918-2923, Apr. 2004.

[5] S. A. Ahmadi, T. Sielhorst, R. Stauder, M. Horn, H. Fuessner, N. Navab, “Recovery of surgical workflow without explicit models,” Medical Image Computing and Computer Assisted Interventions, vol. 9, pp 420-428, Oct. 2006.

[6] F. Lalys, L. Riffaud, X. Morandi, P. Jannin, “Surgical phases detection from microscope videos by combining SVM and HMM,” International Medical Image Computing and Computer Assisted Interventions Workshop, pp. 54-62, 2010.

[7] N. Padoy, G. D. Hager, “Human-machine collaborative surgery using learned models,” Proceedings of the IEEE International Conference on Robotics and Automation, pp. 5285-5292, May. 2011.

[8] N. Padoy, T. Blum, A. Ahmadi, H. Feussner, M. O. Berger, N. Navab, “Statistical modeling and recognition of surgical workflow,” Medical Image Analysis, vol. 16, pp. 632-641, Apr. 2012.

[9] C. S. Hundtofte, G. D. Hager, A. M. Okamura, “Building a task language for segmentation and recognition of user input to cooperative manipulation systems,” 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 225-230, Mar. 2002.

[10] H. C. Lin, I. Shafran, D. Yuh, G. D. Hager, “Towards automatic skill evaluation: detection and segmentation of robot-assisted surgical motions,” Computer Aided Surgery, vol. 11, no. 5, pp. 220-230, Sep. 2006.

[11] L. Tao, E. Elhamifar, S. Khudanpur, G. D. Hager, R. Vidal, “Sparse hidden Markov models for surgical gesture classification and skill evaluation,” Information Processing in Computer Assisted Interventions, vol. 7330, pp. 167-177, Jun. 2012.

[12] C. E. Reiley, H. C. Lin, B. Varadarajan, B. Vagvolgyi, S. Khundanpur, D. D. Yuh, G. D. Hager, “Automatic recognition of surgical motions using statistical modeling for capturing variability,” Studies in Health Technology and Informatics, vol. 132, pp. 396-401, Jan. 2008.

[13] B. Varadarajan, C. Reiley, H. Lin, S. Khudanpur, G. Hager, “Data-derived models for segmentation with application to surgical assessment and training,” Medical Image Computing and Computer Assisted Interventions, vol. 12, no. 1, pp. 426-434, Sep. 2009.

[14] N. Ahmidi, G. D. Hager, L. Ishii, G. Fichtinger, G. L. Gallia, M. Ishii, “Surgical task and skill classification from eye tracking and tool motion in minimally invasive surgery,” Medical Image Computing and Computer Assisted Interventions, vol. 13, no. 3, pp. 295-302, Sep. 2010.

[15] M. Li, A. M. Okamura, “Recognition of operator motions for real-time assistance using virtual features,” 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 125-131, Mar. 2003.

[16] D. Aarno, D. Kragic, “Layered HMM for motion intention recognition,” Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp. 5130-5135, Oct. 2006.

[17] A. James, D. Vieira, B. Lo, A. Darzi, G. Z. Yang, “Eye-gaze driven surgical workflow segmentation,” Medical Image Computing and Computer Assisted Interventions, vol. 10, pp. 110-117, Oct. 2007.

[18] B. Haro, L. Zapella, R. Vidal, “Surgical gesture classifications from video data,” Medical Image Computing and Computer Assisted Interventions, vol. 7510, pp. 34-41, Oct. 2012.

[19] O. Golubitsky, S. M. Watt, “Online stroke modeling for handwriting recognition,” Proceedings of the 18th International Conference on Computer Science and Software Engineering, pp. 1705-1713, Dec. 2008.

[20] G. Carballo, R. Alvarez-Nodarse, J. S. Dehesa, “Chebychev polynomials in a speech recognition model,” Applied Mathematics Letters, vol. 14, no. 5, pp. 581-585.

[21] T. Cserhati, Z. Illes, “Comparison of two principal component analysis methods to evaluate reversed-phase retention data,” Journal of Pharmaceutical & Biomedical Analysis, vol. 9, no. 9, pp. 685-691, 1991.

[22] A. Likas, N. Vlassis, J. Verbeek, “The global k-means clustering algorithm,” Pattern Recognition, vol. 36, no. 2, pp. 451-461, Feb. 2003.

[23] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speed recognition,” Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286, Feb. 1989.

[24] M. K. Karmakar, X. Li, M. H. Ho, W. H. Kwok, P. T. Chui, “Real-time ultrasound-guided paramediam access: evaluation of a novel in-plane technique,” British Journal of Anaesthesia, vol. 102, no. 6, pp. 845-854, Apr. 2009.

Appendix

The following parameters were used in the workflow segmentation algorithm to produce the reported results:

1) *Gaussian Filter:* $\sigma = 0.22$
2) *Orthogonal Transformation:* $Order = 3, \Delta t = 0.30s$
3) *Principal Component Analysis:* $Components = 6$
4) *K-means Clustering:* $k = 700$
5) *Markov Modelling:* $\delta = 0.2$