Robotic Scrub Technician

NSF Grant Number DMI-0319860

Phase I Technical Report

Research Summary

This SBIR Phase I project addresses a specific and realistic opportunity to use an automated robotic device to improve efficiency and reduce costs in the operating room (OR). Our *robotic scrub technician*, shown in Figure 1, will deliver instruments to the surgeon, retrieve them when they are no longer needed, and keep track of the instrument count. This technology will allow hospitals to reallocate scarce personnel resources to clinical areas that are currently understaffed. The Phase I research objective was to determine the feasibility of using artificial intelligence and statistical techniques to predict the surgeon's instrument requests. This will allow the robot to keep one step ahead of the surgeon, much like an experienced human scrub technician. With this predictive capability the robot can decide how to organize its limited storage space to keep those instruments likely to be needed soon closer to the surgical field. This will greatly improve the responsiveness of the device, a critical factor in achieving clinical acceptance and ultimately commercial viability.

We conducted a survey of over 50 surgical procedures, cataloging over 2000 individual instrument requests. We then used this time series data to train and evaluate various prediction algorithms. At each point in the surgical procedure, these algorithms produce a *prediction score* for every known instrument type. These scores are based on a training set of previous cases and the list of recent instrument requests from the current case. To measure the prediction accuracy of our algorithms we first trained them on one set of surgical procedures. We then ran simulations of the remaining procedures using the prediction scores to decide which instruments to move forward and which to move back. The right moves will keep the requisite instruments close to the surgeon, minimizing our primary metric, average instrument delivery time.

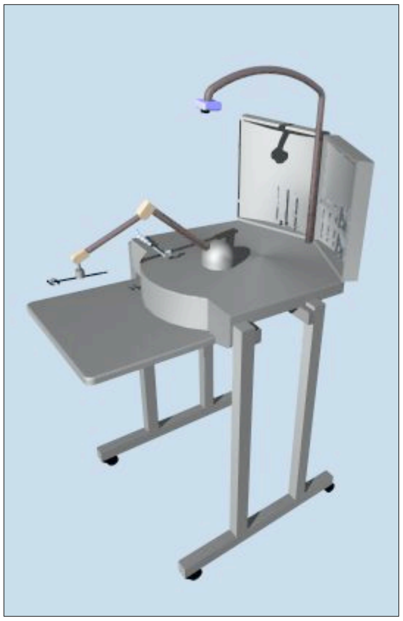


Figure 1. The robotic scrub technician.

We found that our best algorithm, a *modified N-gram sequence matcher*, could recommend favorable moves 88% of the time. This resulted in a 52% decrease in the average instrument delivery time as compared to our baseline. Moreover, when trained on each surgical procedure from the data set in chronological order, the algorithm's performance improved over time – trending away from the baseline. This indicates that if the robot had been fielded in the test facility and had been used on these procedures, it would have learned from each case making better predictions as time went by.

These results will provide a crucial framework for the broader task of creating a *cognitive architecture* for the robot. This architecture will control the robot's behavior and enable it to adapt to the ever-changing environment in the OR. A reliable instrument prediction capability will allow the robot to exhibit *proactive* behavior, as opposed to merely *reacting* to explicit commands. This is a fundamental distinction, separating traditional devices from truly autonomous robots. We believe that this autonomy is an essential advance for robotics in the OR. The results of this Phase I research indicate that reliable instrument prediction can be achieved. With the addition of a cognitive architecture allowing the robot to further reason and plan its behavior, we believe we will have a firm technological foundation for an effective, efficient robotic scrub technician.

Data Survey

We have compiled a large database of instrument request time series from actual surgical procedures. For each procedure we captured the timestamped list of verbal instrument requests from either the primary surgeon or an assistant. This data was used to train our prediction algorithms and to test their accuracy. The data collected in this survey is not intended to be generalizable across geographic regions or other surgical procedures. Our goal was to collect a large, site-specific dataset representing a realistic field environment in which the robotic scrub technician might be deployed.

We cataloged 45 hours of surgery over 4 months, spanning 51 cases. This resulted in 2,105 instrument requests for 24 distinct instrument types. Cases performed by 8 different primary surgeons, assisted by 8 different scrub nurses or technicians, were recorded.

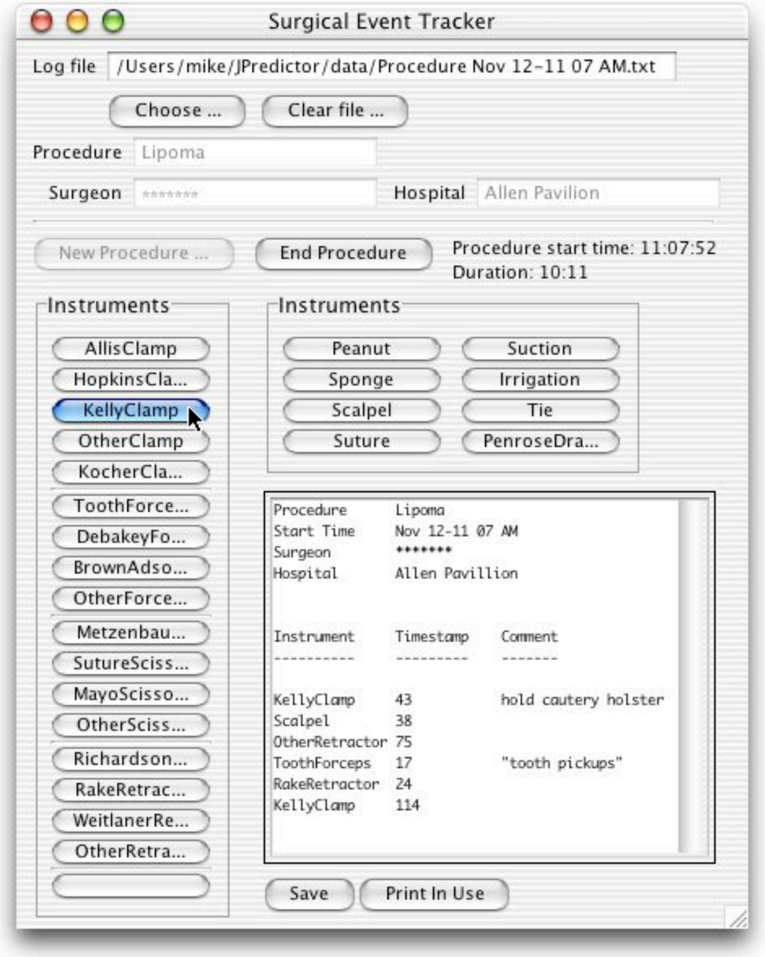
Prototype Surgical Procedures

The data survey was conducted using as case prototypes 1) excision of subcutaneous lesions such as lipoma or other masses including lymph nodes of trunk, arms, or legs; and 2) repair of inguinal or umbilical hernias. These procedures were chosen as good representations of cases performed using what is commonly called a minor instrument set. The minor instrument set usually contains around 108 instances of 42 different instrument types and is used for most general surgical procedures. Largely for business reasons, our goal is to introduce the first version of the robotic scrub technician supporting only the minor instrument set. This will give us a large market base for the first release and a clear, high margin upgrade path – supporting the slightly larger major instrument set through a software upgrade – for the next version.

Data Collection Technique

The survey was conducted so as to minimize selection bias for the surgical cases. No factors were used to discriminate which procedures would be recorded aside from these preconditions: that the surgery was performed at the test facility, the Allen Pavilion of the New York-Presbyterian Hospital; that it was one of our two prototype procedures; and that permission could be obtained from the patient and the surgeon. Aside from these preconditions, cases were selected as they appeared on the surgical schedule.

A custom designed software application called the *Surgical Event Tracker*, shown in Figure 2, was used for data collection. The test administrator used a laptop computer running this application in the OR to record all instrument requests during the procedure. The software first logged the type of procedure, the surgeon, and the facility. Then, as the surgery progressed, the test administrator pressed the button for each requested instrument. This would log the instrument type, a timestamp, and comment if supplied. We used the comment field to note any additional interesting information, such as instrument name synonyms and shifts in the phase of the procedure.



Log file /Users/mike/JPredictor/data/Procedure Nov 12-11 07 AM.txt

Procedure Lipoma

Surgeon ******* Hospital Allen Pavilion

Procedure start time: 11:07:52
Duration: 10:11

```
Procedure     Lipoma
Start Time    Nov 12-11 07 AM
Surgeon       *******
Hospital      Allen Pavillion


Instrument    Timestamp   Comment
----------    ---------   -------


KellyClamp    43          hold cautery holster
Scalpel       38
OtherRetractor 75
ToothForceps  17          "tooth pickups"
RakeRetractor 24
KellyClamp    114
```

Figure 2. Screen capture showing the Surgical Event Tracker, a software application used to record instrument requests during surgical procedures.

Baseline Prediction Analysis

Our baseline is derived from the overall instrument usage histogram shown in Figure 3. This histogram shows the occurrence counts for each instrument type across all 51 surgical procedures in the data survey. The most commonly requested instrument, the Hopkins clamp, occurred 331 times, or 16% of the time. Thus, as a first order prediction baseline, we can see that one can achieve a 16% accuracy rate by simply predicting the Hopkins clamp at all times.

The baseline we chose for our prediction analysis is more realistic. As a baseline, we assume that one instance of each of the *top 12* instruments from the histogram is kept close to the surgeon for speedy delivery. We chose 12 because that is the number of instruments that can fit on the robot's primary sterile tray called the Mayo stand. Statistically, this accounts a very impressive 87% of the instrument requests. Our goal was to produce a prediction algorithm that can reduce average instrument delivery time as compared to this baseline.

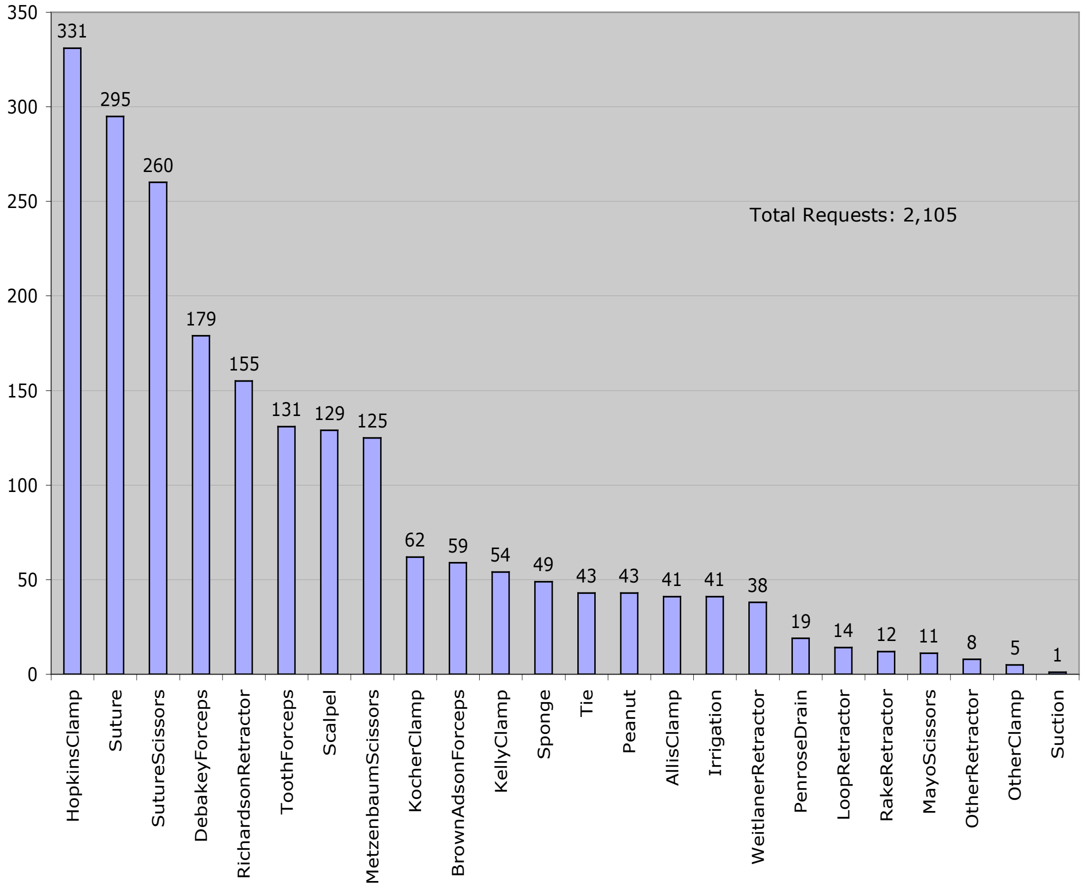


Figure 3. Overall instrument usage histogram, showing the number of times each type of instrument was requested in the 51 surgical procedures surveyed.

Prediction Goals

The primary metric for our prediction capability is instrument delivery time. This is the time from the verbal request to the handoff of the instrument. Surgeons are particularly sensitive to this delay and will look unfavorably on the robot if it isn't quick enough. While the speed of the robotic arm is an obvious factor, safety concerns impose clear limits on the maximum arm speed. So we will rely on instrument prediction to minimize delivery time more intelligently. In particular, we want to gain insight into the probable near-term future of the surgical procedure so the robot can better marshal the instruments in its control. In the best-case scenario, the robot can have the next instrument ready in its gripper for the surgeon. But there are other advantageous scenarios to consider as well. There are a number of ways the robot can minimize future delivery time for instruments likely to be needed soon. Our prediction algorithm should ideally help us take advantage of all these strategies.

Instrument Caches

An instrument cache is a sterile surface used by the robot to store instruments. Instrument caches are designated by levels indicating their proximity to the surgeon, where a first level cache is closest and thus requires the shortest delivery time. The instrument caches are the back tray (3rd level), the staging zone (2nd level), and the Mayo Stand (1st level). A fourth area, called the transfer zone, might be thought of as a 0th level cache. The surgeon returns instruments to this area when they are no longer needed. The robot can also lay instruments here for the surgeon to pick up without having to ask. The locations of these instrument caches are shown in Figure 4.

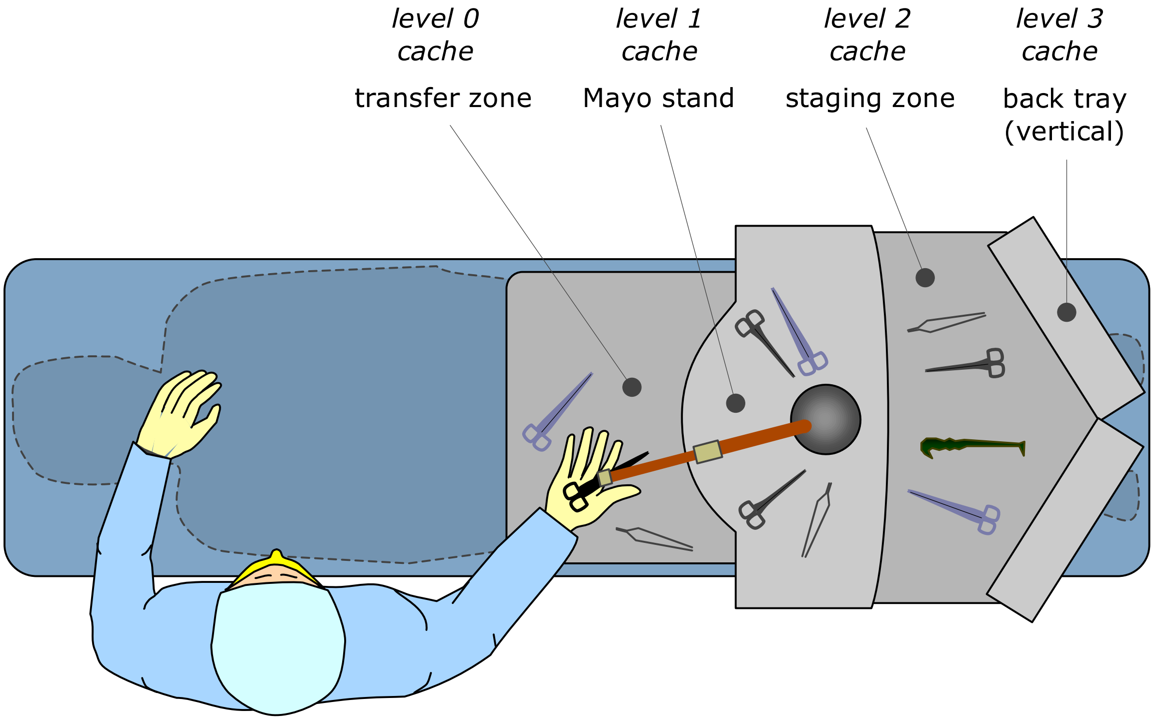


Figure 4. An overhead view of the robotic scrub technician (shown without sterile draping for clarity). Note the various surfaces on which the robot stores instruments. Some are close to the surgeon for quick delivery, while others require the robot to spin around taking more time.

As the procedure progresses the robot can move instruments from one cache to another, trying to keep the most useful instruments close to the surgeon. The space for instruments is limited however, so tradeoffs must be made. In our design there is room for up to 4 instruments at a time on the transfer zone and 12 each on the Mayo stand and staging zone. The back tray holds all 108 instruments in the minor set, but it is only used as a source – instruments are never returned to the back tray once they have been removed. The problem, not dissimilar from that of minimizing memory access times in computer systems, is determining which moves to make and when to make them. From our data survey results we can see that the average time between requests is 76 seconds, more than enough time for the robot to make several instrument moves.

Direction	Move	From		To	*Notes*
forward	F3-2	back tray	⇒	staging zone	All instruments start off on the back tray. Instruments are never returned to the back tray however. This is analogous to a *read-only* memory cache.
	F2-1	staging zone	⇒	Mayo stand	The staging zone is nominally for instruments not needed in the current stage of the procedure. They are brought onto the Mayo stand just before they are needed.
	F1-0	Mayo stand	⇒	transfer zone	This is a way to offer the surgeon an instrument that may or may not be needed. The surgeon is encouraged to pick things up off the transfer zone, but not the other caches.
	F1-G	Mayo stand	⇒	gripper	If we are confident we know the next instrument the surgeon will need, we can hold it in the gripper. This is risky, however, in that we must put the instrument back down if the prediction is incorrect.
backward	B0-2	transfer zone	⇒	staging zone	The robot must periodically police up instruments from the transfer zone to minimize clutter and conserve space. Unneeded instruments should be brought back first.
	B1-2	Mayo stand	⇒	staging zone	If there is an instrument on the Mayo stand that has not been used recently and is not likely to be needed soon, we can move it to the staging zone to free the space.

Table 1. Useful inter-cache instrument moves.

Table 1 shows 6 inter-cache moves we have identified, either to move useful instruments closer to the surgeon or to move less useful instruments out of the way. Move F2-1, for example, is a forward move from the level 2 cache, the staging zone, to the level 1 cache, the Mayo stand. Move B1-2 is the reverse operation, going backward from the Mayo stand to the staging zone. Reliable instrument prediction will be the basis on which we will plan these moves.

Inter-cache Move Rules

Our predictions identify not only those instruments that will be needed soon, but also those that are *not* likely to be needed. This allows us to choose among the forward and backward moves wisely. Thus our prediction algorithm ranks *all* instrument types from most likely to least likely. More precisely, we compute a *prediction score* for every instrument type. The higher the prediction score, the more likely that instrument will be needed soon. We then divide the instrument types into categories according to their prediction scores. Empirically we have determined that 3 categories work well. The top 8 scorers are grouped into category 1, the next 3 are in category 2, and the rest are put into category 3.

With these prediction scores as a basis, we have developed a rule-based system to plan inter-cache instrument moves as the surgical procedure progresses. In between requests we evaluate a list of rules to determine which moves, if any, we should make. These are heuristic rules, designed to minimize the average instrument delivery time.

Rule 1 *if* $(|\mathtt{Cache}_2| < 12) \wedge (\exists\, i : I \mid i \in \mathtt{Cache}_0 \wedge i \in \mathtt{Cat}_3)$

then $\mathtt{move}(i,\ \mathtt{B}_{0\text{-}2})$

Rule 2 *if* $(|\mathtt{Cache}_0| < 4) \wedge (\exists\, i : I \mid i \in \mathtt{Cache}_1 \wedge i \in \mathtt{Cat}_1)$

then $\mathtt{move}(i,\ \mathtt{F}_{1\text{-}0})$

Rule 3 *if* $(|\mathtt{Cache}_1| < 12) \wedge (\exists\, i : I \mid i \in \mathtt{Cache}_2 \wedge i \in \mathtt{Cat}_2)$

then $\mathtt{move}(i,\ \mathtt{F}_{2\text{-}1})$

Figure 5. An example rule set. I is the set of all instruments. $\mathtt{Cat}_x$ is the set of all category x instruments and $\mathtt{Cache}_x$ is the set of all instruments in cache x.

Figure 5 shows a simplified example rule set. Rule 1 states that if there is a category 3 (low scoring) instrument on the transfer zone (the 0th level cache) and there is space available on the staging zone (the 2nd level cache), that instrument should be moved there. Rule 2 tries to move high scoring instruments from the Mayo stand to the transfer zone, while rule 3 tries to move instruments from the middle category forward from the staging zone to the Mayo stand. The order of the rules is important. We try rule 1 first because it might free up some space on the transfer zone that we can use in rule 2. Similarly, rule 2 might free up space on the Mayo stand that rule 3 can utilize.

A small number of these simple rules of thumb result in some remarkably complex behavior. The robot moves these instruments around on its own, without any human guidance. When the surgical procedure begins, the OR staff will wheel the robot in and open the back tray of sterile instruments. From there the robot will take over, initially unpacking the instruments the surgeon will need, then shuffling them back and forth throughout procedure, often having the next instrument waiting on the transfer zone for the surgeon. Aside from minimizing delivery times, this autonomous behavior will greatly improve peoples' perception of the robot and its ability to do its job.

Modified N-gram Sequence Matching

We evaluated several candidate prediction techniques of varying complexity. Our final approach synthesizes N-gram sequence matching, higher-order Markov modelling, and the rule-based system described above. This approach seems very well suited to the domain of surgical instrument requests. From our survey data we observed many relatively small patterns of requests that occurred repeatedly. Within these patterns there was a fair degree of order flexibility. These observations led us to an algorithm based on a combination of Markov-style occurance counts and N-gram sequence matching.

Training

During training we adjust a three-dimensional array of occurance counts. If the set of all known instrument types is I and maximum sequence length we are considering is N, this occurance count array will be $|I| \times |I| \times N$. Then for each training sequence of ordered instrument requests R, we adjust the occurance count array, C, as follows.

$$\text{for all } i \in I,\ j \in I,\ k \in \{1{:}N\}$$

$$C_{i,j,k} = \sum_{r=1}^{|R|} \begin{cases} 1, \text{ if } (R_r = i) \wedge (R_{r-k} = j) \\ 0, \text{ otherwise} \end{cases} \tag{1}$$

Thus each element in the occurance count array is the number of times instrument i appeared k requests after instrument j. Table 2 shows an example of a trained occurance count array. In this example I = {a, b, c, d, e, f} and N is 5. In the k=2 subtable, when j=c and i=b we see a count of 3 indicating that the sequence c∗b (a c followed by some symbol, followed by a b) occurred 3 times in the training sequences. Looking at the input sequences we can see those 3 occurrences: cab in the first training sequence, ccb in the second, and another cab in the third.

Alphabet: abcdef
Training sequences: decab
ccbaedb
bacab

j∗∗∗∗i *k*=5						j∗∗∗i *k*=4						j∗∗i *k*=3						j∗i *k*=2						ji *k*=1						
j						*j*						*j*						*j*						*j*						
a	b	c	d	e	f	a	b	c	d	e	f	a	b	c	d	e	f	a	b	c	d	e	f	a	b	c	d	e	f	*i*
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1	0	0	2	2	0	0	0	a
0	0	1	0	0	0	0	2	0	0	1	0	2	0	0	0	1	0	0	0	3	0	1	0	2	0	1	1	0	0	b
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	1	0	c
0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	d
0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	e
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f

Table 2. An example occurrence count array for a small alphabet. There are 6 symbols in the alphabet and N is 5. Therefore, the count array is 6 × 6 × 5 with 180 elements.

This training scheme has several advantages. First it's quite simple, making it easy to implement and computationally inexpensive. Second, the size of the database is independent of the amount of training performed. The first version of our robot will supprt 42 instrument types. With N=5 and using standard 4 byte integers for the counts, we have a very manageable fixed-sized 34Kb database. The third advantage is that it captures sequence ocurrance information without strict ordering requirements and can represent sequences with occasional variations. We see examples of variations from our data survey where one instrument is occasionally substituted for another in a recurring pattern.

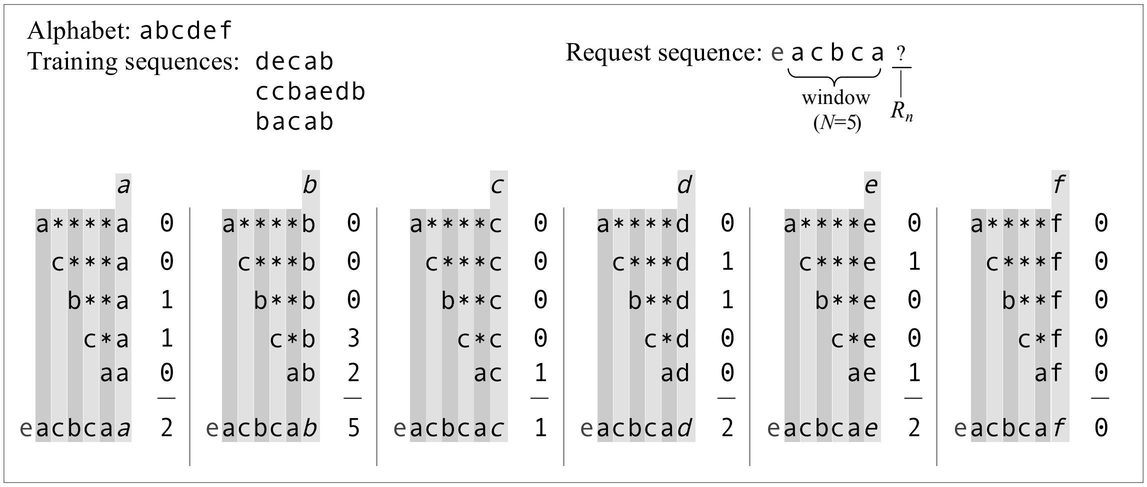
Making Predictions

Once trained, we can use the occurrance count array to compute prediction scores for new sequences. If we are processing an instrument request sequence, R, and the last request was R_{n-1}, we can compute a prediction score S_i for each instrument i in the instrument set as shown in equation (2). The prediction score S_i is the relative likelihood that the next request, R_n, will be the instrument i.

Given the sequence from R_1 to R_{n-1}
for each $i \in I$

$$S_i = \sum_{k=1}^{N} C_{i,\, R_{n-k},\, k} \tag{2}$$

Table 3 revisits the earlier example so demonstrate how prediction scores are calculated. Here we have encountered the request sequence eacbca and we wish to make predictions for the next request, R_n. Since N is 5, we use a window of the last 5 symbols in the sequence for matching. We compute scores for each symbol i, indicating the likelihood that R_n will be i. For each k from 1 to 5 we sum the occurance count from the database of R_{n-k} coming k requests before i. In this case the highest scorer is b with a score of 5. In our training set c∗b matched 3 times and ab matched twice, giving us a total of 5.



Alphabet: abcdef
Training sequences: decab
ccbaedb
bacab

Request sequence: $\mathtt{e}\underbrace{\mathtt{a\,c\,b\,c\,a}}_{\text{window }(N=5)}\,\underset{R_n}{?}$

a		*b*		*c*		*d*		*e*		*f*	
a****a	0	a****b	0	a****c	0	a****d	0	a****e	0	a****f	0
c***a	0	c***b	0	c***c	0	c***d	1	c***e	1	c***f	0
b**a	1	b**b	0	b**c	0	b**d	1	b**e	0	b**f	0
c*a	1	c*b	3	c*c	0	c*d	0	c*e	0	c*f	0
aa	0	ab	2	ac	1	ad	0	ae	1	af	0
	—		—		—		—		—		—
eacbca*a*	2	eacbca*b*	5	eacbca*c*	1	eacbca*d*	2	eacbca*e*	2	eacbca*f*	0

Table 3. An example set of prediction scores based on the trained database from Table 2. The request sequence thus far is eacbca. Prediction scores are computed for each of the 6 symbols in the alphabet indicating the likelihood that that symbol will be next. In this example, b has the highest score of 5, so we would predict that R_n will be b.

Predicting the first request, R_1, is a special case since our sequence is of length zero at that point and we have nothing to match against. To make this prediction we keep a *lead-off count* for each instrument – the number of times that instrument has been the first one requested. We have found that we can then start using the full algorithm with excellent results for R_2 onward.

It is important to note that these prediction scores are not probabilities. They *are* based on the relative frequencies of one instrument occurring *k* requests after another, which led us to our hypothesis that they are a reasonable approximation of the corresponding indefinite probabilities of those instruments repeating in a similar pattern in the future. We tested this hypothesis by experimentation and, for our data survey, this does indeed seem to be the case. As another concept validation test, we trained our modified N-gram sequence matcher on the first 2 sentences of Lincoln's Gettysburg Address. The algorithm was then able to correctly predict the next letter in the remainder of the speech approximately 25% of the time with *N* equal 4.

Results

In our most complete test, the modified N-gram sequence matcher was able to reduce average instrument delivery time by 52% as compared to our baseline. That baseline strategy is to keep one instance of each of the top 12 most requested instruments on the Mayo stand at all times – a very effective strategy and a difficult target to beat. Thus, we consider the 52% reduction very significant. As another measure of effectiveness, consider that our rules are designed to move category 1 and 2 instruments forward, closer to the surgeon. 88% of the time this included the instrument the surgeon was about to request. Moreover, our algorithm's performance shows clear improvement as the amount of training data increases. This indicates that the algorithm is able to learn from experience, making better predictions as time goes by.

To test our algorithm on a surgical procedure, we ran through the case trying to predict each instrument request. We measured the accuracy rate of these predictions and the instrument delivery time for the request. This delivery time depends on which cache the instrument was in at the time of the request. From our engineering analysis we have determined that deliveries from the Mayo stand will take 1 second, those from the staging zone will take 3 seconds, and deliveries from the back tray will take 4 seconds. The latter two take more time because the robotic arm must swing around to reach these areas. Instruments on the transfer zone take no time to "deliver", as the surgeon can simply pick them up at will. We computed the baseline instrument delivery time in the same manner.

Test Regimes

We used two different test regimes to evaluate our prediction algorithms: *full round-robin tournaments* and *chronological learning*. For round-robin tournaments we took each test case from the data set in succession. We trained the algorithm on all the surgical procedures *except* that case and then performed the test. These tournaments exercised the algorithm on the widest variety and largest volume of training data. For chronological learning tests we sorted all the surgical procedures from the data survey by date. We then ran through the procedures, training the algorithm cumulatively on each one after it was tested. This simulates the experience the robot would have had if it were fielded in the test facility. It would begin with no training at all and build it's occurrence count database as it is used. This test regime shows how performance changes as a function of the amount of training data.

Parameter Tuning

There are several parameters we can adjust to control the behavior of our algorithm. The most important parameter is the value of *N*, the number of past instrument requests to consider when making predictions. This is highly domain dependent as it reflects the size of the largest discernable repeating pattern. In full round-robin tournament testing we found that small values for *N* worked best. Performance peaked at *N* equals 3, resulting in an average delivery time of 0.69 seconds. This confirmed our intuition that most common surgical instrument request patterns are 2 or 3 requests in length.

The category 1, 2, and 3 prediction score cutoffs are another important set of adjustable parameters. These numbers work in conjunction with the inter-cache move rules to define the mapping between prediction scores and the robot's behavior. If an instrument is in category 1 the rules treat it as highly likely to be needed again soon, while category 2 instruments are only somewhat likely. Category 3 means it's highly *unlikely* the instrument will be needed again. The cutoffs determine how many instruments will be in each of these categories, greatly affecting the flow of instruments back and forth.

An interesting possibility for a future enhancement is to adjust these category cutoffs dynamically, either after each surgical procedure or even during the procedure itself. When our voice recognition software detects the actual next instrument request we can compare it to our predictions, thus providing a basis for a feedback loop. If the performance of the prediction algorithm is poor, we could adjust the category cutoffs to put fewer instruments in category 1 and more in category 2. This would effectively lower our confidence level in the prediction scores temporarily, allowing us to run more conservatively when we encounter a truly novel instrument request sequence.

Performance And Learning

Table 4 shows the accuracy rates our algorithm achieved in full round-robin tournament testing. Prediction scores are computed for every instrument type giving us a ranking for each instrument. We define accuracy rates for various ranges within this list, indicating the percentage of the time the actual next instrument was ranked within that range. So 28.1% of the time we correctly predicted the exact next instrument. 42.2% of the time the next instrument was ranked either number 1 or 2. Our category 1 prediction score cutoff is 8, therefore 82.2% of the time we tried to move the correct next instrument onto the transfer zone The cutoff for category 2 is 11, so 88.1% of the time we tried to move the correct next instrument forward, either to the transfer zone or Mayo stand.

Range	Rate	Range	Rate	Range	Rate
1	28.1%	1-5	66.6%	1-9	84.9%
1-2	42.2%	1-6	72.3%	1-10	86.6%
1-3	51.4%	1-7	78.2%	1-11	88.1%
1-4	59.9%	1-8	82.2%	1-12	89.5%

Table 4. Overall accuracy rates for the top ranking prediction scores. Prediction scores are computed for every instrument type and ranked from highest to lowest. The rates indicate the percentage of the time the actual next instrument was ranked within the corresponding range.

Figure 6 shows the results of our chronological learning tests. Each of the 51 surgical procedures in the data survey, from first to last, is shown along the horizontal axis. For the first procedure the occurance count database is empty. Then, as each procudeure is completed, the data from that case is added to the database. For each case two instrument delivery times are shown, one using the prediction algorithm and one using the baseline strategy. The baseline strategy (keeping the top 12 instruments on the Mayo stand) is static and thus doesn't change over time. The prediction algorithm, however, learns from each case. For the first few procedures it actually performed worse than the baseline. It improved quickly though and from the fourth case on our algorithm outperformed the baseline. Note that this learning curve continues to diverge from the baseline even beyond the 4 months worth of data in our survey. This means that if the robot had been fielded in the test facility, it would have performed better and better over this period.

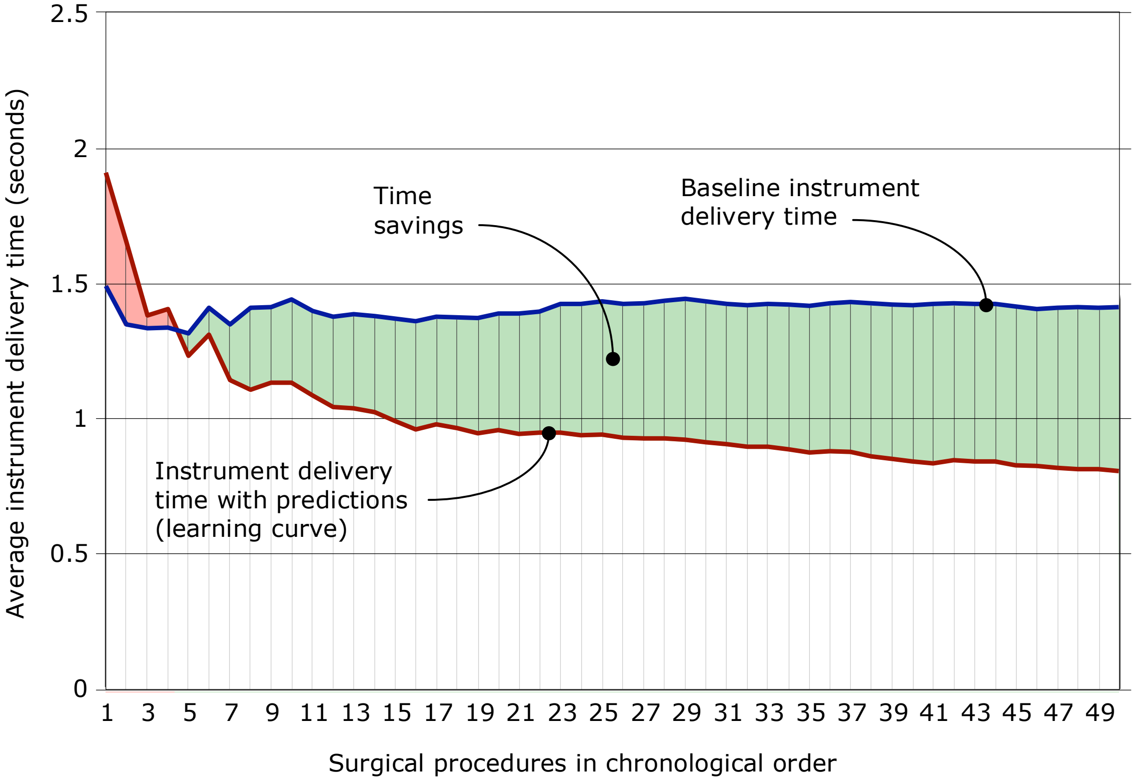


Figure 6. The chronological learning curve showing the decrease over time in instrument delivery time as the prediction algorithm is trained on more and more data. The baseline instrument delivery time, which is not a function of the amount of training data, is also shown. Note that the delivery time curve with predictions is continuing to diverge from the baseline. Also note that all times are cumulative average delivery times at the given point in time.

Future Work

We are very encouraged by the results of this research and feel it will form a strong foundation for future work. As we've shown, this prediction algorithm provides a good framework for making decisions about how to organize the surgical instruments. This is a good start, though only one of the many decisions the robot will need to make in the dynamic OR environment. Human scrub technicians have the advantage of a *situational awareness* of the surgical procedure as it progresses. They are able to sense what is happening and adapt their behavior accordingly. While a model of likely future instrument requests is an important aspect of that situational awareness, it is just the beginning.

To perform effectively and safely we feel the robot will need a deeper awareness of many facets of its envirorment. It should be capable of monitoring the phase of the surgical procedure, from initial incision to final closing. It should be able to gauge the current level of emergency or lack thereof. It should be capable of understanding and adapting to the surgeon's preferences. It must also have the ability to recognize its own errors and rectify them in the future. These are all good examples of the advanced situational awareness we intend to incorporate in our robot.

Our cognitive architecture is the means to this end. It will build on the simple, yet effective reasoning capabilities we've explored in this research effort. It will add support for a more general knowledge representation language and mechanisms for reasoning on a wider variety of topics. It will allow the robot to generate and test hypotheses about the current state of the surgical procedure. It will also utilize more of the robot's senses as input to these reasoning processes. The instrument prediction capabiltiy described here relies on the robot's voice recognition system for feedback. The full cognitive architecture will support input from other senses as well, including the machine vision system and the force feedback system.

We feel that this research has been invaluable in advancing our goal of a clinically and commercially viable robotic scrub technician. It has, for the first time, shown how a medical device in the OR can form an *opinion* about the probable near-term future of the surgical procedure and act autonomously on that opinion. The addition of a full cognitive architecture to broaden the scope and deepen the insight of these opinions, will complete the cognitive framework we'll need to enable our robot to do its job safely and effectively. We then intend to deliver the first robot that can be accepted as a capable, reliable – albeit nonhuman – OR team member.