Real-Time Identification of Operating Room State from Video

Beenish Bhatia and **Tim Oates**
Department of CSEE
University of Maryland Baltimore County
1000 Hilltop Circle
Baltimore, MD 21250
{beenish1,oates}@cs.umbc.edu
Voice: 1-410-455-3082
Fax: 1-410-455-3969

Yan Xiao and **Peter Hu**
Department of Anesthesiology
School of Medicine
University of Maryland
685 West Baltimore Street
Baltimore, Maryland 21201
{yxiao,phu}@umaryland.edu
Voice: 1-410-706-1859
Fax: 1-410-706-2550

Abstract

Managers of operating rooms (ORs) and of units upstream (e.g., ambulatory surgery) and downstream (e.g., intensive care and post-anesthesia care) of the OR require real-time information about OR occupancy. Which ORs are in use, and when will each ongoing operation end? This information is used to make decisions about how to assign staff, when to prepare patients for the OR, when to schedule add-on cases, when to move cases, and how to prioritize room cleanups (Dexter *et al.* 2004). It is typically gathered by OR managers manually, by walking to each OR and estimating the time to case completion. This paper presents a system for determining the state of an ongoing operation automatically from video. Support vector machines are trained to identify relevant image features, and hidden Markov models are trained to use these features to compute a sequence of OR states from the video. The system was tested on video captured over a 24 hour period in one of the 19 operating rooms in Baltimore's R. Adams Crowley Shock Trauma Center. It was found to be more accurate and have less delay while providing more fine-grained state information than the current state-of-the-art system based on patient vital signs used by the Shock Trauma Center.

Introduction

Scheduling operating rooms (ORs) is an important problem. Optimizing the use of the vast array of physical and human resources required in an OR suite can both improve patient outcomes and reduce the hospital's costs. Although scheduling is a hard problem in general, scheduling ORs is particularly difficult. Each day begins with a schedule of planned cases, but this schedule invariably falls apart due to unplanned cases (e.g., trauma cases), doctors being unavailable for surgery at the appointed time because of other responsibilities, doctors adding or removing procedures dynamically during surgery, procedures taking more or less time than expected, patients whose condition either improves or degrades, thus making the planned surgery unnecessary or ill advised, and so on. This can result in doctors, nurses, and other hospital staff waiting for resources to be free, and in prolonged waiting times for patients, who are typically asked to arrive at the hospital very early and are required to fast until after surgery.

All of these sources of variance make obtaining accurate, real-time information about OR occupancy vitally important. Which operating rooms are in use, and when will the ongoing procedures end? This information is used to make a variety of decisions, such as how to assign staff, when to prepare patients for the OR, when to schedule add-on cases, when to move cases, and how to prioritize room cleanups (Dexter *et al.* 2004). It is also used by units upstream (e.g., ambulatory surgery) and downstream (e.g., intensive care and post-anesthesia care) of the OR to make similar decisions about resource allocation.

Despite the fact that the medical technology available in the OR is stunning in its power and sophistication, the technology used to establish and maintain schedules is decades old. White boards and sticky notes are the primary tools of the trade. Nursing and anesthesia staff record the times at which patients enter and leave the OR manually based on a variety of time sources (e.g., the OR clock, a wrist watch, or a clock on a monitor in the OR), and OR managers physically walk about the suite, peering in windows to gather information on the state of procedures in an effort to estimate when they will end.

Recently, (Xiao *et al.* 2005) significantly advanced the state of the art in OR management by implementing a system that remotely monitors patient vital signs (VS) to answer three questions about each OR: (1) Is the OR occupied by a patient? (2) When did the current patient enter the OR? (3) When did the last patient leave the OR? This system is currently in production use in a suite of 19 operating rooms in R. Adam's Crowley Shock Trauma Center, which is the nation's only trauma hospital.

The work reported in this paper applies machine learning techniques to video to estimate OR state. Support vector machines are trained to identify relevant image features, and hidden Markov models are trained to use these features to compute a sequence of OR states from the video. Our system represents a significant advance over the VS system in several ways. First, our system is more accurate than the VS system in determining OR occupancy. Second, due to noise in the VS data, the VS system's output lags real time by 3-5 minutes, whereas our system produces real-time output for each frame of video as it is processed. Third, because our system is based on video, it provides more fine grained information about the state of an ongoing procedure than is

Figure 1: The operating room and operating table are ready for a new patient. We call this the *single bed* or SB state.

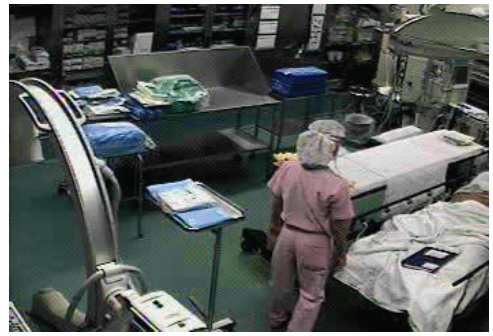

Figure 2: A patient is either entering or leaving the operating room. We call this the *double bed* or DB state.

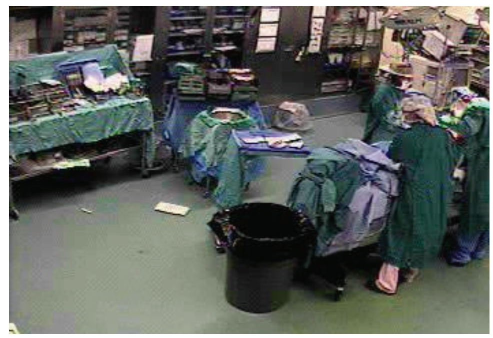

Figure 3: The patient is covered with a drape and the operation is underway. We call this the *drape on* or DON state.

possible using just vital signs.

The remainder of this paper is organized as follows. The next section provides additional background information on the problem (determining OR state), the existing solution based on vital signs, and the specific machine learning techniques used (support vector machines and hidden Markov models) by our new system. The two sections after that give details on our video-based system and the results of experiments on 24 hours of video from one of the ORs at the Shock Trauma Center. The last section concludes and points to future work.

Background

Though no two procedures performed in an OR are ever precisely the same, at some level of abstraction (indeed, at a level that is very close to the one at which OR managers seek information) all procedures go through the same sequence of states. As shown in Figure 1, each OR contains a variety of equipment that can vary from room to room and procedure to procedure, but one constant is the operating table, which is on the right-hand side in the middle of the figure. The white sheet on the table indicates that it is clean and ready for a patient. In contrast, when an operation ends and the room is cleaned, the sheet is removed and the black table beneath is visible.

When a patient is ready for an operation, they are wheeled in on a second bed as shown in Figure 2, transferred to the operating table, and the second bed is wheeled back out of the OR. The patient is then prepared for surgery, which includes covering the patient with a (typically) blue drape as shown in Figure 3. When the drape is placed on the patient, the procedure is about to begin. When the procedure is done, the drape is removed (see Figure 4), the second bed is wheeled back into the OR, the patient is transferred to the second bed and wheeled out, the bed and room are cleaned, and the OR returns to the state shown in Figure 1. We refer to the OR states shown in Figures 1 - 4 as *single bed* (SB), *double bed* (DB), *drape on* (DON), and *drape off* (DOFF), respectively. This high-level sequence of states is repeated for each operation, though the details of each procedure while the patient is covered by the drape, including the use of auxiliary equipment, can vary dramatically.

The system for determining OR state to support OR scheduling currently in use at the Shock Trauma Center does not use video, but uses patient vital signs.[1] Patients are typically connected to a variety of VS monitors (e.g., non-invasive blood pressure, arterial blood pressure, temperature, electrocardiogram, and pulse oximetry) soon after entering the OR, and are disconnected shortly before leaving. The presence of clinically valid readings from these sensors indicates that a patient is in the OR, and the absence of such readings contra-indicates the presence of a patient. One problem with VS sensors is noise. False negatives can occur, for example, when the sensors are removed to reposition the patient or when electrocautery interferes with the sensors. False positives can occur when the sensors are left on and are handled by the staff when not connected to a patient.

The VS system performs three stages of processing (Xiao *et al.* 2005). First, the VS data are validated to ensure that they are clinically valid. For example, a heart rate of 10 -

[1]The developers of the VS system had considered simply adding a switch to each OR that a nurse flips one direction when the patient enters the room and flips the other direction when the patient leaves the room. This was rejected because it adds another responsibility to already overburdened nursing staff, could easily be forgotten by the staff, and does not allow for easy extension to more fine grained information about the state of a procedure (as compared to, for example, video).

Figure 4: The operation is over and the drape has been removed. We call this the *drape off* or DOF state.

200 beats per minute is treated as if it came from a patient, and heart rates outside that range are treated as anomalous signals (e.g., from a disconnected sensor). Second, the data are windowed to filter out short transient changes (false positives and false negatives) to establish the OR state (occupied or unoccupied) with greater certainty. Third, the state of the OR is presented to the OR manager showing for each OR when the current case started (and how long it has been ongoing) and when the last case finished (and how long the OR has been unoccupied). It is the second step of processing that introduces significant lags in the output of the system.

Note that the ability to classify images as either SB or not SB (i.e., DB, DON, or DOFF) is sufficient to replicate the functionality of the VS system. Because image data are richer than VS data, we can do more than simply say whether an OR is occupied or not. For example, images such as those shown in Figure 3 indicate not only that the OR is occupied, but also that the operation is underway. Also, for the OR shown in Figure 2 the VS system would indicate that the patient is not connected, but the image data tells us in addition that the patient is still in the room.

However, the image in Figure 2 (an instance of DB) is ambiguous. It could be the case that the operation has not begun and the patient is being wheeled in, or that the operation is complete and the patient is being wheeled out. The image in Figure 4 (an instance of DOFF) is ambiguous as well. It could be the case that the operation is about to begin and the drape has not yet been placed on the patient, or that the operation is complete and the drape has been removed. Resolving this ambiguity requires information about the images that preceded the one in question. For example, if a DB image occurs after an SB image, the patient is being wheeled into the room for an operation. If a DB image occurs after a DOFF image, the patient is being wheeled out of the room after the operation. Likewise, ambiguity of DOFF images can be resolved by noting whether they were preceded by DON images (indicating the operation is ending) or DB images (indicating the operation is beginning).

Therefore, our system is based on two machine learning technologies. First, we use support vector machines (SVMs) to classify images as being instances of SB, DB, DON, or DOFF. Second, we use hidden Markov models (HMMs) to to remove ambiguity in sequences of these labels as produced by the SVMs and generate a sequence of operating room states, such as *patient entering*, *patient exiting*, *operation beginning*, and *operation ending*. HMMs have been shown to be effective in similar applications in the past (Xie & Chang 2002). We now briefly review SVMs and HMMs.

Support vector machines (Scholkopf & Smola 2002) are used to solve supervised classification problems. Given training instances of the form (x, y) where $x \in \Re^n$ and $y \in \{-1, 1\}$, the goal is to find a weight vector $w \in \Re^n$ and bias $b \in \Re$ such that $x \cdot w + b > 0$ when $y = 1$ and $x \cdot w + b < 0$ when $y = -1$. SVMs can be used to find linear decision surfaces or, using a method called the "kernel trick", non-linear decision surfaces. Different kernels (which are essentially measures of similarity between vectors) make different kinds of non-linearities available to the learning algorithm. In our system, the x values are derived from images (in a way that will be described in the next section), and the y values are, for example, 1 if the image is an instance of SB and -1 otherwise. The labels (i.e., whether an image is an instance of SB, DB, DON, or DOFF) were obtained manually by marking regions of the video with the correct label. We chose to use SVMs because they are efficient, provide strong theoretical guarantees on generalization performance, and have been shown to perform well in practice in a variety of domains, including image classification.

A (finite) hidden Markov model (Rabiner 1989) is a 5-tuple (S, O, A, B, π) where S is a finite set of states, O is a finite set of observations, A is a set of transition probabilities that maps from $S \times S$ to $\Re$, B is a set of observation probabilities that maps from $S \times O$ to $\Re$, and π is a probability distribution over initial states. An HMM generates observations as follows. An initial state s is selected probabilistically according to π. An observation (an element of O) is then produced according to the distribution over observations defined by $B(s)$. Finally, a next state (an element of S) is chosen according to the distribution over states defined by $A(s)$, a new observation is produced for this new state and the process repeats.

In our system, the observations correspond to the output of the SVMs, i.e., whether an image contains SB, DB, DON, or DOFF. The HMM states correspond to states of the operation, such as *patient entering* and *operation ending*. Given an HMM and an observation sequence, there exists an algorithm, the Viterbi algorithm (Viterbi 1967), for finding the most likely state sequence that produced the observation sequence. We use this algorithm to extract sequences of operating room states from sequences of image features identified by the SVMs in the video.

Methods

This section describes in detail how we use a combination of image processing, SVMs, and HMMs to determine OR state. Several of the 19 operating rooms in the trauma bay at the Shock Trauma Center are equipped with two video cameras (Hu *et al.* 2006). The video used in the experiments described in the next section were obtained from one of the cameras in one of the ORs over a 24 hour period. The video resolution was initially 352×240 pixels per image recorded at 30 frames per second. We downsampled to 3 frames per second to reduce the amount of computation required by image processing.

All of the images were initially in RGB format, which we converted to HSV (hue, saturation, and value). Each image was then converted into a hue histogram with 256 bins. Hue values are integers in the range 0 to 255, so histogram bin i for an image contains the number of pixels in the image with hue i. Histogram counts were then normalized to be in the range 0 - 1 by dividing all bins by the count in the largest bin. These normalized histograms are used to represent the contents of the images and serve as the x value (see the previous section's discussion of SVMs) in the training instances supplied to the SVMs. Note that this representation captures only the distribution of colors in the images, regardless of how they are arranged spatially. We initially considered using more complex image features, such as edges and textures, but hue histograms are much easier to compute and, as will become clear shortly, yield good results.

Next, we split the 24 hours of video into 24 one hour videos, and manually labeled regions of the video according to whether they contained instances of SB, DB, DON, or DOFF. This labeling process was not tedious or time consuming because each of these states persists for minutes (e.g., DB when a patient is entering or leaving the OR) or hours (e.g., DON when a procedure is in progress) so the number of transitions between states is small. Of the original 24 hours, there were 3 hours of SB or DB, and 4 hours of DON or DOFF. The freely available SVM-Light implementation of support vector machines (Joachims 1998) was used to learn two different classifiers, one to discriminate SB from DB and one to discriminate DON from DOFF. In the SB/DB case, the training instances provided to the SVM algorithm were the hue histograms paired with class label $y = 1$ if the image contained SB or $y = -1$ if the image contained DB. In the DON/DOFF case, the class label was $y = 1$ if the image contained DON or $y = -1$ if the image contained DOFF. We tried a variety of kernels, including linear, polynomial, hyperbolic tangent, and Gaussian, each with a wide variety of settings of the relevant parameters. The linear kernel performed best, so that is what we used in the experiments.

Recall from the previous section that an HMM is a 5-tuple (S, O, A, B, π). The outputs of the two SVMs are treated as the observations, O, for the HMM. Both classifiers are applied to each image, resulting in one of the following four observations: SB and DON; SB and DOFF; DB and DON; DB and DOFF. That is, the first classifier will assert that the image contains either SB or DB, and the second classifier will assert that the image contains either DON or DOFF, resulting in the four possible combinations above. Each of these four combinations is treated as an observation that can be generated from an HMM state.

Figure 5 shows the states, S, and transitions in the HMM. Because we don't know the state of the OR when the camera is turned on, the distribution over initial states, π, is uniform. The leftmost state (labeled SB) corresponds to the OR being ready for a patient. A transition to the second state (labeled DB) occurs when a patient enters the room. A transition

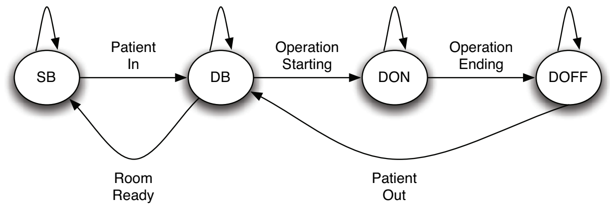

Figure 5: The states and transitions of the HMM used to capture the sequential nature of procedures in an operating room. In a typical operation, the patient enters the room (SB to DB), the operation begins (DB to DON), the operation ends (DON to DOFF), the patient leaves the room (DOFF to DB), and the room is ready for a new patient (DB to SB).

Observation	SB	DB	DON	DOFF
SB, DON	0.0118	0.0037	0.9004	0.4190
SB, DOFF	0.9840	0.1757	0.0128	0.4604
DB, DON	0.0023	0.0315	0.0849	0.0771
DB, DOFF	0.0019	0.7891	0.0019	0.0435

Table 1: Observation probabilities for each state in the HMM in Figure 5.

to the third state (labeled DON) occurs when the drape is placed on the patient and the operation begins. A transition to the fourth state (labeled DOFF) occurs when the operation ends and the drape is removed. A transition from the fourth state to the second state (DOFF to DB) occurs when the patient is removed from the OR, and a transition from the second state to the first state (DB to SB) occurs when the OR is once again ready for a new patient.

To fully specify the HMM it remains to determine the transition probabilities, A, and the observation probabilities, B. Because each state, once entered, persists for quite a long time relative to the sampling rate of the images (which is 3 per second), we set the probability of transitioning to a new state to be small ($\epsilon \approx 0$) and the probability of remaining in the same state via a self transition to be high ($1 - \epsilon$). In the experiments reported in the next section $\epsilon = 10^{-40}$.

The observation probabilities, A, are shown in Table 1. Each row corresponds to one of the four observations, and each column corresponds to one of the states in the HMM. Each cell corresponds to the probability of the given observation occurring in the given state. Note that the columns sum to 1 because the observations in the table are mutually exclusive and exhaustive. These probabilities were obtained by applying the SVMs to the video to obtain the observations, and then manually assigning each video frame to a state in the HMM. It was then a simple matter of counting, for each state, the number of occurrences of each observation and dividing by the total number of observations (visits to that state) to obtain the probabilities in Table 1.

Note that in some cases the SVM outputs alone are sufficient to accurately identify the state of the OR. For example, when the observation is SB/DOFF, the probability of being in the SB state is 0.9840. In contrast, when the observation

Accuracy on SB/DB (%)
96.54
70.93
57.33

Table 2: Accuracy of the SB/DB SVM when trained on 2 hours of video and tested on the third hour.

is SB/DON the probability of being in the state DOFF is 0.4190. That is, the output of the DON/DOFF SVM is misleading in this case. As we will see if the next section, the HMM is remarkably effective at using the sequential nature of the observations to correct for errors made by the SVMs. In addition, the HMM allows us to determine OR states such as *patient in* and *patient out* by noticing the path taken to the DB state which, by itself, merely indicates patient movement.

The process of converting images to hue histograms, training SVMs, and estimating the parameters of the HMM is interactive but highly automated. Therefore, retraining the entire system for new operating conditions, such as a different hospital, requires some expertise but little effort.

Results

First we explore the accuracy of the SVMs with respect to the task of classifying images as SB/DB and DON/DOFF. Then we turn to the accuracy of the HMM when using the SVM outputs as observations and the HMM states as proxies for OR states.

As the last section mentioned, 3 of the 24 one hour videos contained exclusively SB or DB. The total number of images in these three hours was 32,377, which, for insignificant reasons, is slightly less than the $60 * 60 * 3 * 3 = 32,400$ images that results from sampling 3 times per second for 3 hours. The number of SB images was 24,534 and the number of DOFF images was 7,843. To determine the accuracy of the SVM on this task we initially performed leave-one-out cross-validation in which the SVM is trained on all of the images but one, with each of the 32,377 having a turn being the held out image, and the using the SVM to classify the held out image. However, the results were skewed (toward higher accuracy) because the images are not independent of one another. That is, if we hold out the image taken at time t, it will be extremely similar to the images taken at times $t + 1$ and $t - 1$. In this case, a simple nearest neighbor classifier would be expected to get close to 100% accuracy.

To meliorate this problem, we iteratively trained on two of the hours and tested on the third, resulting in three accuracy scores. The results are shown in Table 2. Note that the performance of the SVM varies quite a bit depending on the contents of the training and testing sets, with a maximum accuracy of 96.54%, a minimum accuracy of 57.33%, and a mean accuracy of 74.93%.

A similar method was used to test the accuracy of the DON/DOFF SVM, but because there were 4 hours of data containing exclusively those two states the training data con-

Accuracy on DON/DOFF (%)
93.43
88.03
65.52
84.18

Table 3: Accuracy of the DON/DOFF SVM when trained on 3 hours of video and tested on the fourth hour.

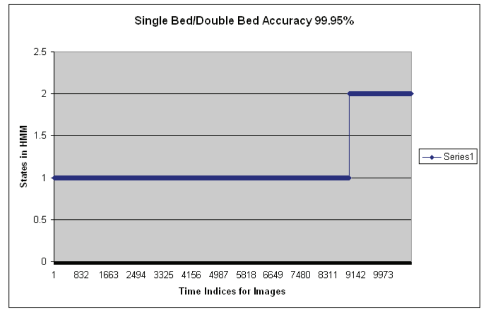

Figure 6: State assignments made by the HMM to one hour of SB/DB data.

sisted of 3 hours with the remaining hour used for testing. The resulting accuracies are shown in Table 3. Again, the accuracies vary widely, with a maximum of 93.43%, a minimum of 65.52%, and a mean of 82.79%.

Despite our initial hopes that the SVMs alone would be sufficient, the doctors working with us were clear that the accuracies in Tables 2 and 3 were insufficient for a production system. Therefore, we now turn our attention to the results obtained by coupling the SVMs and the HMM.

The HMM shown in Figure 5 as described in the previous section was used to find the most likely parse (the Viterbi parse) of a sequence of observations produced by one hour of SB/DB data, using the other two hours for training the SVMs. The states assigned by the HMM are shown graphically in Figure 6. The horizontal axis is the index of the images parsed by the HMM. The vertical axis corresponds to the states of the HMM, with SB being state 1 and DB being state 2. The HMM assigned all images until about image number 9000 to the SB state, and all subsequent images to the DB state. The accuracy was 99.5%, with the errors occurring at the state transition, either transitioning from SB to DB too early or too late.

Figure 7 shows similar results for an hour of the DON/DOFF data, again using the other three hours for training the SVM. In this case, state 3 is DON and state 4 is DOFF. Note that the HMM begins in DON (state 3) and quickly transitions from DON to DOFF (state 4), which remains constant until about image number 3000. Then there is an erroneous momentary transition to DB (state 2), and a quick change back to DON. The accuracy on this dataset is 99.3%.

Finally, Figure 8 shows the results of the HMM on two hours of data consisting of the DON/DOFF merged with the SB/DB data. Again, there are a few short obvious errors, but

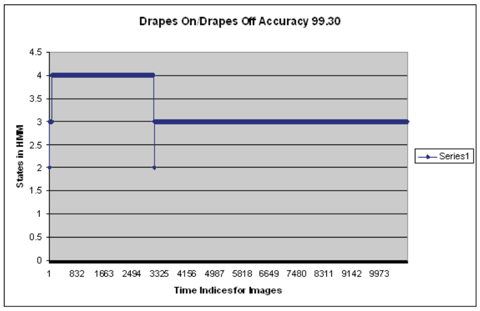

Figure 7: State assignments made by the HMM to one hour of DON/DOFF data.

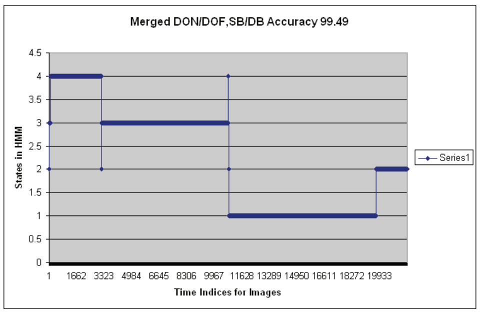

Figure 8: State assignments made by the HMM to the merged SB/DB and DON/DOFF data.

the HMM quickly corrects itself and obtains an accuracy on the full two hours of data of 99.49%.

Because the errors in the figures above tend to occur at the transitions between states, we experimented with windowing the data. That is, rather than output the state of the HMM as determined by the Viterbi algorithm, we collect n consecutive states and output the majority vote within that window. The results for different window sizes are shown in Table 4. No window size helped significantly, and larger window sizes hurt performance because they made the predictions lag behind reality.

Window Size	SB/DB	DON/DOFF	Merged
13	100.00	99.30	99.49
23	99.95	99.29	99.45
53	99.81	99.29	99.31
93	99.63	99.29	99.19
193	99.15	99.01	98.58
293	98.67	98.52	97.87
393	98.17	98.03	97.15
493	97.67	97.52	96.43

Table 4: The effects of windowing on accuracy on the SB/DB, DON/DOFF, and merged datasets.

Conclusion

This paper presented a system for determining the state of an ongoing operation automatically from video. Support vector machines are trained to identify relevant image features, and hidden Markov models are trained to use these features to compute a sequence of OR states from the video. The system was tested on video captured over a 24 hour period in one of the 19 operating rooms in Baltimore's R. Adams Crowley Shock Trauma Center. It was found to be more accurate (roughly 99% vs. roughly 95% for similar circumstances (Xiao *et al.* 2005)) and have less delay (output 3 times a second in real time vs. a 3 - 5 minute delay (Xiao *et al.* 2005)) while providing more fine-grained state information than the current state-of-the-art system based on patient vital signs used by the Shock Trauma Center.

Future work will involve both training and testing on more data from different ORs at the Shock Trauma Center, adding new states to the HMM such as "lights out in the OR" and "operating table stripped" and "room being cleaned", and ultimately integrating into the environment at the Shock Trauma Center to augment the current vital signs system.

References

Dexter, F.; Epstein, R. H.; Traub, R. D.; and Xiao, Y. 2004. Making management decisions on the day of surgery based on operating room efficiency and patient waiting times. *Anesthesiology* 101:1444 – 1453.

Hu, P.; Ho, D.; Mackenzie, C. F.; Hu, H.; Martz, D.; Jacobs, J.; Voit, R.; and Xiao, Y. 2006. Advanced telecommunication platform for operating room coordination: Distributed video board system. *Surgical Innovation* 13(2):129–135.

Joachims, T. 1998. Making large-scale support vector machine learning practical. In Scholkopf, B.; Burges, C.; and Smola, A., eds., *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

Rabiner, L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257–285.

Scholkopf, B., and Smola, A. J. 2002. *Learning with Kernels*. MIT Press.

Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* 13(2):260–269.

Xiao, Y.; Hu, P.; Hu, H.; Ho, D.; Dexter, F.; Mackenzie, C. F.; Seagull, F. J.; and Dutton, R. P. 2005. An algorithm for processing vital sign monitoring data to remotely identify operating room occupancy in real-time. *Anesthesia and Analgesia* 101(3):823 – 829.

Xie, L., and Chang, S.-F. 2002. Structure analysis of soccer video with hidden markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*.