MODELING HUMAN MOTION FOR PREDICTING USAGE OF HOSPITAL OPERATING ROOM

A Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Ilyes Sghir

August 2014

MODELING HUMAN MOTION FOR PREDICTING USAGE OF HOSPITAL OPERATING ROOM

Ilyes Sghir

APPROVED:

Dr. Shishir Shah, Chairman
Dept. of Computer Science
University of Houston

Dr. Edgar Gabriel
Dept. of Computer Science
University of Houston

Dr. Saurabh Prasad
Dept. of Electrical and Computer Engineering
University of Houston

Dean, College of Natural Sciences and Mathematics

"Great and sacred are the thoughtful deliberations required to preserve the lives and health of Thy creatures" - ***Moses Maimonides***.

Acknowledgements

My special thanks go out to my advisor Prof. Shishir Shah from the University of Houston for his patience, his support and his precious advices.

I would like to thank the European Union and the United States, the Atlantis CRISP Dual degree program, and the Alsace Region who provided me financial support. My appreciation goes out to Prof. Christophe Collet from the University of Strasbourg who made this exchange possible.

I also would like to thank all the members of the Quantitative Imaging Laboratory for answering my questions, and helping me.

Finally, and most importantly, a huge thank you to my parents, brother and sisters who supported me during my studies and encouraged me to reach my dreams.

MODELING HUMAN MOTION FOR PREDICTING USAGE OF HOSPITAL OPERATING ROOM

An Abstract of a Thesis

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Ilyes Sghir

August 2014

Abstract

We present a system that exploits existing video streams from an hospital operating room (OR) to infer the OR usage state through Bayesian modeling. We define OR states based on common surgical processes that are relevant for assessing OR efficiency. The human motion pattern within the OR is analyzed to ascertain usage states. The system proposed takes advantage of a discriminatively trained part-based human detector as well as a data association algorithm to reconstruct motion trajectories. Human motion patterns are then extracted using kernel density estimation and a Bayesian classifier is used to assess OR usage states during testing. Our model is tested on a large collection of videos and the results show that human motion patterns provide significant discriminative power in understanding usage of an OR.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Operating Room (OR) is by far the most complex and expensive environment within any hospital. With the advent of technology and the increase in the number of minimally invasive surgeries, ORs have become high costs / high revenues assets. Nonetheless, their effective utilization hasn't been fully realized. Although no published formal data assessing their performance can be found, it was estimated in 2003 that ORs generated almost half of a hospital's revenues while running at only 68% of their capacity [1]. Assessing workflow performance would significantly improve quality of healthcare delivery and increase financial outcomes for a hospital.

Unplanned events, inefficient supply chain management, but most importantly lack of operational discipline highly affect OR performance. In fact, start-time delays [6, 8, 22], as well as, unregulated turnover time [14, 1] have been identified as major causes of OR inefficiency. Studies focusing on start-time delays have been performed in numerous hospitals in Europe and the United States [8]. Does *et al.* [8]

focused on the start-time delay of the first operation of the day and harvested 4-weeks data from 13 hospitals in Belgium and the Netherlands. By defining the start-time as the time of the first incision they concluded that delays range from 25 min to 103 min [8]. Turnover time or the time-lapse between 2 different surgeries lasts 30 min on average while in best practice it should last only 15 min [1]. Macario estimated in 2010 that in US hospitals a running OR costs about $20/min in material supplies while generating on average $60/min in revenue [17]. If we approximate start-time morning delays to be 60 min and the cost of an OR to be $2000/hour, then a hospital with 10 ORs running 250 days a year, can potentially save 5 million dollars each year. Optimizing clinical processes within the OR also affects quality of healthcare delivery. In fact, reducing time loss would allow scheduling of more surgical procedures and reduce the average waiting time for patient treatment.

According to Ciechanowicz and Wilson [6], regular local audit of OR usage is important to optimize the clinical processes within the OR and the perioperative environment. One of the most recent benchmarks involved 22 German hospitals and more than 20,000 case analysis over a 9-months period [22]. Nonetheless, studies performed until now have been primarily based on manual data acquisition by nurses. Daily and automated information about OR efficiency would be of high value at the administrative level for continuous quality improvement. Furthermore, large scale robust comparative studies are needed that could be much easily conducted at any time and for different periods of time.

The more general problem of workflow monitoring is already being addressed in more constraint industrial environments such as car manufacturing [23]. Various

solutions have been proposed in the literature for enhancing OR throughput by facilitating its management. In 2007, one of the systems used at the MIT General hospital was the OR-Dashboard, which is a solution offered by a company called LiveData [16]. OR-Dashboard displays information about the patient and the surgical procedure. Other commercial solutions can be found such as ORBIT [15] or AwareMedia [2]. More recently, in 2011, Niu *et al.* proposed a simulation model for performance analysis of the OR [19]. Unfortunately, all these solutions rely on human intervention and manual data entry. To address this inconvenience, alternative approaches consist of leveraging electronic signals present in the OR in order to identify automatically its usage state without human intervention. In 2005, Xiao *et al.* [24] proposed to use patient's vital signs in order to monitor when the subject is in the OR or not. Later on, in 2007, Bhatia *et al.* [3] designed a system analyzing video streams and automatically recognizing the OR state using Machine learning algorithms (SVM and HMM). In 2009, Padoy *et al.* [20] exploited a multiple-camera system for extracting low level 3D motion features that are ultimately fed into a workflow-HMM. In 2010, Lange *et al.* [15] proposed a phase recognition system using sensor technology. Yet, designing a system that doesn't involve embedded body-worn sensors is more convenient. In 2011, Nara *et al.* [18] introduced an ultrasonic location aware system that tracks continuously the 3D position of the surgical staff in order to recognize discriminant human motion patterns.

The effective utilization of video streams within the OR hasn't been fully realized. A single video can provide cues that can be used to ascertain the usage state. In this paper, we present a system that exploits existing video streams from an OR

in order to extract human motion trajectory data and that infers the OR usage state through Bayesian modeling. Unlike Bhatia *et al.*, we exploit a single feature that has a physical meaning, the human motion pattern. Further, we do not define our OR states based on the presence of objects in the scene (second bed, drape on and off). Phase recognition isn't based on velocity values as proposed by Padoy *et al.* [20], but instead on spatial distribution of the positions occupied by different staff members. Instead of using a large ultrasonic location-aware system like Nara *et al.* [18], we take advantage of a detection algorithm based on a discriminatively trained part-based upper-body model developed using Felzenswalb *et al.*'s object detection framework [11, 10]. We use a data association algorithm based solely on a 2D geometrical feature. Spatial occupancy of the OR by staff members is then evaluated using a kernel-based method. Finally, OR usage state is derived from the density of the detected features, instead of predetermined frame rates.

Chapter 2

OR Usage-State Model

2.1 Usage-state transition model

Typically, when a patient is admitted in an OR, an anesthesiologist starts administrating anesthesia. Once the patient is ready, surgeons proceed to the first incision [22]. At the end of the surgical procedure, all the instruments are wrapped up, the surgical staff proceeds to clean up and the patient is transfered to the recovery room. In this paper, we propose a three-stage usage-state transition model. Human motion patterns vary in the presence or absence of a patient within the OR. This simple observation is the motivation for the states in our model as shown in Figure 2.1.

"Setting Up" is the usage state in which the surgical staff is either cleaning-up or getting the OR ready for surgery while no patient is within the OR. Once

the patient is introduced in the room and if no surgeon is performing surgery, the usage state transitions to "Patient Preparation". The latter state is distinguishable from the earlier by the patient-centered movement of the staff members. "Patient Preparation" consists of the movement of a great number of persons around the patient. Once the surgery starts the usage state transitions to "Ongoing Surgery". This state is typically characterized by small movements of a small number of persons around the OR table: the surgeon(s) and his or her assistants. Finally, when the surgery is over, the usage state transitions back to the previous states.

Figure 2.1: OR usage state transition model.

2.2 Overview

The following is an overview of the proposed system. Given a camera in an OR, and assuming we know the underlying usage state, we first detect people using an upper-body detector. Then, we estimate their position on the ground through camera calibration (Figure 2.2).

Projected ground points are used to reconstruct their trajectories over time (Figure 2.3). Finally, we extract a state model by quantifying how OR space is utilized by each person over time using Gaussian Kernel Density Estimation (GKDE) (Figure 2.4). This process is repeated to establish a model for each OR usage states.

Having learned the models, given a new input video stream, we utilize Bayesian inference to obtain the usage states of the OR (Figure 2.5). In the following, we provide further details about each of the modules of the proposed approach.

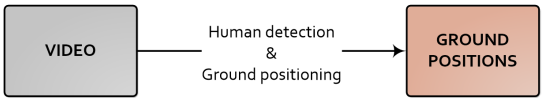


Figure 2.2: Extraction of detected ground positions from each video.

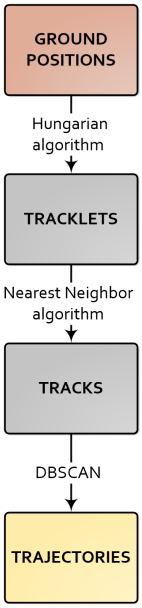


Figure 2.3: Trajectory reconstruction from ground positions.

Figure 2.4: State Model extraction.

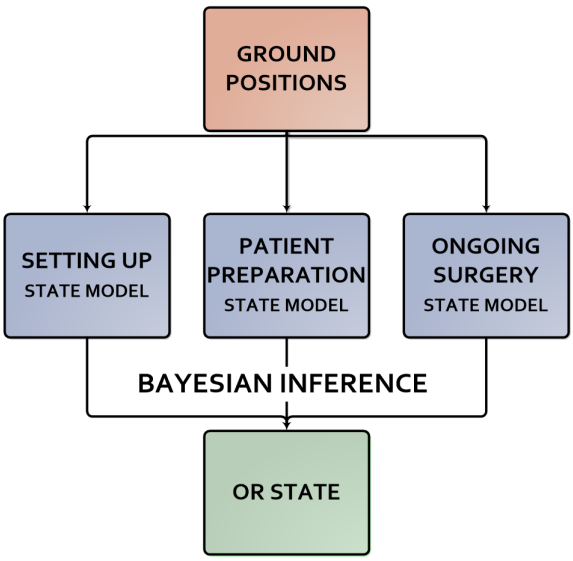


Figure 2.5: Bayesian inference using new ground positions.

Chapter 3

Detection & Ground Positioning

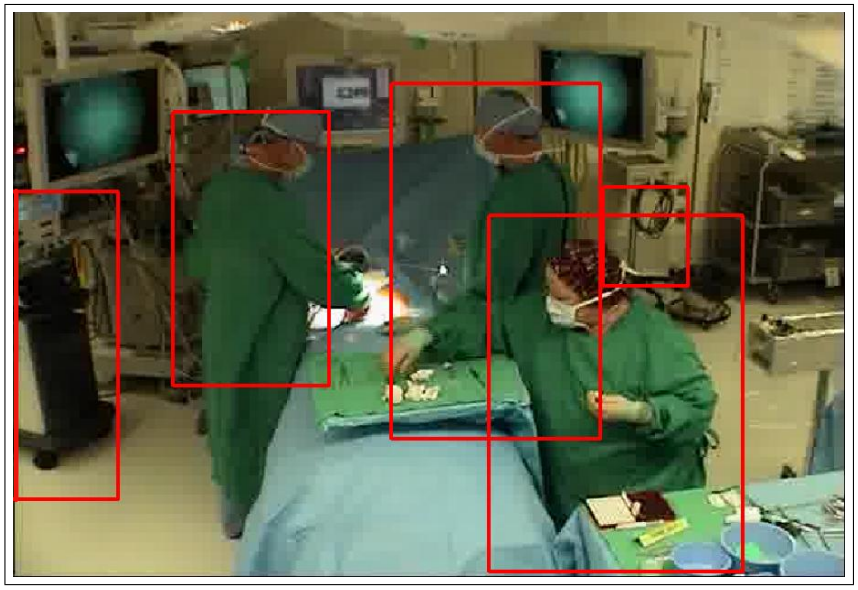
3.1 Upper-body Detection

In OR videos, feet and faces are often occluded depending on one's position and orientation. Considering an upper-body detector instead of a face detector or a human-body detector is therefore extremely relevant. Obviously, image parts or upper-body features such as gloves, masks or head protections are specific to OR environments.

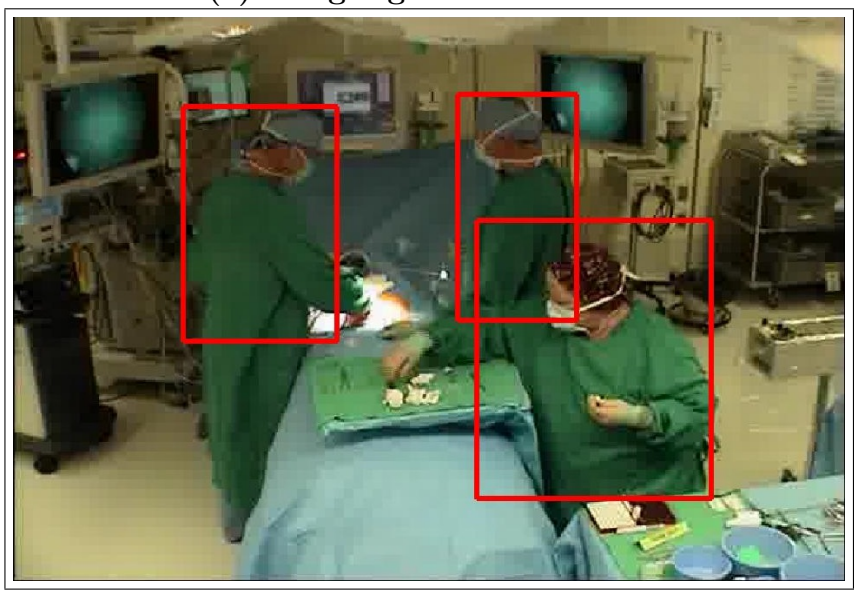
Using pre-trained human detectors for such an environment tend to be erroneous and result in large number of false or missed detections (Figure 3.1(a)). Therefore, we developed our own OR-trained upper-body model based on Felzenswalb *et al.*'s part-based detector [11, 10]. Training was done over a manually defined set consisting of 400 negative samples and 800 positive samples extracted from OR videos in each

state. We validated our trained model on over 700 images, and 420 images provided detections. A few bounding boxes defined within others appeared and were ruled out by exclusion filtering. 587 accurate bounding boxes were detected with no false detections (Figure 3.1(b)).

The overall efficiency of our upper-body detector achieved was 60%. Further, more accurate bounding boxes were detected, as can be seen in the example in Figure 3.1(b). Having obtained the detected bounding boxes in each frame, camera calibration is used to estimate their position on the ground.



(a) using a general detector.



(b) using our OR-trained upper-body model.

Figure 3.1: Upper-body Detections.

3.2 Calibration

Calibration is laborious on these kind of images. In fact, usually a calibration checkerboard is integrated in the scene. In this paper, we assume that images are obtained by perspective projection so that we can define a projection matrix and homographies. The method we use is based on Criminisi *et al.* and Hoeim *et al.*'s [7, 13] work on Single View Metrology.

3.2.1 Reference plane calibration

The reference plane is considered to be the ground (Figure 3.2). Reference plane calibration is done on a carefully selected image of an Empty OR from our dataset. Selecting an image that offers as many lines on the floor as possible was required. Lets suppose we have N corresponding points $\Big(U_i, X_i\Big)$. Let $U_i = \Big[u_i, v_i, 1\Big]^\top$ be the homogenous coordinates of point i in the image plane and $X_i = \Big[x_i, y_i, 1\Big]^\top$ its homogenous coordinates on the ground plane. We want to compute the homography matrix $\mathbf{H}$ such that:

$$\forall 1 \leq i \leq N, \quad X_i = \mathbf{H}U_i \quad \text{where} \quad \mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad \text{and} \quad h_{33} = 1 \qquad (3.1)$$

The system of $2N$ linear equations can be rewritten as follows: $\mathbf{Ah} = \mathbf{b}$, where:

– $\mathbf{h} = \Big[h_{11} \quad h_{12} \quad h_{13} \quad h_{21} \quad h_{22} \quad h_{23} \quad h_{31} \quad h_{32}\Big]^\top$ is a vector of length 8.

– $\mathbf{b} = \begin{bmatrix} x_1 & y_1 & x_2 & y_2 & x_3 & y_3 & x_4 & y_4 & ... \end{bmatrix}^\top$ is a vector of length $2N$.

– $\mathbf{A} = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -x_1v_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -x_1u_1 & -x_1v_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -x_2v_2 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -x_2u_2 & -x_2v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$ is a matrix of size $2N \times 8$

As there are 8 unknowns and $2N$ equations introduced by the corresponding points, a minimum of 4 independent points are required. In our case, we identified 6 corresponding points as defined on Figure 3.4. The problem is solved using the least square minimization method presented in Hartley and Zisserman [12]. And, we finally get:

$$\mathbf{h} = (\mathbf{A}^\top \mathbf{A})^{-\mathbf{1}} \mathbf{A}^\top \mathbf{b} \tag{3.2}$$

3.2.2 Parallel plane calibration

Our next step is to find an estimate of the image coordinate of a person's feet $\left(u_b, v_b\right)$ on the floor based on the detected upper-body bounding box. Hoeim *et al.* [13], offer a solution that allows us to get, knowing the image coordinates of a pixel that lies on the parallel plane, its corresponding image coordinates once projected on the reference plane (Figure 3.3). Defining top $\left(u_t, v_t\right)$ and bottom $\left(u_b, v_b\right)$ points of known objects allows us to retrieve the camera height (Figure 3.5). Indeed, if we know the height of the object, then the camera height y_c can be approximated as follows, where v_0 is the horizon line computed from the homography matrix $\mathbf{H}$ [7]:

$$y_c = h\frac{(v_0 - v_b)}{(v_t - v_b)} \tag{3.3}$$

We selected an image that contains one or several height references such as tables or beds. We computed several camera heights and finally computed the average to be $y_c = 2.3\ m$ which is plausible. The simplified version of the formula is generally applied to outdoor scenes where the camera tilt is very small. In our indoor situation, we get an overhead view of the scene. The camera tilt cannot be neglected and therefore the formula involves the focal length f. Various assumptions were made. First, we consider the mid-upperside of the bounding box to be the top of the head $\left(u_t, v_t\right)$. Then, we assume that the person stands straight, and their feet can be estimated along the vertical line $u_b = u_t$. We also assume that the average height of a person is $y_p = 1.65\ m$. As suggested by Hoeim *et al.* [13], we estimate the focal length as being 1.4 times the image height and we adjust v_c. Finally, we compute an

estimate of the image coordinate v_b of the person's feet as follows:

$$v_b = \frac{A + v_0 y_p}{A + y_p} \quad \text{where} \quad A = \frac{y_c}{(1 + \frac{(v_c - v_0)(v_c - v_t)}{f^2})} \tag{3.4}$$

Using camera calibration information, we project the detections in each frame to estimate the corresponding ground position. For a sample video depicting each of the distinct OR usage states, the ground projections are as shown in Figure 3.6. Now that we have the detections and the projection of the position of each person onto the floor, we want to identify the trajectory of each person as represented by the different colors in Figure 3.6.

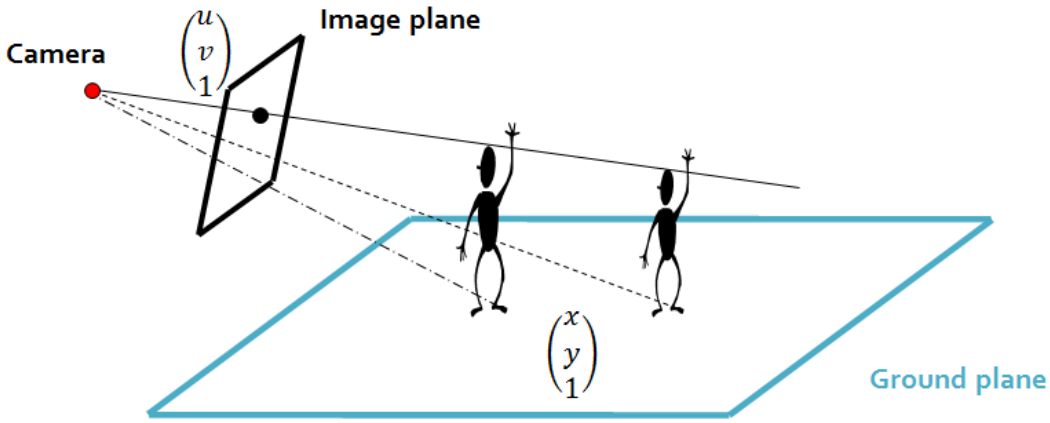


Figure 3.2: From image plane to reference plane.

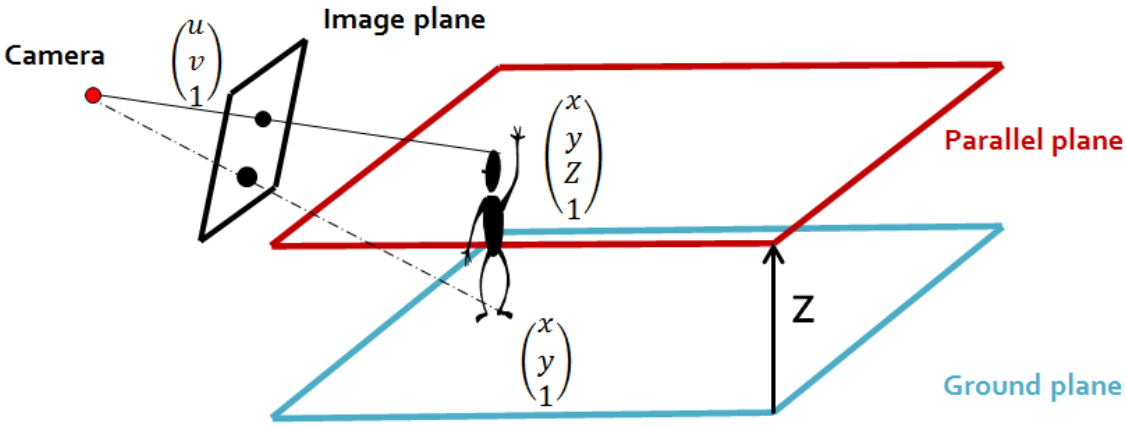
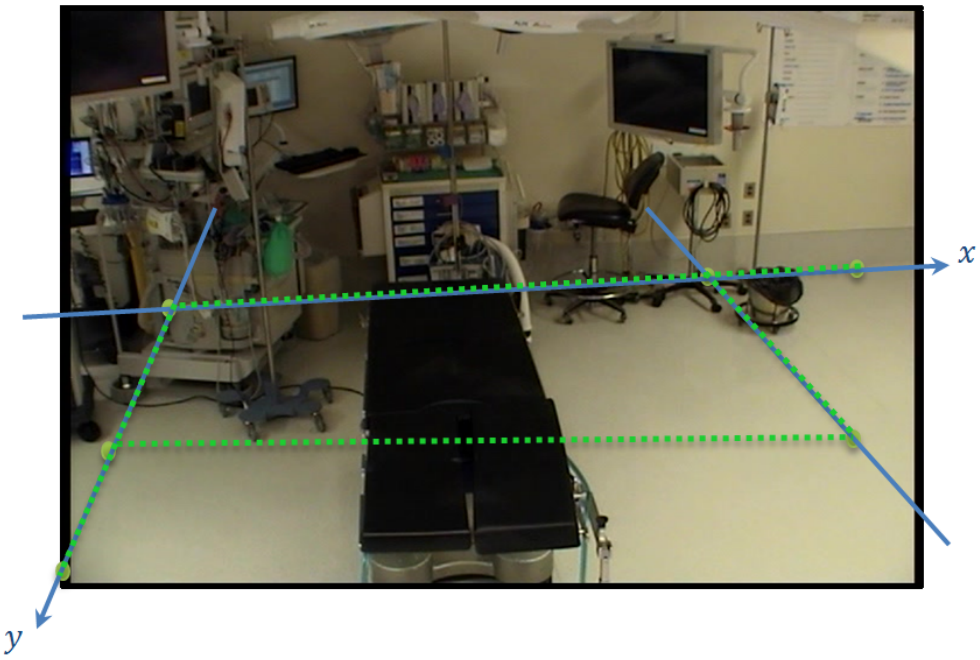
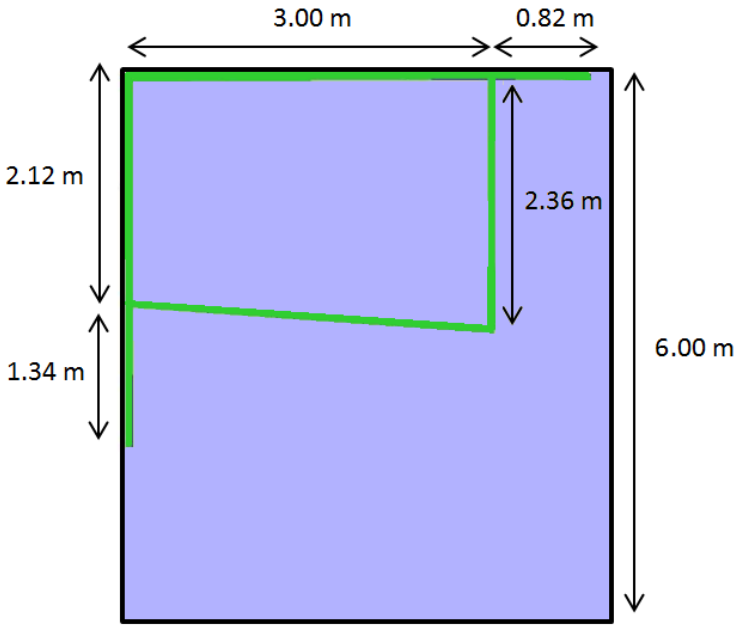


Figure 3.3: From parallel plane to reference plane.

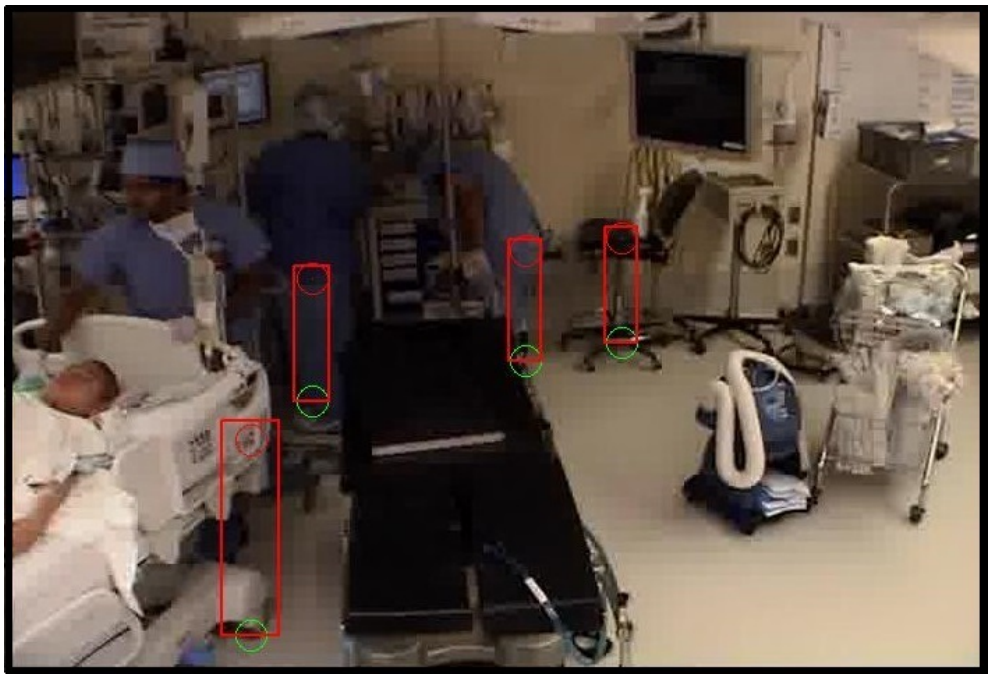


(a) Ground plane outline.

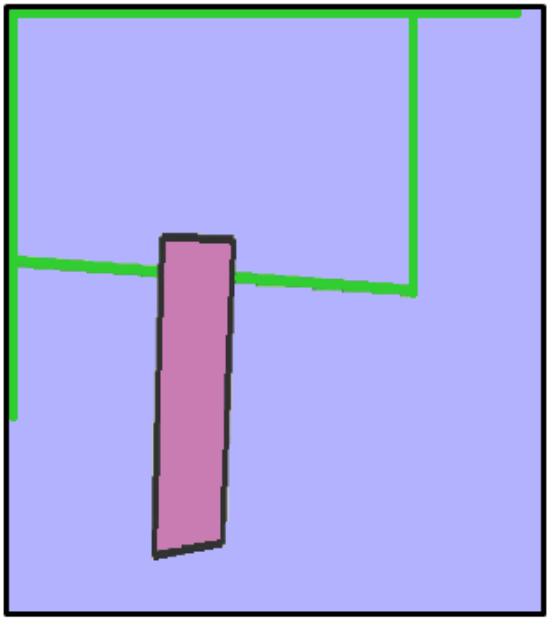


(b) OR dimensions in meters.

Figure 3.4: Reference plane calibration.

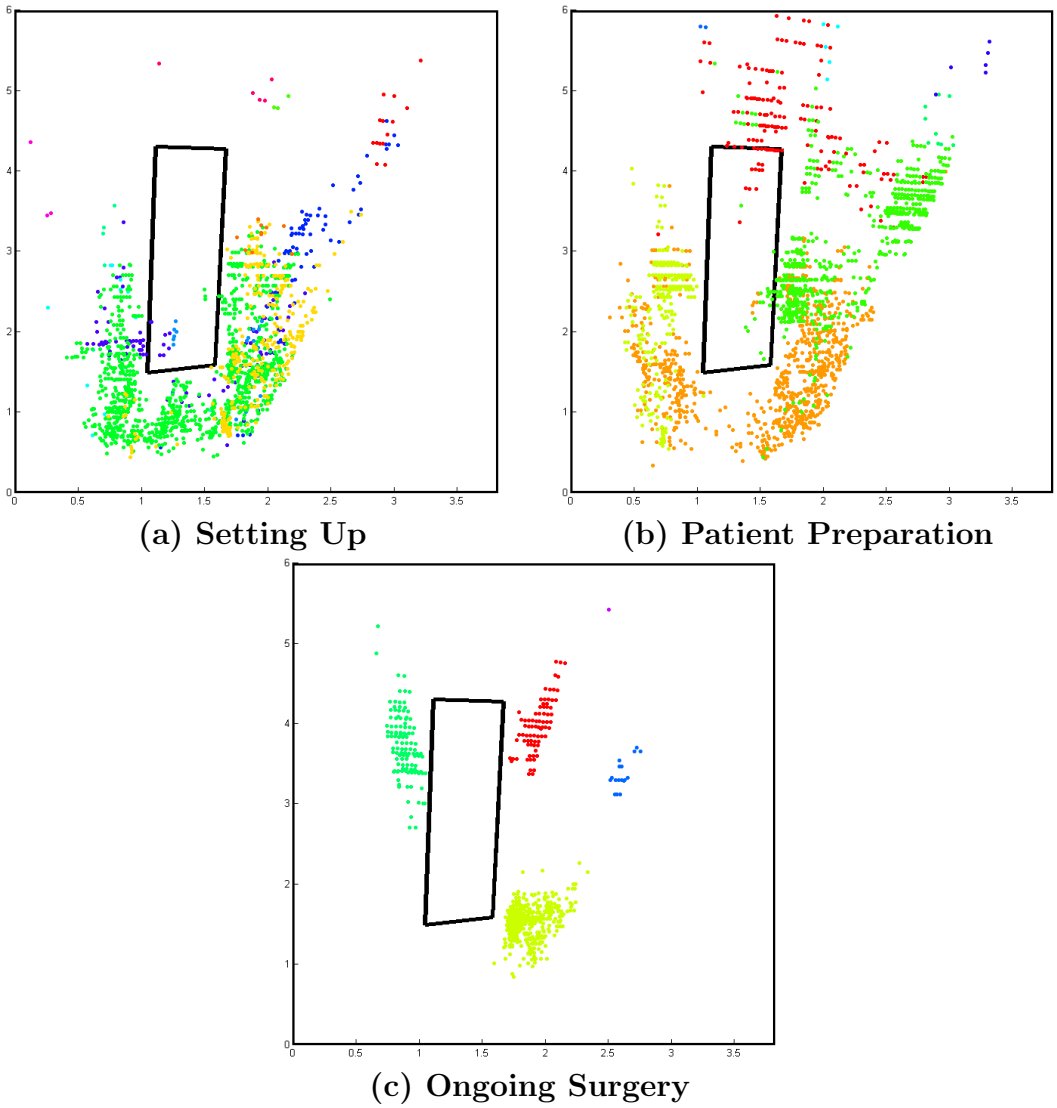


(a) Calibration using height references (bed, leg, chair).



(b) Resulting table projection onto the ground plane.

Figure 3.5: Parallel plane calibration.



(a) Setting Up **(b) Patient Preparation**

(c) Ongoing Surgery

Figure 3.6: Ground positions (each color represents different staff members).

Chapter 4

Trajectory Reconstruction

4.1 Assignment problem

In order to extract trajectories from our video, we first compute tracklets by solving a frame to frame assignment problem using the Hungarian algorithm. A mathematical formulation is presented by Pentico in his survey on assignment problems [21].

If we consider 2 consecutive frames (Figure 4.1), l with n detections and $l+1$ with m detections, we can compute a distance matrix $C = (c_{ij})$ where c_{ij} represents the distance between object i in frame l and object j in frame $l+1$. The Hungarian algorithm then solves the problem by minimizing the following objective function:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \tag{4.1}$$

under the following constraints, where $x_{ij} = 1$ if the bounding box i in frame l is assigned to bounding box j in frame $l + 1$ and $x_{ij} = 0$ if not:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1 \ ... \ m, \qquad \sum_{j=1}^{m} x_{ij} = 1 \quad i = 1 \ ... \ n \tag{4.2}$$

Hence, IDs are assigned to one or several bounding boxes as they move along time. When a bounding box is detected in one frame and is not detected in the next frame then a gap occurs (in blue Figure 4.1). We would like to avoid obtaining false new tracks. Therefore, a nearest neighbor algorithm [4] deals with gaps by linking tracks that break. Tracklet end points are linked to following tracklet start points in order to form trajectories.

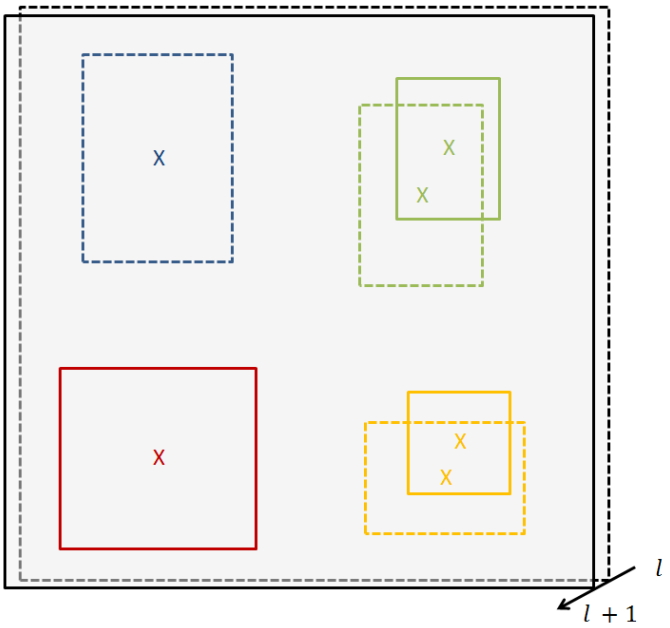


Figure 4.1: 2 consecutive frames l and $l + 1$ containing several bounding boxes.

4.2 Tracks clustering using DBSCAN

As there are mis-detections and because of the fact that people move out of the camera field of view, the data association solution can result in multiple tracklets for the same person. In order to deal with that, we further cluster tracklets using the DBSCAN clustering algorithm [9]. We've chosen this algorithm as it has a physical meaning when it comes to clustering points. In fact, DBSCAN, Density-Based Spatial Clustering, finds clusters based on density reachability. Two parameters have to be specified: $minPts$, the minimum number of points that belong to a cluster and ϵ the radius around a point that the algorithm has to look at for merging. Centroids – that is, mean positions over time of data points associated to each single track – are considered for clustering. We set the minimum number of centroids to form a cluster to be $minPts = 1$ and the radius to be $\epsilon = 0.5$ meters (Figure 3.6). This technique allows us to reduce the number of trajectories to build our state model.

Chapter 5

Models Extraction

5.1 Bivariate Gaussian Kernel Density Estimation

If we consider the 2D histogram representing the spatial distribution of points, we can account for the fact that there are areas where people stay the most or simply move through. A Kernel Density Estimator [4] provides a non-parametric estimate of the probability density function (pdf) $g_{ik}(\mathbb{X})$ over each trajectory $\mathbb{X}_k = \begin{bmatrix} x_{k1} \dots x_{kN} \end{bmatrix}$ associated to a state $\mathbb{S} = i$ as follows, where states 1, 2 and 3 are respectively "Setting Up", "Patient Preparation", and "Ongoing Surgery":

$$g_{ik}(\mathbb{X}) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{2\pi h^2} \exp\{-\frac{\left\| \mathbb{X} - x_{kn} \right\|^2}{2h^2}\} \tag{5.1}$$

Basically, x_{kn}'s are successively occupied ground position throughout time by a

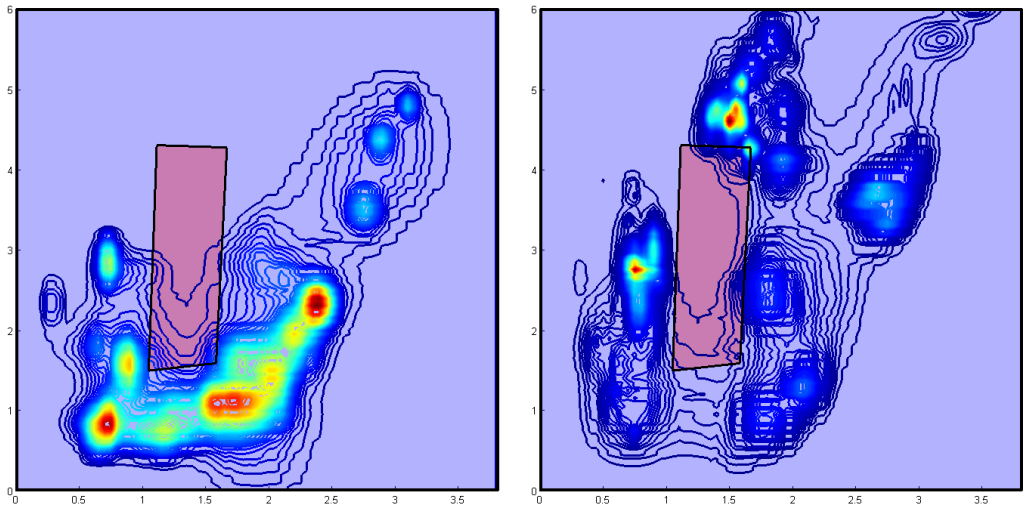
single staff member. Each and every one of them lie at the center of a hypercube (here a square) of side h to which we associate a kernel function. Choosing a Gaussian kernel function results in a smoother density model where h represents the standard deviation of the Gaussian components. The bandwidth h is selected as suggested by Bowman and Azzalini [5].

5.2 Model construction

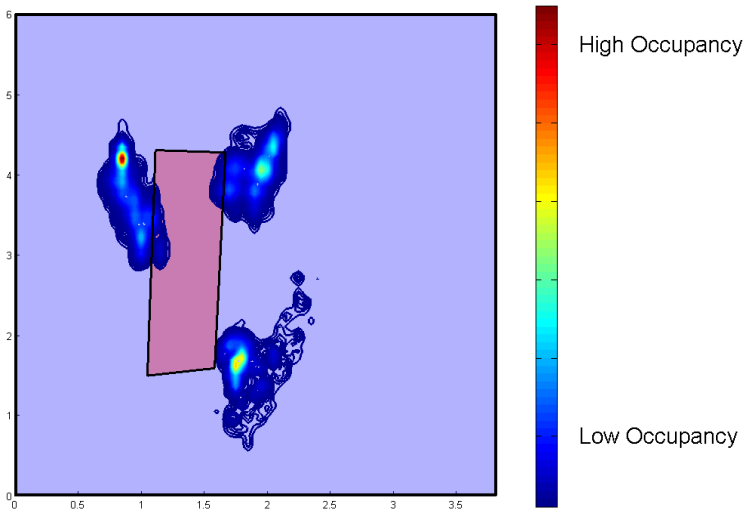
The K previously computed pdfs $g_{ik}(\mathbb{X})$ are then combined to give the pdf $f_i(\mathbb{X})$ characterizing the usage state $\mathbb{S} = i$ as follows, where $\sum_{k=1}^{K} \pi_k$ is the total number of data points associated to state $\mathbb{S} = i$ and π_k the number of points in trajectory $\mathbb{X}_k$:

$$\forall i = 1 \ldots 3, \quad f_i(\mathbb{X}) = p(\mathbb{X}|\mathbb{S} = i) = \frac{1}{\sum_{k=1}^{K} \pi_k} \sum_{k=1}^{K} \pi_k g_{ik}(\mathbb{X}) \tag{5.2}$$

We end up with models characterizing each state as seen in Figure 5.1. In Figure 5.1(a), occupancy is spread all over the room except for the upper left corner of the room due to the presence of diagnostic tools. In Figure 5.1(b), staff members tend to have a patient centered activity, and one can easily notice the anesthesiologist's position behind the OR table. Finally, in Figure 5.1(c), 2 surgeons on both sides of the OR table, as well as an assistant on the lower right (handing out surgical tools) adopt more constrained motion pattern around assigned positions.



(a) Setting Up. **(b) Patient Preparation.**



(a) OngoingSurgery.

Figure 5.1: Estimated distribution of trajectories over the three states (as seen on color bar: High Occupancy in red, Low Occupancy in blue).

Chapter 6

Experiments & Results

6.1 State prediction - Bayesian Inference

Having learned a model for each usage state, a Bayesian classifier is used for inference of a new video such that we can make a state decision based on maximum likelihood estimation. The idea is that, assuming that we know the model for each state, new temporal observations $\mathbb{X} = \mathbb{O}_t$ can be used to obtain evidence about the underlying state they characterize.

$$p(\mathbb{S} = i|\mathbb{X} = \mathbb{O}_{1:n}) \propto p(\mathbb{X} = \mathbb{O}_{1:n}|\mathbb{S} = i) \times p(\mathbb{S} = i) \tag{6.1}$$

If we assume $p(\mathbb{S} = i)$ to be same for all states i, then:

$$p(\mathbb{S} = i|\mathbb{X} = \mathbb{O}_{1:n}) \propto p(\mathbb{X} = \mathbb{O}_{1:n}|\mathbb{S} = i) \tag{6.2}$$

If we assume that observations are conditionally independent, then:

$$p(\mathbb{X} = \mathbb{O}_{1:n}|\mathbb{S} = i) = \prod_{t=1}^{n} p(\mathbb{X} = \mathbb{O}_t|\mathbb{S} = i) = \prod_{t=1}^{n} f_i(\mathbb{O}_t) \tag{6.3}$$

And finally:

$$\mathbb{S} = \underset{i}{\operatorname{argmax}}\ p(\mathbb{S} = i|\mathbb{X} = \mathbb{O}_{1:n}) = \underset{i}{\operatorname{argmax}} \prod_{t=1}^{n} f_i(\mathbb{O}_t) \tag{6.4}$$

6.2 Video database

Results presented in this paper are based on videos taken by a single camera at different moments in the same OR. All usage states of the OR are shown in Table 6.1. To ease our work, videos were converted into a set of frames with a rate of 10 frames/sec.

States	*Setting Up*	*Patient Preparation*	*Ongoing Surgery*
Time length	12min31sec	75min12sec	57min35sec
Number of frames	7510	45125	34552

Table 6.1: Video database.

In order to develop a system that recognizes the OR usage state, models discriminating each of them have to be extracted from our videos. We now want to assess the accuracy of our model. For that, we performed a 10-fold Cross Validation by considering 60% of our data for training and 40% for testing.

6.3 Training and Testing dataset definition

To extract the state models, trajectories have to be computed. Therefore, we need to consider consecutive frames as our training set. If a set of frames in our data is of length L, we randomly select an integer n in the 40^{th} percentile. Then, we consider to be our training set the following interval: $[n, \lfloor n + 60\% \times L \rfloor]$. The remaining data is then used for testing.

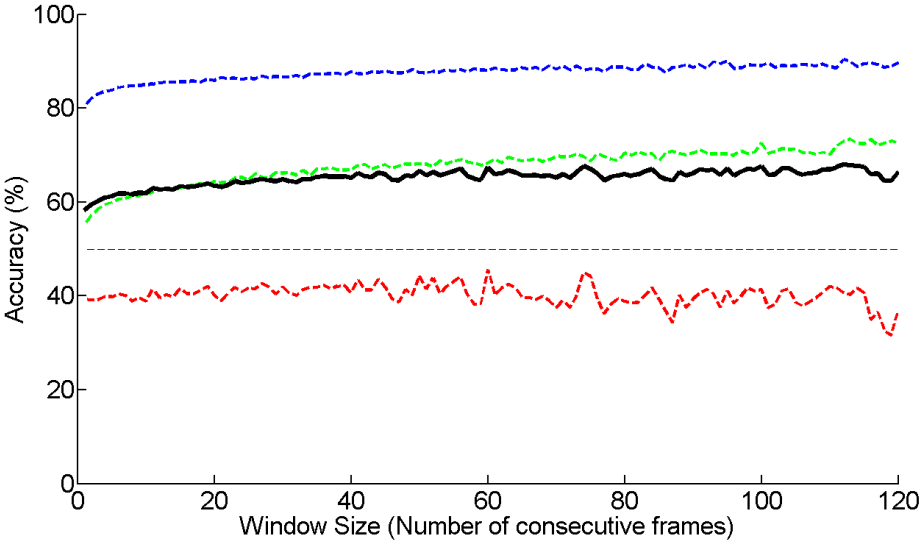
6.4 Testing

For testing, state decision is either based on data within a sliding temporal window or based on a minimum number of data points (point density). For each usage state model, we draw the accuracy of the system either as a function of the window size or the point density (Figures 6.1). Window size is defined as the number of consecutive frames after which the system makes a state decision. When looking at Figure 6.1(a), the total accuracy of our system increases with the increase in window size. Point density is defined as the number of consecutive data points after which our system makes a state decision. With approximately 85% accuracy, the "Ongoing Surgery" state has the best recognition rates whether we consider window size or point density (table 6.2). If we base state decision on point density, a significant positive slope of the "Patient Preparation" state curve appears and results in improved performances. In fact, "Patient Preparation" and "Ongoing Surgery" recognition rates tend to be similar with the increase in point density. Hence, better performance is achieved by

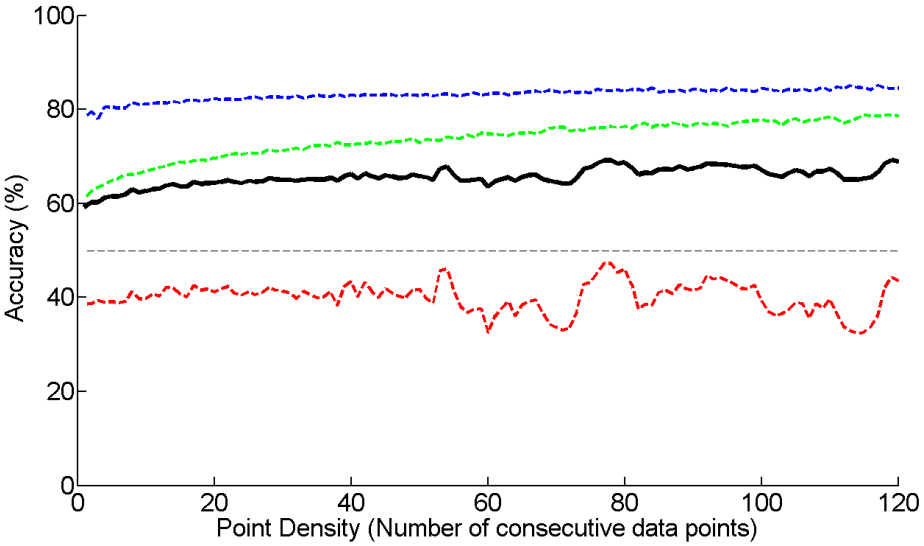
considering point density. However, our system performs poorly for the "Setting-Up" state recognition for both window size and point density. In this case, results still need to be investigated as human motion is spread out all over the room. Basically, staff members cross over wider or smaller areas but at a constant recording frame rate. Therefore, considering window size overlooks the plasticity of the occurrences of the detections that vary significantly depending on the usage state. Nevertheless, the results of the system proposed in this article are encouraging since it offers approximately 70% overall accuracy.

Optimal Value	*Setting Up*	*Patient Preparation*	*Ongoing Surgery*	*Total Accuracy*
Window size	40% (±11%)	67% (±7%)	87% (±5%)	**67%**
Point density	40% (±16%)	74% (±10%)	83% (±5%)	**69%**

Table 6.2: Usage state inference optimal accuracies and standard deviations.



(a)



(b)

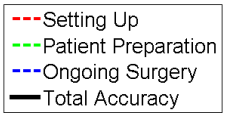


Figure 6.1: 10-fold Cross validation (60% training /40% testing).

Chapter 7

Conclusion

We presented a system that exploits existing video streams from an OR to infer the OR usage state through Bayesian modeling. We defined our OR states based on common surgical processes that are relevant for assessing OR efficiency. For this purpose, we exploited a single feature that has a physical meaning, the human motion pattern. We took advantage of a detection algorithm based on a discriminatively trained part-based upper-body model. We used a data association algorithm based solely on a geometrical feature to reconstruct trajectories. Spatial occupancy of the OR by staff members was then evaluated using a kernel based method. Finally, encouraging results were achieved by considering density of the detected features, instead of predetermined frame rates.

Future work would involve enhancing trajectory reconstruction by exploiting image features. Further, the independence assumption made in using the Bayes classifier is rather simplistic. Therefore, taking advantage of the established usage state transition models, such as a Hidden Markov Model (HMM), would be a good next step.

Bibliography

[1] H. F. M. Association. Achieving operating room efficiency through process integration. *Journal of the Healthcare Financial Management Association*, 57(3):suppl–1, 2003.

[2] J. Bardram, T. Hansen, and M. Soegaard. Awaremedia: A shared interactive display supporting social, temporal, and spatial awareness in surgery. *Proceedings of the 2006 20th anniversary conference on Computer Supported Cooperative Work*, pages 109–118, 2006.

[3] B. Bhatia, T. Oates, Y. Xiao, and P. Hu. Real-time identification of operating room state from video. *AAAI*, 2:1761–1766, 2007.

[4] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, ISBN: 0387310738, 2006.

[5] A. W. Bowman and A. Azzalini. *Applied Smoothing Techniques for Data Analysis*. Oxford University Press, ISBN: 0191545694, 1997.

[6] S. J. Ciechanowicz and N. Wilson. Delays to operating theatre lists: Observations from a uk centre. *Internet Journal of Health*, 12(2), 2011.

[7] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000.

[8] R. J. Does, T. M. Vermaat, J. P. Verver, S. O. R. E. N. Bisgaard, and J. Van den Heuvel. Reducing start time delays in operating rooms. *Journal of Quality Technology*, 41(1):95–109, 2009.

[9] M. Ester, H. Kriegel, S. J., and X. X. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD*, 96:226–231, 1996.

[10] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[12] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[13] D. Hoeim, A. Efros, and M. Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008.

[14] B. Kodali, D. Kim, R. Bleday, H. Flanagan, and R. Urman. Successful strategies for the reduction of operating room turnover times in a tertiary care academic medical center. *Journal of Surgical Research*, 187(2):403–11, 2014.

[15] P. M. Lange, K. L. G. Nielson, and S. T. Petersen. *Phase recognition in an operating room using sensor technology*. Masters Thesis, IT University of Copenhagen, 2010.

[16] LiveData and the NYP Wall of Knowledge. Or-dashboard. http://www.livedata.com/healthcare-solutions/education-and-research/or-dashboard-at-nyp/.

[17] A. Macario. What does one minute of operating room time cost? *Journal of Clinical Anesthesia*, 22(4):233–6, 2010.

[18] A. Nara, K. Izumi, H. Iseki, T. Suzuki, K. Nambu, and Y. Sakurai. Surgical workflow monitoring based on trajectory data mining. *New Frontiers in Artificial Intelligence*, 6797:283–291, 2011.

[19] Q. Niu, Q. Peng, T. El Mekkawy, Y. Tan, H. Bruant, and L. Bernaerdt. Performance analysis of the operating room using simulation. In *Proceedings of the Canadian Engineering Education Association*, 2011.

[20] N. Padoy, D. Mateus, D. Weinland, M.-O. Berger, and N. Navab. Workflow monitoring based on 3d motion features. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 585–592. IEEE.

[21] D. W. Pentico. Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176(2):774–793, 2007.

[22] M. Schuster, M. Pezzella, C. Taube, E. Bialas, M. Diemer, and M. Bauer. Delays in starting morning operating lists: An analysis of more than 20 000 cases in 22 german hospitals. *Deutsches Arzteblatt International*, 110(14):237, 2013.

[23] G. Veres, H. Grabner, L. Middleton, and L. Van Gool. Automatic workflow monitoring in industrial environments. In *Computer Vision–ACCV 2010*, pages 200–213. Springer, 2011.

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