Towards Automated Activity Recognition in an Intensive Care Unit

Colin S. Lea[1], James C. Fackler[2], Gregory D. Hager[1], and Russell H. Taylor[1]

[1] Department of Computer Science, Johns Hopkins University, Baltimore 21218, USA
{clea1@jhu.edu, hager@cs.jhu.edu, rht@jhu.edu}
[2] Johns Hopkins Medical Institutions, Baltimore, Maryland 21224, USA
{jim@jhmi.edu}

Abstract. In this work we develop a new system for automated activity recognition in an Intensive Care Unit. Our system is capable of recognizing a set of 7 high-level activities using only 3D perception. We take a three step approach to classifying actions from depth images. First, temporal action sequences are created by segmenting and tracking people throughout several hours of video footage. Second, for each sequence we derive features based on position, orientation, and multi-person interactions. Lastly, a comparative evaluation is provided for Support Vector Machine and Decision Forest classification techniques. We achieve 75% overall accuracy on data collected at a Pediatric ICU and show that this approach could realistically be used for increase safety and efficiency.
Keywords: Activity recognition; Task Analysis; 3D perception

1 Introduction

An Intensive Care Unit (ICU) is a hectic place. Dozens of staff members come in and out on a regular basis and perform a large number of small but important tasks necessary for a patient's proper recovery. Estimates from doctors at our hospital's Pediatric ICU indicate that there may be around 800 micro-tasks completed during the average patient's stay. Enumerating all of these tasks is a problem of it's own and evaluating when an individual task is completed provides additional complications. Assigning too many intense tasks in a short amount of time results in high nurse cognitive load which has negative effects on work quality [1, 2]. Some typical tasks include giving medicine, checking diagnostics, inserting IVs, and performing arterial-line insertions. While monitoring tasks is important, it is not practical to do so unless the job can be automated.

In this work we develop a task-recognition system using a depth sensor that can classify up to seven actions such as performing minor procedures, checking diagnostics, and ventilator use. We take a discriminative classification approach using a set of 17 features and compare evaluation with Support Vector Machines and a Decision Forest. For a list of classified actions see Table 1 in Section 5.

The particular problem we focus on is retrospective analysis in an ICU. Our goal is to detect personnel actions throughout the course of a patient's stay. This is helpful for a number of reasons. Safety can be increased by checking

if certain activities are completed, such as giving medicine at the appropriate time. Workflow can be optimized by determining which activities are the most time consuming and then spreading them out. Sometimes multiple nurses do separate activities in a room at the same time. It may be more efficient to have several actions be completed by the same person. By monitoring the needs of each patient, the hospital can efficiently allocate the number of staff members and resources on site. This could prevent problems of over- or under-staffing. Additionally, the quality of the nurses could be measured to determine how well they work with different types of patients. This could increase the quality of care by partnering nurses with specific patients that better suit their skills.

Video-based activity monitoring has become a prevalent research topic over the past couple of decades. In recent years there has been interest in translating this to a surgical setting. Lo provided early work on assessing surgical skill for Minimally Invasive Surgery by tracking tissue and instrument interactions [3]. More recent work in both surgical and more general settings have relied on time series models. Duong and Kasteren use variations on Hidden Markov Models to accomplish activity recognition in a home setting [4, 5]. Padoy developed Workflow HMMs to analyze series of activities during a surgery [6]. While his approach worked well in an Operating Room, we don't think it is as appropriate for the ICU because of the actions taken place. In the ICU there are a large number of independent tasks spread across the room while in the OR activities are often sequential and are more focused on the patient. While there has been recent success in activity recognition using bag of word type approaches, for example using space-time interest points (STIP) [8, 9], we don't think this type of approach is as conducive to our setting. STIP features generally do not take into account spatial characteristics or scene information. For example, we can use features like the proximity of humans to specific medical instruments in our model. For other activity recognition techniques see [10].

Our approach tends away from these time-series and bag of words methods and relies on discriminative models such as Support Vector Machines and Decision Forests for activity recognition. We have been influenced by the likes of Stikic who demonstrates a semi-supervised multi-instance learning method with SVMs for classification using on-body sensors [7].

We believe our problem can be accomplished using 3D perception techniques and, in particular, using an Xbox Kinect. This device is capable of generating depth and color images. It is more cost effective and provides better pixel coverage than most stereo or time of flight camera systems. Note that restrictions from the Internal Review Board prevent us from using the color video data outside of the ICU. While it is possible to extract histograms or other features from the color data, for this study we only use depth images.

Our pipeline takes the following approach, as shown in Figure 1. In the ICU, we record several hours worth of footage using an Xbox Kinect. Next, we track personnel and use scene analysis techniques to generate derived signals. These signals are encoded into a set of 17 features used for activity recognition. Using the depth images we are able to extract people using a background subtrac-

tion technique and then extract features such as body position, velocity, and orientation. *Action sequences* are formed by tracking people over time. These are defined by the set of features corresponding to each individually segmented person over a set of frames. An *event* is more generally defined by the actions taking place in this particular sequence. For example, one event could be a nurse walking into the room, checking a patient's vitals, and walking out of the room. The prominent action in each event is then classified based on the corresponding feature vector from its action sequence.

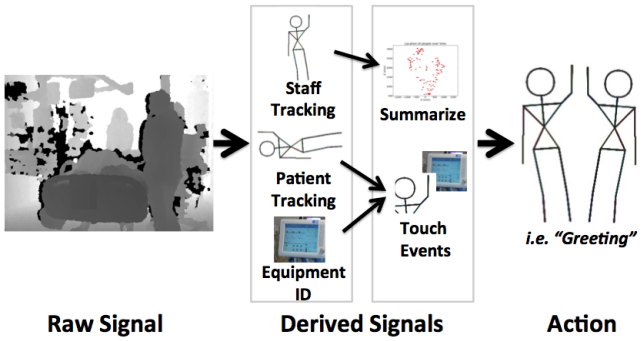

Fig. 1. The pipeline includes three key elements: (left) recording data from depth cameras at the ICU, (middle) extracting features based on cues from our body tracking, and (right) classification of each sequence into high-level actions.

2 Segmentation and Tracking

Before we can classify each task we must generate a set of action sequences from the depth video. These sequences are created using a two step process. First, people are segmented using a straight forward background subtracting technique. Second, each person in the scene is tracked over time. Our person segmentation process includes four steps: (1) perform background subtraction by differencing a background model with the current frame, (2) compute an outline of the people based on large gradients in the image, (3) use a set of morphological filters, and (4) extract the connected-components that are above a certain pixel-count threshold.

The background model is generated by averaging 10 frames without any people in them. To reduce noise, these images go through a preprocessing step that fills in any holes in the image using a nearest neighbor technique (see the black holes on the left side of Figure 2). Segmentation results are shown in Figure 2 where the orange blobs are segmented people.

Each individual is tracked over time and put into their own activity sequence. Due to occlusions in the images, for example if one person walks in front of

another, tracking isn't a trivial problem. A person may be in the scene but could be missing from the image for a few seconds. For each new image we look for the closest set of matches using the euclidian distance of the segment's center of mass in 3D between the previous frame and the current frame. If there isn't a person that is close enough then the tracker looks back to the previous few frames. In practice new segments are added to sequences if their distance to a previous person is less than 0.5 meters and was last seen within 5 frames.

Fig. 2. Personnel are segmented from the depth images and placed into individual action sequences.

It is worth mentioning results using the Microsoft skeleton tracker included with the Kinect OpenNI drivers. In good conditions this tracker will output a set of joints corresponding to positions of body parts such as the head, chest, hands. Our results in Figure 3 show that performance in the ICU is poor. The skeletons are commonly miss-configured, especially if a user walks into the scene sideways. This is especially problematic if the sensor is placed perpendicular to the patient's bed because personnel tend to look towards the patient and not towards the camera. Additionally, large objects such as the curtain in the left side of the image is perceived as a person. In the ICU this is problematic because new medical equipment is sometimes brought into the room. Ultimately, only a small percentage of the skeletons detected were accurate.

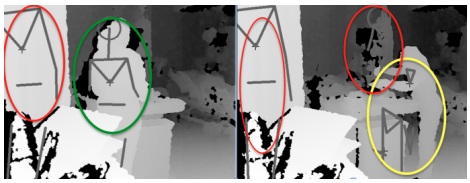

Fig. 3. Examples of skeletons generated from the Microsoft tracker using OpenNI. Red means a bad skeleton, green means good, and yellow means correct detection but bad pose estimation

3 Feature Extraction

In our approach, each event is labeled based on the prominent action taking place. Despite the changing dimensionality of the scene's feature-space – a result of people coming in-and-out of the room – we develop a finite set of features based on statistics on each action sequence. Based on our knowledge of the data, we know that personnel sometimes work together on tasks, thus our features take into account interactions between multiple people. For example, there are times in our dataset when two people are working together to use the ventilator. It is also common for nurses and parents to talk together around the patient's bed. We developed four types of features for our system: summary statistics, virtual touch sensors, orientation-based, and interaction-based.

Summary Statistics: Each time step of an action sequence includes two important pieces of information: center of mass in 3D and current time. We use these to form three useful features that summaries activity: average center of mass, path length, and path velocity. Path length is calculated as the integral of changes in center of mass over time and path velocity is simply the path length divided by the sequence duration. Note that in practice we also tried using the frame count. We saw in our results that this had an especially low importance weighting in our decision tree so it was removed from the system.

Virtual Touch Sensors: Looking through the data it is apparent that one of the distinguishing factors between whether a nurse is performing a procedure or checking vitals is how long he or she spends at either the head or foot of the patient. This is because personnel insert medicine into tubes that are on the bottom half of the patient's body. Using this and other domain knowledge, we strategically place virtual touch sensors within the room. These are manually placed sensors that detect if an individual is nearby. For example one could be triggered if a doctor touches the patient's head. In practice we use two on the patient, one on the ventilator, and one at the staff computer. We experimented with two methods of triggering these. In the first, they were activated if the bounding box of a personnel comes within a certain radius of the sensor. In the second, we simply recorded the closest distance the staff got with each sensor. The first acts as a binary sensor whereas the second uses real numbered values. Ultimately, the latter gave significantly improved results. Effectively in the latter our classifier is learning a cutoff value for the sensor instead of relying on a preset value. The circles in Figure 4 depict the locations of two of the touch sensors.

Histogram of Orientations: The orientation of a person can be very useful because it gives an estimate of where they are looking and what they are doing. For instance, a nurse could be in the same position when doing a procedure or looking at the video monitors. However, their orientation may be different. We have found that we can calculate a rough estimate of orientation by using Principal Components Analysis on the segmented 3D data in each frame. The output of this technique is a set of 3 orthogonal directions corresponding to the orientation. The vector associated with the largest eigenvalue is generally directed upwards. This is because there is the most variation in the vertical axis.

The second eigenvalue is generally the direction the person is facing. This is converted to an angle about the upward axis for convenience in feature-space.

To include the orientation as a set of finite feature variables we calculate the histogram of orientations over the activity sequence. In practice we see that the vectors may point in the same direction if the person is facing either backwards or forwards. To alleviate this error we create a semi-circular representation of angle where we merge the data from 0-180 and 180-360 degrees. Thus, forwards and backwards are the same. In practice we use use 12 bins from 0 to 360 degrees in our original histogram and merge these into 6 bins from 0 to 180 degrees. Figure 4 depicts this feature.

Interaction Coefficient: Multi-person interactions are modeled using the orientation estimates with the assumption that people look at each other when they interact. While this may not be valid for all interactions, we see in Figure 5 that this feature does have a positive impact on our results. We calculate the projection of each person's forward-looking vector against all others currently in the room, as shown in Figure 4. The average of each projection within a sequence is used as a feature and is mathematically described as follows where f is a feature, N is the number of people in the room, t is time, v is the orientation vector, and c is the current sequence,

$$f_{i=1..N} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{v_c}^t \cdot \boldsymbol{v_i}^t \tag{1}$$

Virtual Touch Sensors Orientation Estimation Interaction Coefficient

Fig. 4. (Left) Spherical virtual touch sensors are used to see if a nurse is interacting with the upper or lower part of a patient's body (Middle) A segment and its corresponding orientation vectors (Right) The forward vectors for two segments in a scene

4 Recognition

In our approach, the feature vector from each action sequence is classified independently. Looking at the data using PCA and manifold learning techniques it is apparent to us that the data is not linearly separable in a low-dimensional space. Thus, we evaluate our data using Support Vector Machines and Decision Forests

classification techniques. The SVMs are motivated by positive results in this area from Stikic [7] and others. Decision Forests have shown to provide good results in other areas of computer vision like object recognition and segmentation [11].

A Support Vector Machine (SVM) uses the idea that the data may be separable in a higher dimensional state. The SVM finds the optimal hyperplane that separates each class based on combinations of each input variable. In the multi-class case there are two ways of using an SVM. The first is called the one-versus-all method which uses one classifier per class and trains on all of the data. It has two labels: class and not-class. This means there are generally many more not-class points than class points. The second method uses pair-wise classifiers. There are a total of $N \cdot (N-1)$ classifiers where N is the number of classes. An SVM is fit between every class and every other class. The final classification is generally done by using the class that wins the most tests. In our preliminary experiments the pairwise method achieved significantly superior results.

A Decision Forest generates a large number of simple decision trees where the nodes in the tree are randomly picked features. This technique has garnered a lot of attention due to its success in many areas including computer vision [11]. An extension of this method, Extremely Randomized Trees [12], was used due to its computational efficiency. Both SVM and Forest methods are available using the SciKit-Learn Python library [13].

5 Evaluation

5.1 Experiment

Data was recorded in a single-patient room at our hospital's Pediatric ICU (PICU) with two Kinects. One device was placed close to the patient's bed to detect near-range actions and the other was setup farther from the bed to detect motion within the whole room. There were many different people coming in and out of the room including nurses, parents, and other personnel. According to the nurses, our dataset is representative of a typical day in the ICU.

Data was collected using a dynamic frame rate recorder to reduce the amount of data captured. A compressed set of data requires 216 Gb of space per hour per Kinect when running at 30 frames per second. Our recorder only captures when there is motion in the room to decrease the storage space. We record uncompressed at a rate of about 10 FPS if there is motion in the room and otherwise 1 frame every 3 seconds. This equates to between 20-135 Gb per hour.

Our classification experiments are performed on a single dataset of depth images spanning a period of 5.5 hours in the PICU during the early afternoon. A total of 122 action sequences were extracted totaling 2 hours and 12 minutes worth of interactions. Each sequence was hand annotated with the help of an attending doctor at the PICU. A set of 7 actions was used with sample counts ranging from 5 to 47 sequences per label, as noted in Table 1. Validation was done with a randomized leave-one-out strategy. For each of the 100 iterations, one sequence from each class was removed for training. This left-out set was then

evaluated using pairwise SVMs and a Decision Forest. The best results were achieved using a radial basis function kernel for the SVM with a regularization value, C, of 100. The Decision Forest classifier obtained the best results with 200 estimators and 17 max features per tree.

5.2 Results and Discussion

Based on our preliminary results, the Decision Forest was able to provide better results per-class (65%) and averaging over all events (75%) than the SVM's 50% and 55%. For reference, guessing the class by chance results in an accuracy of 14%. Table 1 shows per-class accuracies for each classifier. The Forest achieves higher accuracy in 6 of 7 classes. It is interesting that the SVM is able to classify the "Ventilator" much more accurately than the Forest. We find that by increasing the number of estimators in our forest we see that the per-class and overall averages start to converge a little bit higher than the current per-class average.

Actions	# Samples	SVM Accuracy	DF Accuracy
Documentation	11	62%	**71%**
Observing	47	60%	**88%**
Checking Diagnostics	28	60%	**89%**
Procedure	14	52%	**55%**
Urinary Catheter Removal	5	63%	**80%**
Ventilator Use	9	**40%**	27%
Other	8	15%	**43%**
Average Per-class		50%	**65%**
Average Overall		55%	**75%**

Table 1. Action set and corresponding per-class accuracies for the Decision Forest

Figure 5 shows the class confusion matrix based on the results from the Decision Forest. It is apparent that most miss-classified actions are labeled as "observing." This is likely a result of the large number of "observing" samples in the training data when compared to the rest of the classes. It comprises 39% of the samples but 14% of the classes. The second most common miss-classified label is "Checking Diagnostics" which is the second most common occurrence (23% of the data). This suggests that other techniques should be explored that compensate for skew in the training sample sizes.

One of the advantages of the Decision Forest is its ability to show the importance of each input feature. Figure 5 shows a 17-dimensional chart with the importance of each feature in the Forest's classification. We see that the virtual touch sensors have the greatest importance followed by the path length and center of mass. This makes sense because the touch sensors offer insight into what parts of the scene the individual is interacting with. While center of mass is important, in our analysis there appears to be a non-linear relationship between

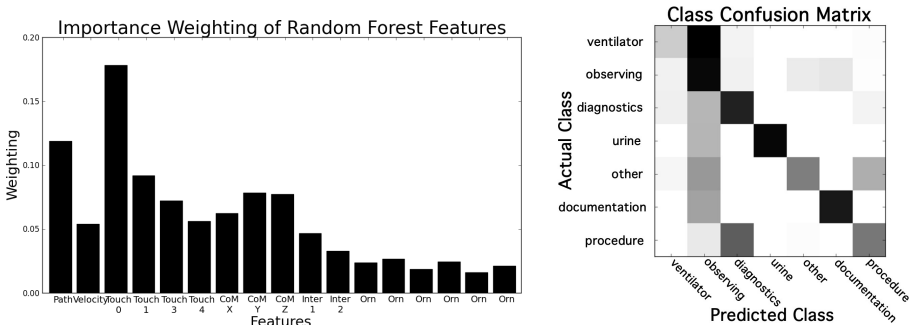

Fig. 5. (left) Feature importance (right) Confusion matrix using a Decision Forest.

position and action. This is probably why the touch sensors are more important because they give a more distinguishing sense of position. The problem is that they also require more knowledge about the room and the instruments that the personnel are interacting with.

Experimentally we see that only two interaction coefficients are useful in our classification model. This implies that the maximum number of people interacting is three. On occasion we see that there are four people in the room, but this does not have a large enough effect on the data. For reliability reasons the projections are sorted in descending order such that the person whom one is interacting with most gets added as the first feature.

Other variables such as frame count and our other touch sensor model were also evaluated but resulted in especially low importances weights. We believe the frame count importance was low because the same actions can take very different amounts of time. For example, sometimes a person will observe for 20 seconds and other times they will be around for several minutes. We obtained better results by taking these variables out than leaving them in. Due to the randomized tree structure it is better to have slightly fewer features that are much stronger than more features that have low weight.

6 Conclusion

Our preliminary results show that automated activity analysis using 3D vision techniques has the potential to be used in an everyday setting to increase safety, help efficiency, and optimize workflow. While the current system acts retrospectively, this solution may be modified to work in real-time for problems like ensuring task completion or anomaly detection.

While we recorded with two cameras, we found that the camera near the patient's head was not very helpful. It was able to detect when a staff member was near the bed but little else. Part of our future work is to distinguish body parts and potentially do gesture analysis using this data. We will then register the

cameras together so we can track individuals throughout both cameras. This may give better fidelity for differentiating between "Diagnostics" and "Procedures" as well as classifying different types of procedures.

Other methods of classification may lead to different results and types of analysis. For example, if a label was generated for each time step within an action sequence it could describe all activities taking place instead of just the prominent ones for each event. This may also be better at isolating the sub-actions. For example in a sequence where a nurse enters, observes, checks diagnostics, then leaves, the labels could include both observing and checking diagnostics.

While our results are not high enough for clinical use, we think that with further improvements a similar system could realistically be used in the ICU.

Acknowledgments: This research was funded in part by a fellowship from Intuitive Surgical and in part by Johns Hopkins University internal funds. Some equipment was provided by the Johns Hopkins University Applied Physics Lab.

References

1. P. Carayon and M. Wall, "Impact of Performance Obstacles on Intensive Care Nurses Workload , Perceived Quality and Safety of Care , and Quality of Working Life," *Health Services Research*, pp. 422–443, 2008.
2. Y. Donchin, D. Gopher, M. Olin, and Y. Badihi, "A look into the nature and causes of human errors in the intensive care unit," *Critical Care Medicine*, pp. 1–9, 1995.
3. B. Lo and A. Darzi, "Episode classification for the analysis of tissue/instrument interaction with multiple visual cues," *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2003.
4. T. V. Duong, H. H. Bui, D. Q. Phung, S. Venkatesh, R. Ave, and M. Park, "Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model," *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
5. T. V. Kasteren, A. Noulas, G. Englebienne, and B. Krose, "Accurate Activity Recognition in a Home Setting," in *UbiComp*, 2008.
6. N. Padoy, "Workflow Monitoring based on 3D Motion Features," in *International Conference on Computer Vision (ICCV)*, 2009, pp. 585–592.
7. M. Stikic, D. Larlus, S. Ebert, and B. Schiele, "Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors." *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2521–2537, Feb. 2011.
8. C. Schuldt and I. Laptev, "Recognizing human actions: A local SVM approach," *Pattern Recognition, 2004.*, vol. 00, no. C, pp. 3–7, 2004.
9. M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, Jun. 2009, pp. 1948–1955.
10. J. Aggarwal and M. Ryoo, "Human activity analysis," *ACM Computing Surveys*, vol. 43, no. 3, pp. 1–43, Apr. 2011.
11. G. Brostow and J. Shotton, "Segmentation and recognition using structure from motion point clouds," *Proceedings of ECCV*, pp. 1–15, 2008.
12. P. Geurts, D. Ernst, and L. Wehenkel, "Extremely Randomized Trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, Mar. 2006.
13. F. Pedregosa, R. Weiss, and M. Brucher, "Scikit-learn : Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.