A Cognitive Assistive System for Monitoring the Use of Home Medical Devices

Yang Cai
SCS, Carnegie Mellon University, USA
caiyang@cs.cmu.edu

Yi Yang
SCS, Carnegie Mellon University, USA
yiyang@cs.cmu.edu

Alexander G. Hauptmann
SCS, Carnegie Mellon University, USA
alex@cs.cmu.edu

Howard D. Wactlar
SCS, Carnegie Mellon University, USA
hdw@cs.cmu.edu

ABSTRACT

Despite the popularity of home medical devices, serious safety concerns have been raised, because the use-errors of home medical devices have linked to a large number of fatal hazards. To resolve the problem, we introduce a cognitive assistive system to automatically monitor the use of home medical devices. Being able to accurately recognize user operations is one of the most important functionalities of the proposed system. However, even though various action recognition algorithms have been proposed in recent years, it is still unknown whether they are adequate for recognizing operations in using home medical devices. Since the lack of the corresponding database is the main reason causing the situation, at the first part of this paper, we present a database specially designed for studying the use of home medical devices. Then, we evaluate the performance of the existing approaches on the proposed database. Although using state-of-art approaches which have demonstrated near perfect performance in recognizing certain general human actions, we observe significant performance drop when applying it to recognize device operations. We conclude that the tiny action involved in using devices is one of the most important reasons leading to the performance decrease. To accurately recognize tiny actions, it's critical to focus on where the target action happens, namely the region of interest(ROI) and have more elaborate action modeling based on the ROI. Therefore, in the second part of this paper, we introduce a simple but effective approach to estimating ROI for recognizing tiny actions. The key idea of this method is to analyze the correlation between an action and the subregions of a frame. The estimated ROI is then used as a filter for building more accurate action representations. Experimental results show significant performance improvements over the baseline methods by using the estimated ROI for action recognition.

MIIRH'13, October 22, 2013, Barcelona, Spain.

http://dx.doi.org/10.1145/2505323.2505334.

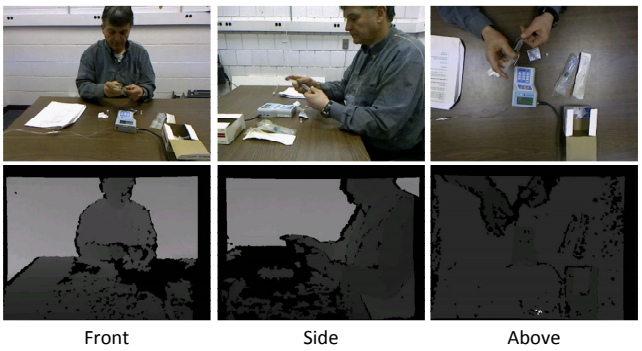

Figure 1: Examples of data recorded in PUMP database. The first row is the RGB data while the second row is the corresponding depth data. All user operations were captured by 3 Kinect cameras from different views.

Categories and Subject Descriptors

C.3 [**Computer Systems Organization**]: Signal Processing Systems

Keywords

Multi-camera, RGB+Depth, Tiny Action Recognition, Kinect, Database

1. INTRODUCTION

Home medical devices (e.g. infusion pumps, inhaler, nebulizers, etc.) which are used by patients at home on their own, are becoming more and more prevalent, due to their cost-saving advantages. However, non-professional users, especially for elderly people with cognitive decline, may sometimes wrongly operate a medical device (e.g. not following required operating procedures). More seriously, the device-use error can lead to fatal results. For example, according to [9], during 2005 to 2010, 710 deaths linked to the use of one kind of home medical devices, the infusion pumps, which intravenously deliver life-critical drugs, food and other solutions to patients. Therefore, it's critical to have some external mechanisms to supervise patient's use of these devices and keep the use-error from happening. One straightforward

solution to this problem would be let a professional person play the supervisor's role. However, because it contradicts to the main objective of home medical devices, i.e. reducing the cost, this solution is obviously infeasible. Instead of using "expensive" human's supervision, in this paper, we propose a cognitive assistive system whose objective is to automatically monitor the use of home medical devices.

The cognitive assistive system has two-fold functionalities: perception and recognition. On one hand, the system should be able to perceive user's operations. Since various advanced sensors been developed in the past decades, we are able to perceive user's operations from many different aspects. For example, the Kinect[1] that was invented recently can provide us not only the RGB information but also the depth that cannot be captured by traditional cameras. On the other hand, another important requirement for building a successful cognitive assistive system is to be able to recognize the operations so that use-errors can be identified. However, unlike the well-developed perception module, it is still unknown whether the current techniques are mature enough to fulfill this requirement. Even though many techniques have been proposed to recognize human actions [1, 12, 15, 11, 5], none of them has been applied to recognize operations involved in using home medical devices and therefore, we are not sure if they are adequate in such scenario.

Since the lack of corresponding database is the main reason causing the situation, we construct a database (called PUMP) which was specially designed for studying the use of home medical devices and present it in this paper. Particularly, we take the example of patients using an infusion pump as a typical type of home medical device operation for collecting this database. An infusion pump is a device that infuses fluids, medication or nutrients into a patient's blood stream, generally intravenously. Because they connect directly into a person's circulatory system, infusion pumps are a source of major patient safety concerns. Because of this significant impact, we used an infusion pump as the sample device. To collect the data we first define an operation protocol for correct use of an infusion pump. Then, each user was asked to simulate the use of infusion pump for several times. Their operations were recorded by 3 Kinect cameras from different views as shown in Figure 1. 17 volunteers participated in data collection and 68 multi-view operation sequences were generated respectively. The operation sequences were then manually annotated by locating the temporal intervals of all operations in each sequence. The database will be released to public for research purpose.

After building the database, we then evaluate the recognition performance of the existing approaches on the database. Even though using state-of-the-art approach which demonstrates near perfect performance in recognizing general human actions, we observe significant performance drop when applying it to recognize device operations. A subtle and overlooked unique characteristic of actions involved in using devices restrains the performance of the existing action recognition algorithms.

The uniqueness is illustrated in Figure 2. It shows four actions and the corresponding extracted MoSIFT features [21]. The regions that are relevant to the target actions are indicated by green boxes. Figure 2(a), 2(b) and 2(c) show

[1]http://en.wikipedia.org/wiki/Kinect

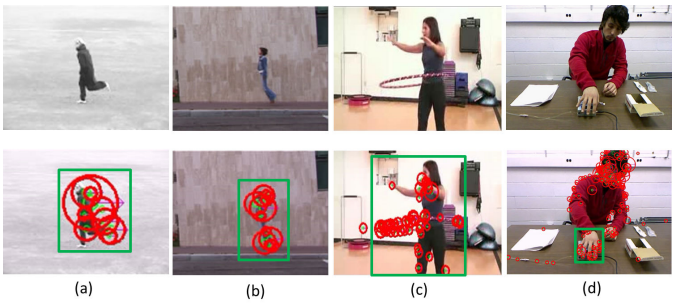

Figure 2: Comparison between four actions and corresponding extracted MoSIFT features [21]. Only features in green box are relevant to actions by definition. (a) is "running" from [7], (b) is "jumping" from [3], (c) is "hula hoop" from [14] and (d) is "turning a device on" recorded by ourself.

the actions of "running", "jumping" and "hula hoop" selected from three popular action recognition datasets [7, 3, 14]. For all three cases, most feature points in the whole frame are inside the green box. Because the "noise" points outside the box are relatively few, it is safe to use all features in the whole frame to model an action. However, as shown in Figure 2(d), for the action "turning a device on", a typical action in using a home medical device, only a very small part of the features lies in the green box compared to all the features extracted from the whole frame. In this case, it is no longer reasonable to use all the features to represent actions, since the representation will be contaminated by substantial amount of essentially random noise. Such differences in feature distributions can be attributed to the fact that the relevant motion of the action in Figure 2(d) is non-dominant and with a relatively small area compared to co-occurring non-relevant motion in the frame. We call this type of actions as *tiny actions*. Most of device operations are tiny actions, because we usually operate a device only with a body part, such as hand or foot, instead of the whole body.

To recognize tiny actions, it's critical to focus on the local area where the target action happens, namely the region of interest(ROI). Therefore, in the second part of this paper, we introduce a simple but effective approach to estimating ROI for recognizing tiny actions. Specifically, the method learns the ROI for an action by analyzing the correlation between the action and the sub-regions of the frame. The estimated ROI is then used as a filter for building more accurate action representations. The experiments show performance improvements over the traditional methods in terms of recognition precision. Note that, the proposed ROI estimation method can be used as a preprocessing step before applying any number of existing methods in the literature of action recognition.

The paper is organized as follows. We first give a review of the related works in Section 2 and then introduce the PUMP database in Section 3. In Section 4, we describe the ROI estimation method for recognizing tiny actions, followed by the experiments on the PUMP database in Section 5. We conclude our work and discuss the future work at Section 6.

2. RELATED WORKS

2.1 Existing action databases

We give a review of current existing action databases and compare them to our proposed one using three taxonomies: (1) RGB videos or RGB+Depth videos, (2) single camera or multiple cameras and (3) significant action or tiny action.

2.1.1 RGB Videos vs. RGB+Depth Videos

The videos of most current databases are RGB videos captured by traditional cameras, such as UCF50 [14], Hollywood2 [8], HMDB [6], KTH [17], Weizmann [3], UT-Interaction [16] and IXMAS [20]. Thanks to the greater availability of RGB+Depth cameras (e.g. Kinect), recently a few 3D databases which contain RGB+Depth videos have been proposed. For example, the MSR3D [19]. Compared to traditional RGB videos, the RGB+Depth videos preserve the additional depth information which could be useful for action analysis. PUMP is a RGB+Depth database.

2.1.2 Single Camera vs. Multiple Cameras

The single camera database refers to those recording actions only use one camera each time, while each action in multi-camera databases was simultaneously recorded by multiple cameras with overlapped views. Again, most of current databases are single camera ones (e.g. UCF50, Hollywood2, HMDB, KTH, Weizmann, UT-Interaction and MSR3D). There are only a few multi-camera action databases have been published, such as IXMAS where each action was captured by 5 cameras from different views. Since we use 3 cameras in PUMP, it is therefore a multi-camera database.

2.1.3 Significant Actions vs. Tiny Actions

We classify actions into two types in terms of their motion strength and relative area compared to whole motion region. The motion of significant actions is strong and dominant one compared to other co-occurred motion in a frame. In contrast, tiny actions' motion is weak, non-dominant and with relative small area. By this definition, the actions in KTH, Weizmann, UT-Interaction, MSR3D and IXMAS are mainly significant actions. For UCF50, Hollywood2 and HMDB, they contain videos with both significant actions and tiny actions. As shown in Figure 4, in PUMP database, all actions are about hands operations, whose movement are weak and only taking a small area compared to co-occurred motion on body, head and etc. Therefore, they are all tiny actions.

2.2 Recognizing human-object interaction

Accurately speaking, the action recognized in cognitive assistive system can be further categorized as human-object interaction which is an important group of actions recognition problems[1]. As [1] indicated, existing methods for recognizing interactions between human and object can generally be classified into two classes by judging if recognition of objects and actions are independently or collaboratively. For methods falling into first categories, objects recognition serves the following action recognition. For example, objects are usually recognized first and then actions are recognized by analyzing the object's motion. As for methods in second class, object and action are recognized in a collaborate fashion and the recognition of objects and actions serve each other.

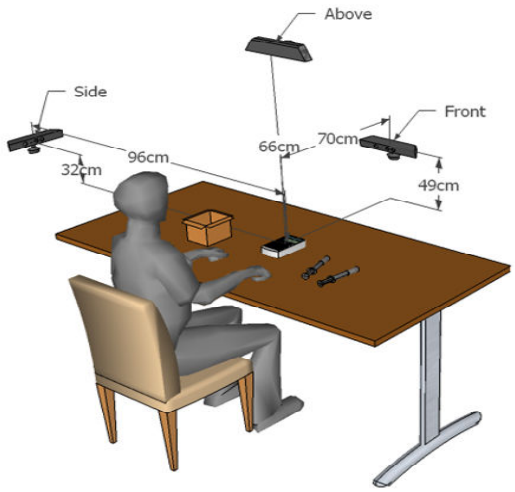

Figure 3: An illustration of data collection setting for PUMP database.

2.3 Detecting salient region

Similar to ROI detction, salient region detection also finds a sub-region in an image or video that is considered to be salient. However, despite the similarity, their difference also worths noting. The saliency of a region is defined by its visual uniqueness, unpredictability, rarity and is caused by variations in image attributes, such as color, gradient, edges, and boundaries [4]. In other words, the saliency detection is not task-dependent by relies on the above rules. However, the ROI detection in cognitive assistive system is task-dependent. For example, the region including device operations may not be salient, due to the weak motion, but is of interest. Due to this difference, existing approaches for salient region detection cannot be directly applied to solve our problem.

3. PUMP DATABASE

3.1 Data Collection Methodologies

We selected the Abbot Laboratories Infusion Pump(AIM Plus Ambulatory Infusion Manager) as an example of home medical devices for research. With the help of a medical devices expert, we first defined an operation protocol for correct use of infusion pump, as shown in Table 1. Then, as illustrated in Figure 3, we set up a "workplace" for data recording. Specifically, we used three Kinect cameras to record user's operation from 3 views: front, side, above. Before each time of recording, an infusion pump with off-state, two refilled syringes and a box of alcohol pads were prepared on the table.

Each user was asked to perform the operation following certain procedures for several times. In each time, they followed either the exact same procedures as described in in Table 1(correct operation protocol) or the predefined wrong procedures(to simulate the use-errors). Table 2 listed 4 types of predefined wrong procedures. They were different from the correct operation protocol by including steps disordering and steps missing. During the data recording, there were

Table 1: The proposed operation protocol for correct use of infusion pump.

1	Turn the pump on.
2-5	Press buttons to set up infusion program.
6	Uncap pump tube end.
7	Clean pump tube end using alcohol pad.
8	Open arm port.
9	Clean arm port using alcohol pad.
10	Flush arm port using syringe.
11	Connect arm port and pump tube end.
12	Press "START" button to start infusion.
13	Press "STOP" button after infusion.
14	Disconnect arm port and pump tube end.
15	Clean pump tube end using alcohol pad.
16	Cap pump tube end.
17	Clean arm port using alcohol pad.
18	Flush arm port using syringe.
19	Clean arm port using alcohol pad.
20	Cap arm port.
21	Turn the pump off.

videos where users unintentionally deviated from the operation protocol they were asked to follow. Since these videos in fact reflected the use error that user made in real-life, we kept them in our database and provided additional error descriptions if the errors they made not belonged to any of the 4 predefined errors (Due to the limited space, we didn't include the descriptions in our paper but kept them as an independent file in the database).

Since some different steps in operation protocol were in fact the same actions, we therefore categorized them into one action class. We further combined the classes with same actions but operating different devices into one class, which finally leaded to 7 action classes. We listed the aggregated classes in Table 3 and gave the snapshots of corresponding actions in Figure 4.

Based on the generalized action categories, we manually annotated all sequences by locating the temporal intervals of all actions in each sequence.

3.2 Database Statistics and Recording Details

There were 17 volunteers participating in the data recording. Each user was asked to operate the infusion pump for 4 times with different appearances. Specifically, 2 times were correct operations while the others 2 came from the wrong operations. We finally constructed a database containing 68 operation sequences where each sequence had three synchronized RGB+Depth videos recorded from different views.

We adopted OpenNI[1] for Kinect recording and stored the raw data in ONI file format. To facilitate the use of this database, we also provided calibrated RGB videos and depth videos extracted from raw OpenNI data in our database. In Table 4, we show detail statistics of the PUMP database.

[1]http://openni.org/

Table 2: The four types of predefined wrong operation protocols. Note that it only listed the differences between the correct protocol and wrong one and non-mentioned parts were same as correct protocol.

W1	Switch step 6 and 7: disorder flushing using syringe with cleaning arm port.
W2	Remove step 4 and 6: forget cleaning arm port and pump tube end.
W3	Remove step 13 and 17: forget cap arm port and pump tube end.
W4	Remove step 9 and 10: forget press buttons to start and stop infusion.

Table 3: The action categories generalized from operation protocol for PUMP database.

1	Turn the pump on/off
2	Press buttons
3	Uncap tube end/arm port
4	Cap tube end/arm port
5	Clean tube end/arm port
6	Flush using syringe
7	Connect/disconnect

Table 4: The inventory of PUMP database. Note that due to the Kinect hardware issue, the frame rate of OpenNI videos was not constant but with little variation.

Video Type	OpenNI	RGB	Depth
View Count	3		
Participant Count	17		
Operation Sequences	68		
File Count	204		
File Format	ONI	AVI	AVI
Frame Rate	≈20	20	20
Resolution	640×480		
Average Duration (m)	4.31		
Total Duration (h)	14.64		
Disk Storage (GB)	157	21	27

The average video duration was 4.31 minutes and the total video duration was 14.64 hours.

4. ROI ESTIMATION FOR RECOGNIZING TINY ACTIONS

4.1 Notations

Let (x, y, z) be the coordinates of the corner index of a cuboid region, A_j be an action of class j, M be the number of action classes. Let $T^{A_j}(x, y, z)$ be a density map that describes the probability for a region at (x, y, z) belonging

to the ROI of A_j. Specifically, we call $T^{A_j}(x,y,z)$ as the *ROI template*.

4.2 Action-Region Correlation Estimation

Noticing the ROI for an action can be interpreted as regions that have strong correlation with the action, we estimate the correlation between an action A_j and each region at (x,y,z) in the "3D" frame recorded by Kinect cameras.

To represent each region, we generate sliding windows starting from the origin of "3D" frame and calculate the bag-of-words(BoW)[18] representation for each cuboid window. To reduce the computation cost in following steps, we then apply the principal component analysis(PCA) on the BoW of each window and only keep the dimensions corresponding to the top K largest eigenvalues. The fisher score[2] is used to estimate the correlations between each region and an action.

Let $D_b^{(x,y,z),A_j}$ and $D_w^{(x,y,z),A_j}$ be action class A_j's *between class distance* and *within class distance*[2] respectively for all regions at (x,y,z) among all training data. Then, $D_b^{(x,y,z),A_j}$ and $D_w^{(x,y,z),A_j}$ are defined as:

$$D_b^{(x,y,z),A_j} = \sum_{k=1}^{M} (\mu_{\delta(k=j)}^{(x,y,z)} - \mu^{(x,y,z)})^T (\mu_{\delta(k=j)}^{(x,y,z)} - \mu^{(x,y,z)})$$

$$D_w^{(x,y,z),A_j} = \sum_{k=1}^{M} \sum_{b \in B_k^{(x,y,z)}} (b - \mu_{\delta(k=j)}^{(x,y,z)})^T (b - \mu_{\delta(k=j)}^{(x,y,z)})$$

where $\delta(z)$ is an indicator function that outputs 1 if z is true and 0 otherwise, $B_k^{(x,y,z)}$ is a set of BoWs of A_k at region (x,y,z), $\mu^{(x,y,z)}$ is the mean of BoWs at region (x,y,z) for all action classes and $\mu_{\delta(k=j)}^{(x,y,z)}$ is the mean of BoWs at region (x,y,z) for action class A_j or action classes other than A_j(depending on if $k=j$). Then the fisher score $F^{(x,y,z),A_j}$ for an action class A_j and a region at (x,y,z) is simply:

$$F^{(x,y,z),A_j} = \frac{D_b^{(x,y,z),A_j}}{D_w^{(x,y,z),A_j}}.$$

If one region (x,y,z) is highly correlated to action A_j, it will then have relatively small *within class distance* $D_w^{(x,y,z),A_j}$ and large *between class distance* $D_b^{(x,y,z),A_j}$, which gives large fisher score $F^{(x,y,z),A_j}$. Therefore, fisher score can be an indicator of correlation between an action and regions.

By normalizing the fisher score at different regions, we get the representation of the ROI template $T^{A_j}(x,y,z)$:

$$T^{A_j}(x,y,z) = \frac{F^{(x,y,z),A_j}}{\sum_{x,y,z} F^{(x,y,z),A_j}}.$$

4.3 ROI Adaption and Noise Filtering

For a given input video sequence, we simply attach the ROI template $T^{A_j}(x,y,z)$ to each frame of the video sequence. Then, all feature points with ROI score lower than threshold λ are removed. In this way, we model the action only based on features in ROI.

5. EXPERIMENTS

5.1 Experimental Setting

The total 68 videos are divided into two-fold and we in turn use one fold as training and the other one as testing data. For each video, we extract the MoSIFT [21] as low-level features and encode them into visual words [18] using a codebook with vocabulary size of 1000. Then three different methods are used for generating the action representations (see Section 5.2 for details). SVM classifier with RBF kernel is adopted for action classification and two-fold cross validation is used for classification model training. Specifically, the training and testing is done independently for each view. To evaluate the effectiveness of different methods, the mean average precision (MAP) [13] which is the average precision (AP) over all actions is computed.

5.2 Action Representation Methods

The bag-of-words model(BoW) is adopted for action representation. Each high dimensional local feature point(e.g. MoSIFT) is first mapped to the closest cluster center using the pre-trained codebook and then the cluster's id is assigned to the feature as "visual word". After that, a pooling step is applied to calculate the statistics of all the visual words in the video segment and represent it as vector with same dimension of the codebook. This vector representation is called the BoW of the video segment. In this paper, we experiment with three different pooling methods for action representation as introduced below.

5.2.1 Whole Frame Based Pooling(WF-BoW)

Most of existing BoW-based action recognition approaches [17, 21] use this pooling method. Namely, all visual words in the whole frame are aggregated together first and the frequency for each visual word is calculated then. Finally, the normalized frequency histogram is used as the final representation.

5.2.2 Depth-Layered Multi-Channel Based Pooling (DLMC-BoW)

Since the videos are recorded by Kinect camera, then each feature point extracted from the key frame has not only the x and y coordinates but also the depth z coordinate. Based on this observation, in [10], z axis is first divided into several depth-layered channels, and then features within different channels are pooled independently, resulting in a multiple depth channel histogram representation. In our implementation, we uniformly divide the depth space into 5 channels.

5.2.3 ROI Based Pooling(ROI-BoW)

In order to estimate the ROI, 200 × 200 × 100 pixels(in the order of x,y and z axis) sliding cuboid windows with moving step of 40 pixels are generated and represented as BoW by aggregating all visual words inside the cuboid window. We then apply the PCA on the BoW of each window and only keep the dimensions corresponding to the top 100 largest eigenvalues. After that, the ROI template is calculated using the method described in Section 4.2. Note that, all these process can be done off-line. At online testing stage, all visual words with ROI score lower than 0.5 are filtered out and the final BoW representation is built only based on the left visual words belonging to the ROI.

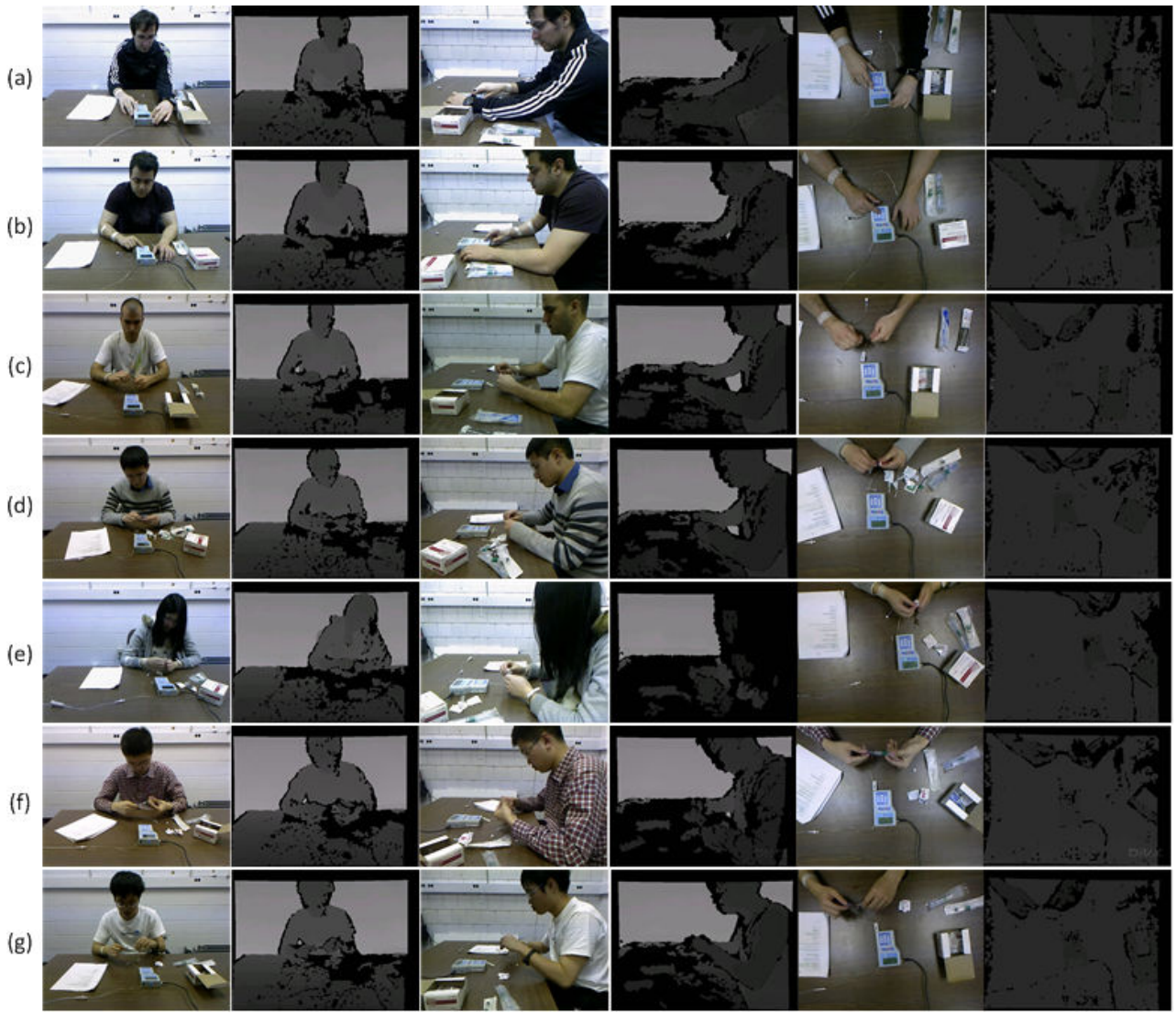

Figure 4: An illustration of 7 action classes generalized from operation protocol of PUMP database. (a)Turn the pump on/off. (b) Press buttons. (c)Uncap tube end/arm port. (d) Cap tube end/arm port. (e) Clean tube end/arm port. (f) Flush using syringe. (g) Connect/disconnect tube end and arm port.

5.3 Experimental Results and Analysis

5.3.1 ROI Visualizations

To qualitatively evaluate the proposed ROI estimation method, in Figure 5, we visualize estimated ROIs for each action of all three views using density map where higher intensity means high probability of belonging to be the ROI. Because the cameras are put at different positions, the ROI for the same action but different views can be different. Comparing the action examples shown in Figure 3 and the corresponding ROIs in Figure 5, we can find the estimated ROIs make sense intuitively. For example, for the "front" view, the ROI is in the middle for action "press buttons" while is on the right for action "Flush using syringe". This is because the "pressing buttons" always happen in the middle of the frame while for "flush using syringe" users always need to take out the syringe from syringe bag which is placed on the right side of the frame. Also, we can observe some of actions' ROI are dense and sharp (e.g. "Connect/disconnect" in side view) while others are relative sparse and soft (e.g. "Cap tube end/arm port" in front view). This difference can be attributed to the different motion patterns of different actions. For example, "Connect/disconnect" concentrates in a narrow area but the motion of other actions like "Cap tube end/arm port" distributes in relatively larger areas.

5.3.2 Action Recognition Performance

In Table 5, we summarize the experimental results of three action representation methods. For each method, the first three columns correspond to the performance on three different views while the last column is the fusion results given by manually selecting the best performance among the three views. The fusion results can be interpreted as the best performance that the cognitive assistive system achieves using that method. Comparing the average fusion results of three

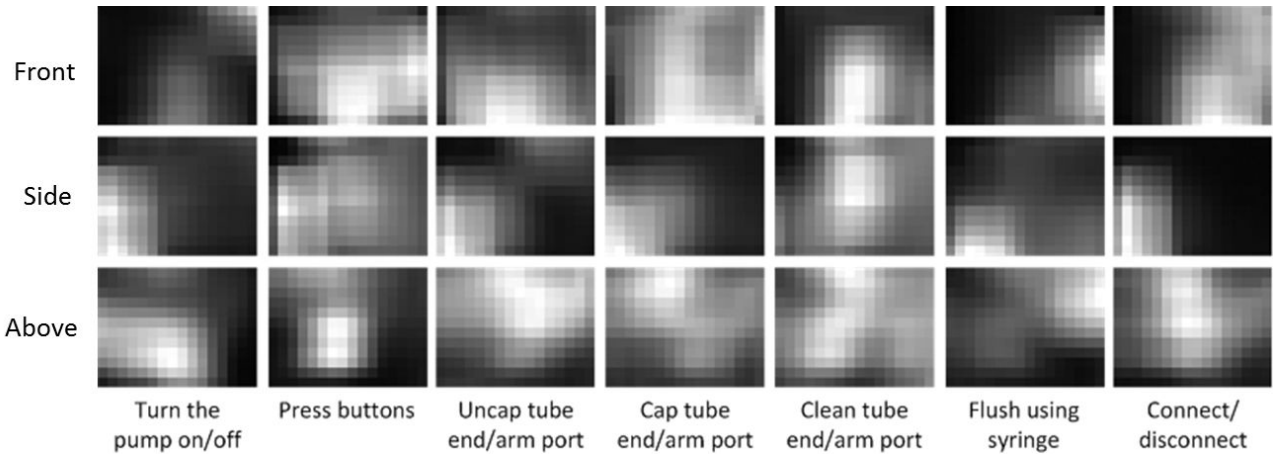

Figure 5: The visualizations of estimated ROI for each action. Each row corresponds to one of the three views. The action examples of different views can be found in Figure 3.

Table 5: MAP comparison of three action representation methods on PUMP database. For each method, the performance is evaluated on each camera independently. The best performance among the three cameras is manually selected and shown in the "fusion" column.

Actions	*WF-BoW*				*DLMC-BoW*				*ROI-BoW*			
	front	side	above	fusion	front	side	above	fusion	front	side	above	fusion
Turn the pump on/off	0.8356	0.5931	0.7222	0.8356	0.863	0.7761	0.8232	0.863	0.894	0.8348	0.8696	**0.894**
Press buttons	0.8044	0.5841	0.7703	0.8044	0.8312	0.7812	0.782	0.8312	0.8833	0.8058	0.8378	**0.8833**
Uncap tube end/arm port	0.4933	0.2679	0.6335	0.6335	0.6252	0.4034	0.5201	0.6252	0.6541	0.5215	0.5326	**0.6541**
Cap tube end/arm port	0.3466	0.2898	0.4356	0.4356	0.3724	0.3303	0.4352	0.4352	0.4455	0.398	0.3923	**0.4455**
Clean tube end/arm port	0.8684	0.6967	0.8676	0.8684	0.8834	0.7331	0.8445	0.8834	0.9202	0.8438	0.8401	**0.9202**
Flush using syringe	0.831	0.6718	0.8329	0.8329	0.8845	0.7589	0.8038	0.8845	0.948	0.9048	0.7931	**0.948**
Connect/disconnect	0.451	0.3361	0.6349	**0.6349**	0.48	0.3704	0.5302	0.5302	0.5037	0.4801	0.5335	0.5335
Average	0.6615	0.4914	0.6996	0.7208	0.7057	0.5933	0.677	0.7218	0.7498	0.6841	0.6856	**0.7541**

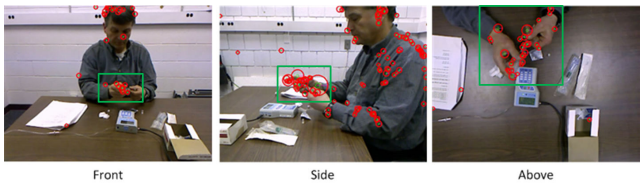

Figure 6: The visualization of extracted MoSIFT features in frames with the same time stamp but of different views.

action representation methods, we can see that the *ROI-BoW* achieves the best performance and improves the MAP for 3.33% compared with *WF-BoW*. *DLMC-BoW* has almost the same performance as *WF-BoW*.

If we further compare three methods' performance on each single view, we observe a different performance changing pattern. For the "front" and "side" views, both *ROI-BoW* and *DLMC-BoW* show significant improvements over the *WF-BoW*. Specifically, on average, *DLMC-BoW* improves for 4.42% and 10.20% while *ROI-BoW* improves for 8.84% and 19.28% on those two views. However, for the "above" view, both *ROI-BoW* and *DLMC-BoW* don't show improvement but in fact slightly decrease the performance compared to *WF-BoW*. The reason causing the inconsistent performance changing pattern on different views is illustrated in Figure 6. It visualizes of extracted MoSIFT features in example frames with the same time stamp but different views for action "Flush using syringe". Again, we use green boxes to indicate the regions that are relevant to the target action. We can see that for "front" and "side" views, only a very small part of the features lies in the green box compared to all the features extracted from the whole frame, while most features are inside the green box for the "above" view. Therefore, due to the difference in camera positions, actions recorded by "front" and "side" cameras are always the most typical tiny actions. Because both *DLMC-BoW* and *ROI-BoW* can be interpreted as feature location based visual word weighting methods (*DLMC-BoW* does it implicitly by using SVM to weight features at different depth differently while *ROI-BoW* does it explicitly by hard weighting features inside ROI 1 and outside 0), they are most effective when actions are typical tiny actions.

However, even though *ROI-BoW* improves the performance significantly, if looking into the absolute performance of each action, we realize only actions "Turn the pump on/off", "Press buttons", "Clean tube end/arm port" and "Flush using syringe" achieve reasonable high average precision while the performance of left three actions is still low.

To build an applicable cognitive assistive system, we have to accurately recognize all the actions. Therefore, it still requires special effort to further boost the performance of the difficult actions.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a cognitive assistive system to monitor the use of home medical device. To build such a system, accurately recognition of user actions is one of the most essential problems. However, since few research has been done in this specific direction, it is still unknown if current techniques are adequate to solve the problem. In order to facilitate the research in this area, we made three contributions in this paper. First of all, we constructed a database where users were asked to simulate the use of infusion pump following predefined procedures. The operations were recorded by three Kinect cameras from different views. All the data was manually labeled for experimental purpose. Secondly, we performed a formal evaluation of some existing approaches on the proposed database. Because we realized current methods can hardly deal with tiny actions involved in using home medical devices, we made our third contribution by introducing an ROI estimation method and applying the ROI for building more accurate action representations. The experiments show significant performance improvements over the traditional methods by using the proposed methods.

Currently, we treat all the actions in operating home medical devices as independent ones and recognize them separatively. However, it obvious they are mutually related because they are operations in a procedure. Therefore, in the future, we will focus on leveraging the inner relations between actions to further improve the recognition performance.

7. ACKNOWLEDGEMENTS

This material is based in part upon work supported by the National Science Foundation under Grant IIS-1251187. Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not reflect the views of the National Science Foundation.

8. REFERENCES

[1] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Computing Surveys(CSUR)*, 2011.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1997.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *International Conference on Computer Vision (ICCV'05)*, 2005.

[4] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. Global contrast based salient region detection. In *IEEE CVPR*, 2011.

[5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, 2005.

[6] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *International Conference on Computer Vision (ICCV'11)*, 2011.

[7] I. Laptev. On space-time interest points. *International Journal of Computer Vision (IJCV)*, 2005.

[8] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *Computer Vision and Pattern Recognition (CVPR'09)*, 2009.

[9] B. Meier. F.d.a. steps up oversight of infusion pumps. *New York Times*, 2010.

[10] B. Ni, G. Wang, and P. Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011.

[11] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision (IJCV)*, 2008.

[12] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *International Journal of Computer Vision (IJCV)*, 2006.

[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition*, 2007.

[14] K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications Journal*, 2012.

[15] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *The International Conference on Computer Vision (ICCV'09)*, 2009.

[16] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA), 2010.

[17] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *International Conference on Pattern Recognition (ICPR'04)*, 2004.

[18] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *The International Conference on Computer Vision (ICCV'03)*, 2003.

[19] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR'12)*, 2012.

[20] D. Weinland, M. Özuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. In *European Conference on Computer Vision (ECCV'10)*, 2010.

[21] M. yu Chen and A. Hauptmann. Mosift: Reocgnizing human actions in surveillance videos. In *CMU-CS-09-161*, 2009.