Proceedings of The National Conference
On Undergraduate Research (NCUR) 2015
Eastern Washington University, Cheney, WA
April 16-18, 2015

Automatic Recognition of Postoperative Shoulder Surgery Physical Therapy Exercises from Depth Camera Images

Sean MacAvaney
Electrical Engineering and Computer Science
Milwaukee School of Engineering
1025 North Broadway
Milwaukee, WI 53202

Faculty Advisor: Dr. Jay Urbain

Abstract

Shoulder rotator cuff surgery is one of the most common orthopedic surgeries performed today, particularly in adults over the age of 65. To restore range-of-motion after this surgery, physical therapy exercises are important, but are often not completed in full due to long recovery times, overly-optimistic patient expectations, and the cost of clinical appointments. We present an application that uses a depth camera (Microsoft Kinect) to aid in shoulder surgery physical therapy exercises by recognizing, measuring, and providing immediate feedback about exercises performed by a patient. This paper evaluates state-of-the-art machine learning algorithms for gesture and motion recognition from skeletal data for use in the aforementioned application. The application and several machine learning algorithms are evaluated using data from clinical trials of pre- and post-operative shoulder rotator cuff patients. Results demonstrate the efficacy of the approach.

Keywords: Machine Learning, Physical Therapy, Depth Imaging

1. Introduction

Injuries to the shoulder's rotator cuff are common, especially among adults over the age of 65[(1)]. These injuries impair the arm's range of motion, cause pain, and ultimately inhibit an adult's ability to live on his or her own. While surgery can help restore motion and reduce pain in the long-term, patient expectations of recovery times are typically too optimistic. As a result, the average number of physical therapy visits is only roughly nine over a period of four weeks, while exercises should be performed more frequently and for a longer duration[(2)].

It has been shown that depth images from a Microsoft Kinect camera can be used to measure a patient's range of motion with statistically insignificant variance with clinician measurements[(3)]. This allows an application to be developed to measure and track a patient's progress over time that is capable of in-home and in-clinic use. This application should both give the patient more realistic expectations of their treatment duration, and increase the frequency of physical therapy exercises by making them more convenient and enjoyable. One challenge in developing such an application is the identification of which exercise the patient performs in real-time. The classification process is the focus of this research.

2. Methodology

In order to produce an effective classification model, one must first transform raw data into a format more suitable for processing by using domain knowledge. For this application, the raw data come in the form a depth camera image – essentially a matrix of distance measurements. Although it would be possible to build a gesture recognition model based on these data, it would require specific training on a variety of body types. Instead, an existing model was used

for transforming the depth image into major skeletal points. The skeletal data is then filtered to account for bad frames and normalized to account for differences in body types and orientation, in attempt to achieve person-invariant normalization. Finally the data is fed into a learning algorithm. A high-level overview of this process is shown in Figure 1.

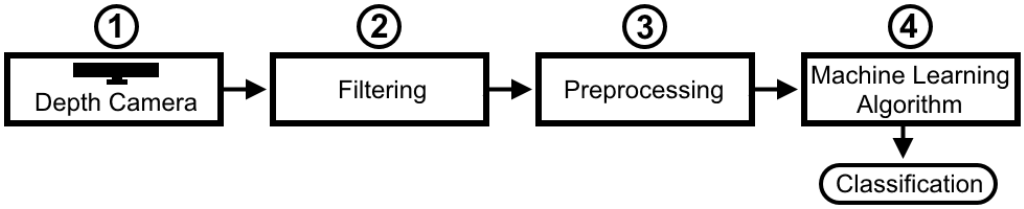

Figure 1: Overview of machine learning process. Each step in the process has an effect on the final classification.

2.1. Data Collection

The Microsoft Kinect depth camera is the data source for all additional steps. The camera provides real-time depth imaging over USB from an array of infrared emitters and detectors[4]. This image can be viewed as a grayscale image in which lighter pixels are points closer to the camera, and darker pixels are farther away. It is often also viewed as a point cloud in which each point is based on the depth image's pixel. The x- and y-coordinates are based on the pixel's position in the image, and the z-coordinate is the depth value (going "though" the screen, simulated by rendering points that are farther away as being smaller). This visualization approach exposes more detail to the viewer than a simple depth image.

The depth image by itself would not work well for this application because there is a lot of irreverent data: everything the background, the folds in clothing, etc. None of these have an effect on which movement is being performed. The natural interface middleware (OpenNI) resolves much of this by identifying people and providing a basic skeletal structure. OpenNI requires a "submissive" pose (Figure 2) before tracking people for calibration, and to verify the user's intention to be tracked. The basic skeletal structure provided by OpenNI consists of 15 points referred to as joints: left and right shoulder, elbow, hand, hip, knee, and foot, as well as the head, neck, and torso. Each point is a 3-diemental coordinate.

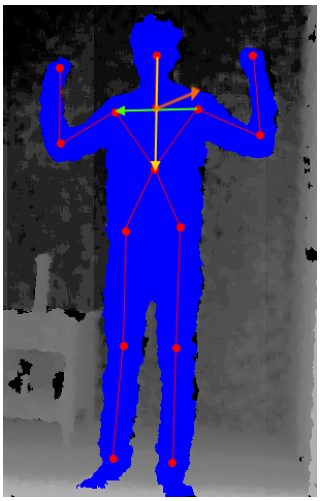

Figure 2: OpenNI distinguishes the subject from the surroundings. The 15 joints are highlighted as red circles. The x, y, and z components of the orientation plane are shown as green, yellow, and orange arrows, respectively.

2.2. Filtering

Joint positions provided by OpenNI are not always perfect. In particular, some frames become severely disfigured due to noise in the depth image. To work around this, joint data undergoes filtering to reduce the effect of such frames on

the future processing. The three techniques evaluated were basic mean filtering, exponential filtering, and median filtering. A comparison of these filtering techniques with an extreme value is shown in Figure 3.

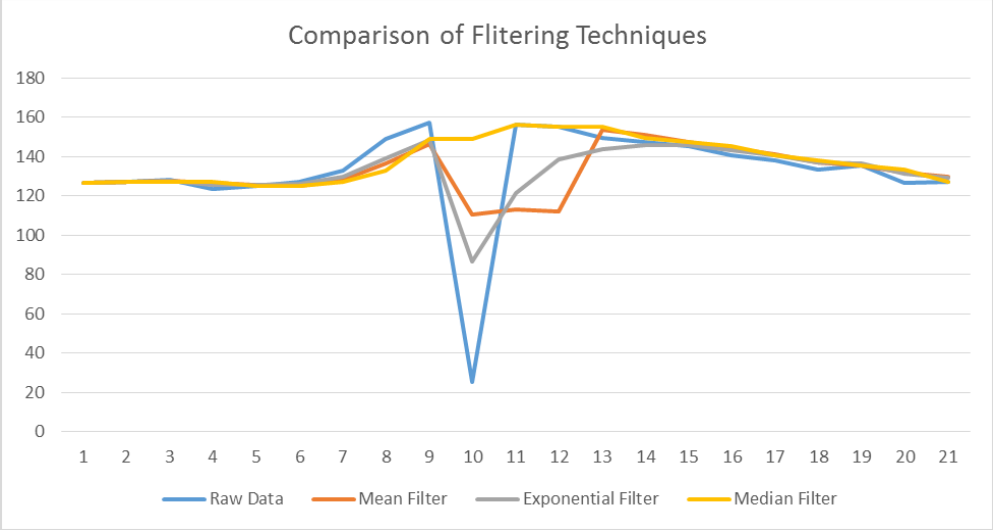

Figure 3: Comparison of filtering techniques. The median filter is least effected by the sudden drop in value at 10.

2.3. Normalization & Parameterization

Despite having the skeletal data extracted from the depth image, the data is not yet ready for analysis because a great deal of irreverent information is still present: the height of the subject, how far the subject is standing from the camera, the subject's orientation to the camera, etc. The normalization step resolves these issues. First, a standard skeletal orientation is applied because the subject often does not face the camera directly. This is accomplished by forming a new coordinate space based on the direction the subject is facing. The vector between the right and left shoulders form the x axis, the vector between the head and torso form the y axis, and the cross product of these vectors forms the z axis. The result is all skeletal data being comparable in terms of direction.

To eliminate the effects of distances encoded in the data, three angles are then calculated around nonterminal joints (e.g. shoulder, elbow). Each angle is simply over each axis. This approach is valid because the subject's orientation is fixed (as the subject rotates his or her body, the angle will remain static). In effect, this outputs a skeleton in which neither the subject's body proportions nor the subject's heading have an effect on the output data – only the relative orientation of each point.

2.4. Markov Chaining

Analyzing each frame in isolation may result in worse-than-desired accuracy in the system. A Markov chain combines previous values in the evaluation of future values. In this way, motion over time could be analyzed by the machine learning algorithm, rather than instantaneous position. An easy way to incorporate this is to simply add past feature sets into present analysis.

2.5. Machine Learning Algorithms

A variety of supervised learning algorithms were tested. Supervised learning algorithms make use of labeled training data to build a model which is capable of predicting the classification of new test data. To test the effectiveness of each algorithm, an additional set of labeled data was fed into the model, and the classification provided by the model was compared with the label assigned manually. The results of this process are displayed in a confusion matrix. A confusion matrix shows counts of classification values and the correct values. This can be used to determine which movements are typically labeled incorrectly. The following machine learning algorithms were evaluated: naïve Bayes, logistic regression, decision trees, and random forests.

An algorithm's bias is the degree to which it makes assumptions about the data. Some algorithms like Naïve Bayes have a high bias, while others like decision trees have low bias. Generally, high bias algorithms will produce a more general model, while low bias algorithms produce a very specialized model that is often susceptible to over-fitting. To produce the most effective model possible, a fusion algorithm is used to combine the results of each machine learning algorithm to produce better results in most situations.

2.6. Subject Motions and Demographics

Four motions were evaluated for both the left and right arms: abduction, forward elevation, external rotation with arm adducted, and external rotation with arm abducted. Abduction is simply the raising of the fully-extended arm to the side over the coronal plane. Forward elevation is the abduction of the fully-extended arm anteriorly along a parasagittal plane. External rotation with the arm abducted is the rotation of the arm with the elbow at about 90° and the upper arm normal to the sagittal plane. External rotation with the arm adducted is a similar motion, but instead with the upper arm alongside the body, normal to the transverse plane.

Five subjects were recoded performing each exercise for both their left and right arms. Four subjects recently had shoulder surgery, and one was healthy and did not have impaired range of motion.

3. Results

Each preprocessing and machine learning algorithm was evaluated in isolation using both self-recorded data and data from clinical trials. Samples were taken with a variety of body types, speed of movements, and body orientation. The accuracy of each trial is defined as the sum of the true positives and true negatives divided by the total number of samples.

3.1. Filtering Evaluation

Median filtering was the most accurate filtering technique. See Table 1. Median Filtering provided roughly a 3% to 4% increase in accuracy when data was then normalized and evaluated against a decision tree. Mean filtering typically provided no significant improvement (and often provided worse results). Exponential filtering provided consistently good results, at around 2% to 3% improvement.

Table 1: Accuracy of various filtering techniques. Median filtering of the past 10 values was most accurate.

Filter	Parameter	Accuracy (%)
None	n/a	87.03
Median	n=4	91.14
Median	n=6	91.44
Median	n=8	91.75
Median	**n=10**	**92.35**
Mean	n=2	86.35
Mean	n=4	86.49
Mean	n=6	86.62
Mean	n=8	86.69
Mean	n=10	87.20
Exponential	α=0.9	88.31
Exponential	α=0.7	90.33
Exponential	α=0.5	89.89
Exponential	α=0.3	90.26
Exponential	α=0.1	87.57

3.2. Normalization & Parameterization Evaluation

The normalization process described in §2.3 proved to be very effective. It greatly improved the accuracy and left and right side consistency. See Table 2. To evaluate this data properly, the model was trained on an "ideal" training set, and evaluated against the data set collected from clinical trials. The somewhat high accuracy for the right side may be attributed to poor anatomical assumptions made due to low sample sizes.

Table 2: Accuracy of various normalization techniques. The normalized angles technique was most accurate.

Normalization Technique	**Accuracy (%)**
No Normalization	3.63
Normalized Skeletal Positions	10.62
Normalized Angles	**76.39**

3.3. Markov Chaining

Markov chaining seemed to effectively improve the accuracy of the model (Figure 4). In particular, including values from 5 samples prior resulted in the best improved accuracy.

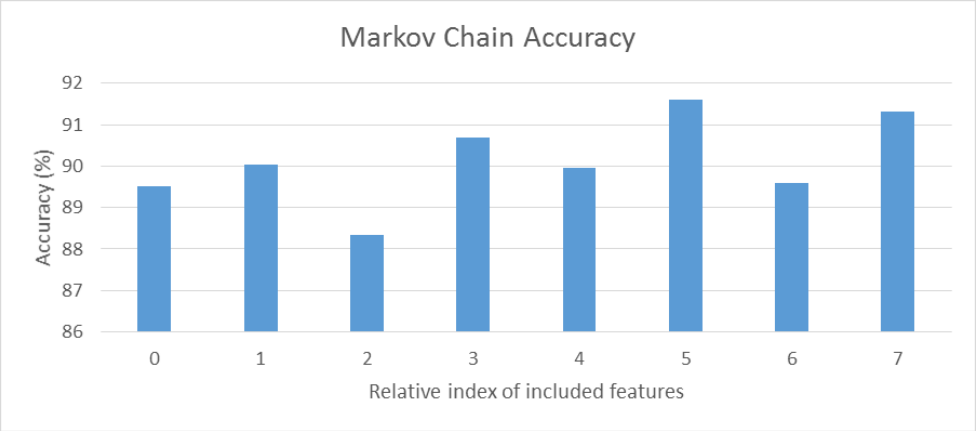

Figure 4: Comparison of Markov chain parameter (relative index of included feature).

3.4. Machine Learning Algorithm Evaluation

Four representative machine learning algorithms were evaluated: Naïve Bayes, Logistic Regression, Decision Tree, and Random Forest (Table 3). The decision tree outperformed the other algorithms. It was particularly surprising that it had a higher accuracy than the random forest, which would theoretically produce a more generalized model.

Analysis of the confusion matrixes (Table 4) indicate that the algorithms had the most difficult time recognizing the difference between abduction and forward elevation. These two movements are very similar (both involve the abduction of the arm), but differ in the direction in which the arm is being raised. Furthermore, at maximum elevation the hand often exits the depth image frame, causing errors in the skeletal tracking by OpenNI.

Table 3: Machine learning algorithm results. The decision tree outperformed other methods.

Algorithm	Accuracy (%)
Naïve Bayes	74.56
Logistic Regression	81.80
Decision Tree	**89.52**
Random Forest	87.94

Table 4. Confusion matrixes for various machine learning algorithms. There is typically confusion between abduction and forward elevation.

Naïve Bayes Confusion Matrix

a	b	c	d	e	f	g	h	<-- classified as
264	31	11	22	40	11	0	1	a = L Abduction
133	**178**	9	7	40	17	1	0	b = L Forward Elevation
10	0	**316**	0	1	2	0	14	c = L Arm Elevated Adduction
5	0	1	**315**	0	19	0	0	d = L Arm Elevated Abduction
5	2	0	1	**227**	87	20	4	e = R Abduction
18	4	4	1	145	**260**	19	1	f = R Forward Elevation
8	0	23	0	4	20	**361**	0	g = R Arm Elevated Adduction
1	0	0	8	0	5	0	**292**	h = R Arm Elevated Abduction

Logistic Regression Confusion Matrix

a	b	c	d	e	f	g	h	<-- classified as
285	59	10	4	15	5	1	1	a = L Abduction
118	**224**	7	2	18	10	4	2	b = L Forward Elevation
2	0	**318**	1	3	1	2	16	c = L Arm Elevated Adduction
2	0	3	**316**	0	16	0	3	d = L Arm Elevated Abduction
11	14	5	1	**246**	42	24	3	e = R Abduction
17	7	1	6	31	**371**	15	4	f = R Forward Elevation
4	2	18	0	0	13	**379**	0	g = R Arm Elevated Adduction
1	1	0	8	4	1	2	**289**	h = R Arm Elevated Abduction

Decision Tree Confusion Matrix

a	b	c	d	e	f	g	h	<-- classified as
328	13	4	3	13	17	1	1	a = L Abduction
105	**261**	0	0	11	7	1	0	b = L Forward Elevation
1	1	**335**	0	1	1	3	1	c = L Arm Elevated Adduction
0	0	3	**326**	0	6	0	5	d = L Arm Elevated Abduction
10	6	0	1	**310**	14	4	1	e = R Abduction
11	5	2	6	17	**406**	5	0	f = R Forward Elevation
2	0	11	0	3	5	**395**	0	g = R Arm Elevated Adduction
0	0	4	3	1	1	1	**296**	h = R Arm Elevated Abduction

Random Forest Confusion Matrix

a	b	c	d	e	f	g	h	<-- classified as
277	81	0	2	8	9	2	1	a = L Abduction
107	**266**	3	0	6	3	0	0	b = L Forward Elevation
2	1	**332**	0	0	0	7	1	c = L Arm Elevated Adduction
0	0	2	**329**	0	5	0	4	d = L Arm Elevated Abduction
8	6	0	1	**310**	17	3	1	e = R Abduction
12	8	1	5	20	**405**	1	0	f = R Forward Elevation
2	0	8	0	4	2	**396**	4	g = R Arm Elevated Adduction
0	0	2	5	0	0	4	**295**	h = R Arm Elevated Abduction

4. Conclusion

Overall, the results show that these techniques can be used to effectively build a model for classification of post-operative shoulder surgery physical therapy movements from depth images. Median filtering is an effective technique for removing bad frames. Skeletal normalization into component joint angles allows the model to be generalized for a variety of body types and orientations. Decision trees provide a simple, yet robust, algorithm for generating a prediction model.

5. Future Work

A variety of techniques could be used to further improve the classification algorithm. One area for significant improvement would be in Markov chaining. The approach analyzed was to simply include the features from a previous sample in the analysis of the current sample. A more effective technique would be to collect statistics over an entire set of frames over a period of time. Dynamic Time Warping (DTW) could be used to either produce a feature for the frame (e.g. "elevation" score), or could be used to dynamically define the length of the period itself.

Other work has shown that DTW can provide valuable metrics about the similarity of a movement sample to an ideal. This could be used to build a custom classification model that attempts to fit the data as best as possible to the ideal, varying the starting point. One big concern with this approach would be computational complexity.

Analysis of parameterizations could also be beneficial. As more features are added with Markov chaining and other metrics, it becomes more important to only focus on the most important factors. An in-depth analysis of this could result in greatly improved performance. This could also involve analysis of higher-order models, where new features are fabricated from the existing set of features in attempt to improve the prediction power of the model.

6. References

1. Yamamoto, A., Takagishi, K., Osawa, T., Yanagawa, T., Nakajima, D., Shitara, H., & Kobayashi, T. "Prevalence and risk factors of a rotator cuff tear in the general population," Journal of Shoulder and Elbow Surgery, 19(1), (2010) 116-120.
2. Milgrom, C., Schaffler, M., Gilbert, S., & Van Holsbeeck, M. "Rotator-cuff changes in asymptomatic adults. The effect of age, hand dominance and gender," Journal of Bone & Joint Surgery, British Volume, 77(2), (1995) 296-298.
3. Jay Urbain *Investigation of Natural Interfaces for Evaluation and Management of Shoulder Impairment* (2012)
4. Greg Borenstein, *Making things see: 3D vision with Kinect, Processing, and Arduino* (Farnham: O'Reilly, 2012)