Sit-to-Stand Movement Recognition using Kinect

Erik Acorn[1], Nikos Dipsis[1], Tamar Pincus[2] and Kostas Stathis[1]

[1]Department of Computer Science and [2]Department of Psychology
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK
Erik.Acorn.2013@live.rhul.ac.uk

Abstract. This paper examines the application of machine-learning techniques to human movement data in order to recognise and compare movements made by different people. Data from an experimental set-up using a sit-to-stand movement are first collected using the Microsoft Kinect input sensor, then normalized and subsequently compared using the assigned labels for correct and incorrect movements. We show that attributes can be extracted from the time series produced by the Kinect sensor using a dynamic time-warping technique. The extracted attributes are then fed to a random forest algorithm, to recognise anomalous behaviour in time series of joint measurements over the whole movement. For comparison, the k-Nearest Neighbours algorithm is also used on the same attributes with good results. Both methods' results are compared using Multi-Dimensional Scaling for clustering visualisation.

Keywords: machine-learning · movement recognition · Kinect · sit-to-stand

1 Introduction

The use of computers to collect, analyse and interpret human movement is becoming increasingly popular. In particular, we are witnessing the development of natural interfaces that allow users to interact with computers while performing ordinary movements. Key to such interfaces is the detection, categorisation and analysis of body movements made by people with different body shapes and physical characteristics.

Because of their intuitive and ordinary application, natural user interfaces can only be expected to increase in prevalence, especially with the introduction of cheaper hardware providing input sensing and supporting enhanced processing techniques. A case in support of this position is the introduction of motion sensing devices such as the Microsoft Kinect [1], which are conveniently used by a wide audience in their home environment.

Although there may be a multitude of uses for which it is necessary to process motion capture data for further analysis, we are motivated by applications such as that proposed by Bragaglia et al [2], who discuss an architecture based on Kinect to recognise home exercises by people, including fall recognition for elderly patients. We are particularly interested in applications that would support people who are suffering from back pain. Low back pain has recently been estimated as the number one ranking cause of daily disability worldwide [3] and there is a great incentive for patients to be able to carry out rehabilitation in their homes.

However, there are expected complications with the use of input devices such as Kinect, in particular those arising from noisy data and variability between examples of a similar action. In a real world example, noise can be due to background movements and inaccurate interpretation of body parts by the capturing software. Variability can be due to differing distances from the capturing device, different body shapes, determination of the start of a motion and duration of the motion.

To address these issues we aim to create an experiment for collecting human movement data with emphasis on compromised movement due to back pain. Our objective is to apply machine-learning to analyse the data and to create enough human movement data from different subjects to be able to effectively apply an algorithm for comparison and to identify anomalies. To make the discussion concrete, we study the movement of standing up from a seated position with a view to categorising the same motion performed by various participants.

One benefit of bringing this kind of movement analysis into the home is that it will allow people with medical problems to be monitored and treated remotely, saving hospital care time and resources. It also has the potential of allowing physiotherapy exercises from home with expert feedback. Such feedback would either be asynchronous feedback from a physiotherapist examining the data after collection or directly from the machine-learning algorithm whereby the anomaly can be identified by means of a trained computer model.

The work provides a framework whereby movement data based on sit-to-stand movements are extracted from a dataset based on joint behaviour. The main contribution underlying subsequent data analysis is the production of the warp distance matrix as a new application of Dynamic Time Warping to human motion data. The matrix produced eliminates the time dimension in the data. This new approach is shown to produce good results when used as input for all of the machine- learning techniques applied. We also show how a bagging implementation of k-Nearest Neighbours (kNN) can average the results over a large number of out-of-bag validations, at the same time producing a kNN distance matrix allowing the use of Multi-Dimensional Scaling for cluster visualisation.

The paper is structured as follows. Section 2 discusses the background work, other work in related fields, the history of movement recognition, different purposes for its use and methods of analysis employed. Section 3 describes the framework of our approach including a description of the set-up, general approach to the boundaries of the work, data structures set up for use in evaluation code, visualization tools and explanations of data analysis algorithms used. Section 4 is the main evaluation, presenting the intermediate processes applied to the data, results of the various techniques used including graphical representations to demonstrate outcomes of the analytical procedures, explanations of results and values of error rates attained. Finally section 5 provides discussion and conclusions of the attained results including a summary of the achievements of the project as well as an outline of further work.

2 Movement Recognition: analysis & models

Motion capture and movement recognition using computational techniques starting with Marker Based Systems (MBS) was described in early work of Cappozzo et al [4]. MBS set-ups such as Vicon use sophisticated stereophotogrammetrics and actors with carefully placed markers on key body joints or anatomical landmarks. This is necessarily carried out in a specialised environment with expensive equipment and requires time consuming preparation. The system cost alone is US$96-120k according to Han et al [5]. Recent developments have seen low cost Markerless Motion Capture (MMC) devices becoming available including Sony Playstation Eye, Prime Sense Sensor, Intel's Creative Camera and Microsoft Kinect. With these depth cameras and Natural User Interface libraries such as Kinect SDK, OpenNI/NITE, Evoluce SDK and others (Shingade & Ghotkar [6]), motion recognition can be carried out in a non-specialised environment.

Fig. 1. Kinect sensor

The Microsoft Kinect sensor is an RGB-D camera costing about US$200. It records colour video and uses an Infra-red emitter, projecting a structured IR light onto the scene in order to calculate depth measurements. The sensor was launched in 2010 with a description of use by Shotton et al [7]. Microsoft's Natural User Interface for Kinect web page [1] contains documents describing skeletal tracking using Kinect. The cost effectiveness of setting up MMC systems has led to increased use in many areas such as motion capture for games and films (Sinthanayothin et al [8]), NUI using hand gestures (Elgendi et al [9]), and chronic pain rehabilitation using 'serious games' (Schonauer et al [10]). A few studies have compared the two methods (MBS and MMC) for relative accuracy. Studies by Bonnechere et al [11] on static body segment recognition concluded that Kinect is very reliable if measurements are calibrated. Linear regression was used to equate measurements from the Vicon MBS with Kinect data. The MBS data is taken as a gold standard. The same group performed another study [12] to compare motion analysis, considering body angles as well as body segment lengths during a deep squat action. This time the conclusion was that the angles at large joints i.e. shoulder, hip and knee were reliable after regression analysis in comparison with MBS. Limb end segments and angles were found not to be reliable; in particular hands feet, forearm and elbow.

Machine-learning techniques were researched for this problem because the data is multi-dimensional. Recognising patterns over the whole data context including over time is extremely complicated and adding to that variation in body shapes and other factors augments the complexity. Machine-learning techniques can be used to reduce the dimensionality and the variability of the data. Data sample alignment is necessary in order to compare like portions of the movement data. Okada & Hasegawa [13] described Data Time Warping (DTW) for this purpose, minimizing Euclidean distance between datasets by warping in the time domain.

Dynamic Time Warping allows two time series that are similar but locally out of phase to align in a non-linear manner. According to Ratanamahatana & Keogh [14], DTW is the best solution known for time series problems in a variety of domains. Their work gives a good insight into bounding constraints and distance metrics when configuring DTW. Han et al [5] use the result of the DTW calculation as a measure of similarity between data-sets. Action recognition is performed on Kinect data in order to compare results with a Vicon (MBS) set-up. The results of this calculation would be applicable to a k-Nearest Neighbours (kNN) algorithm for example.

Herzog et al [15] describe a variation of DTW tailored to account for their Parametrised Hidden Markov Model (PHMM) which considers the hidden state. PHMM is discussed in the following subsection describing Models. Chandola et al [16] suggest two ways to tackle anomaly detection in time series. One is to reduce the contextual anomaly detection problem to a point anomaly detection problem. This may be done using rigid or elastic alignment of time series for example DTW. The second is to model the structure in the data and use the model to detect anomalies. Modelling the structure effectively means reducing dimensions and Han et al [5] use Kernel Principal Component Analysis for this purpose.

Brandao et al [17] presented a comparative analysis of three algorithms applied to human pose recognition using RGB-D images from a Kinect sensor with Kinect SDK. The static pose is represented by coordinates in a bounding box with sides of between 8 and 64 units in different trials. The algorithms used for comparison were C4.5 Gain Ratio Decision Tree, Naive Bayes Classifier, and k-Nearest Neighbour Classifier and were implemented in the data mining tool Weka. The conclusion was that the best classifier was found to be kNN although the Decision Tree Classifier was almost as good; the larger the number of cells used to describe the body part positions, the better the prediction although 8x8 was almost as good as 64x64; and the pose predictions were greater than 90% even when using participants with different body shapes for the data capture.

Herzog et al [15] detail the implementation of Parametric HMM models for recognition of movement using Gaussian output distributions and considering transition probabilities between states which follow each other in the action; the other transition probabilities being zero. The probabilities are calculated using Baum/Welch expectation maximization (EM) for a given training set. As mentioned in the previous section, they used a modified DTW for data alignment to account for the hidden state, first setting up a global HMM model using the whole training set and secondly training local HMMs aligning with reference to the global HMM. This uses the idea that HMMs are temporally invariant.

Random forests are successfully used for human pose recognition by many authors (Shotton et al [18], Rogez et al [19]). Classification of a depth image from a static pose is predicted with random forests grown using a large training dataset.

3 Experimental Set-up and Framework

Data collection experiments were performed in a controlled environment in order to minimize noise and external factors. Window blinds were closed to maintain constant lighting. A plastic based swivel chair without castors was placed at a fixed distance from the sensor which was mounted on a computer desk 1m above floor level and 2.5m from the chair. The motion capture was performed with each participant three times facing the Kinect sensor and three times at 45^o to the line of the sensor. It was decided that imaging with the subject facing at 90^o side-on to the sensor was unreliable because Kinect software tries to interpret body parts which are occluded with this aspect.

Thirty-one random volunteers participated in the collection of the experimental data, 21 men and 10 women of heights between 1.52m to 1.94m. The volunteers were shown a short video describing how to stand up from a seated position. They were then each asked to stand in front of the Kinect sensor, sit down and stand up while a motion capture was performed. During this process, each participant was filmed separately using a video camera in order for later reviewing by a physiotherapist. In this way each participant produced six data samples. Thus 93 example data sets were produced for each of the two aspects. On assessment these examples were confirmed as being a smooth action with balanced weight positioning and without twisting and thus were labelled as normal.

In addition to the normal participants, three additional participants performed purposefully faulty actions of four different categories according to the guidelines, producing 12 extra data sets facing the sensor and 12 angled 45^o to the sensor. The faulty actions were: (i) starting with feet too far forward; (ii) standing with head too far back; (iii) twisting one shoulder while standing and (iv) starting with trunk too far back (>45^o at hip joint).

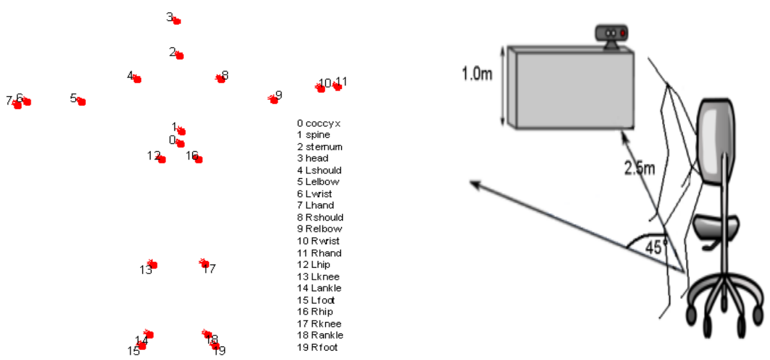

Fig. 2. (a) Kinect anatomical landmarks (b) Experiment facing 45^o to line of Kinect

The data produced by the Kinect sensor is given as 20 body points with x, y and z coordinates relative to an origin, at a frequency of 30Hz. Thus every second 20 x 3 x

30 data points are produced. A data reader was used to store the raw data representing human body movement. This collecting software written in C++ buffers the data stream to RAM in real time and periodically a separate thread writes the data to disk. The figure shows the 20 body points measured by the Kinect API software.

3.1 Data Visualization

We have developed a *Reanimator Tool* to visualize a full joint data set over the duration of a given example. Any data example can be replayed as a 3-d animation (see **Fig. 3**). Reviewing the animated data is helpful in determining the start of the action and any anomalies or corruptions in the captured data up to the start of the action.

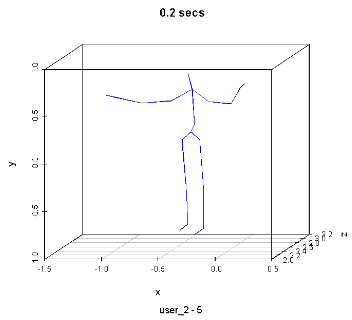

Fig. 3. A data frame taken from the Reanimator Tool

Any joint can be plotted against its warped plot and the reference plot using a function written for this visualisation. The reverse warp from the reference plot to the example is also plotted for comparison. This demonstrates the effect of dynamic time warping with respect to a reference data example on a given example, for a given joint in axis x, y or z. The figure below shows an example of warp plots.

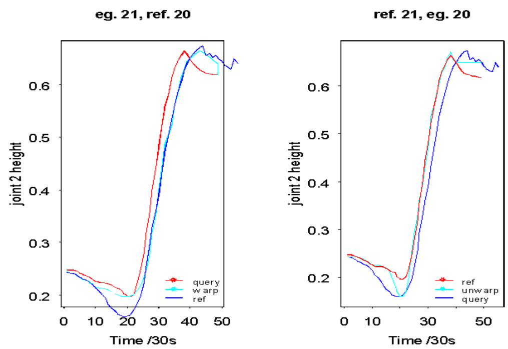

Fig. 4. warp plot function output

Any joint can be plotted in a given axis consecutively for each participant. This gives an idea of the form of the joint's movement over time for various examples and can be applied to the excised data from the start or to the whole data sample. Any abnormal examples can be spotted. **Fig. 5** shows a joint plot of 4 successive participants.

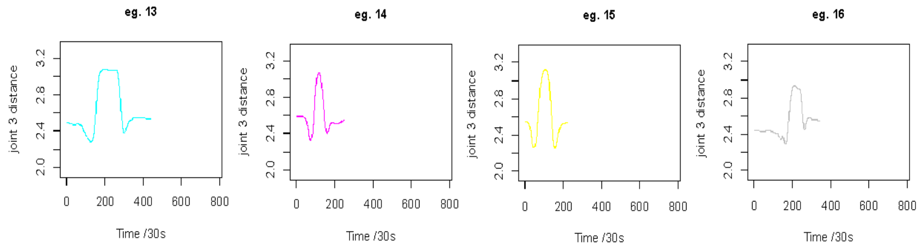

Fig. 5. Head joint plots with time for different participants

3.2 Analysis

It is necessary to determine the start and end points of the action by applying recognisable conditions in the data. Similarly it will be useful to determine a finite number of intermediate key points during conditional recognisable actions in the data to enable alignment. Using DTW allows continuous realignment of data over time measurement intervals as well as providing a way of calculating an averaged distance metric with respect to a reference dataset. This distance is useful as a metric in any classification technique. The principal of DTW will be discussed here and its application to the data is discussed in the Evaluation section.

To compare two similar vectors of time series data of lengths *m* and *n*, dynamic time warping algorithm creates an *mxn* matrix of distances between all elements of each vector. The lowest cost contiguous path from the starting point to the end point is the chosen path whereby indices of the two vectors are mapped effectively dilating and compressing the time line to produce the best fit. Various configurations can be chosen for the distance metric and matrix path.

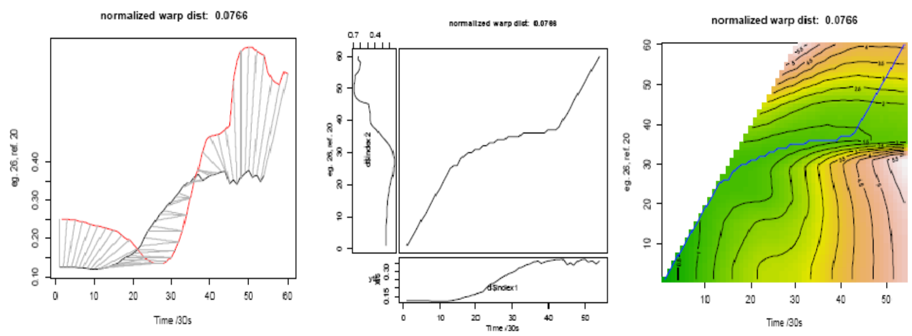

Fig. 6. Three views of a warp between sternum plots of an example and a reference example

Plotting the warp is a good way of determining the configurations producing the most appropriate warp. The 3 plots in **Fig. 6** show different representations of the warp of example 26 against example 20. The chosen step configuration is asymmetric which is why the density plot is missing the upper left corner.

Having carried out pre-processing on the raw data, further work is necessary to fit a model to categorise movement types. Broadly, there are two approaches for categorising the motion; the first being clustering based on likeness or otherwise to other labelled data and the second being to create models representing proper and improper movements. Both methods would involve the analysis of a sequence of events, each event having a context i.e. time series with a number of independent parameters such as body segment angles. The difficulty is to identify anomalies in a set of contexts and for this reason it is necessary to reduce or eliminate the time dimension.

The less complicated approach is to use transduction to categorise by association though it may not supply intuitive reasons for improper movements, which limits the usefulness of a diagnosis. Clustering algorithms use similarity or difference measurements between data points to classify points into groups. There are many different possibilities for the way of measuring similarity. Euclidean distance is a commonly used metric. kNN is the transductive method chosen based on the background reading and its implementation is described in section 4.6.

Using induction to create a model is more complicated but may give more information about the reasons for an improper movement. A method used in some studies is Hidden Markov Model for which transition matrices could be determined for proper and improper movements transitioning between the key intermediate stages of the action. However this requires the identification of those key 'hidden states'. Janssen et al [20] discuss four phases of the sit-to-stand motion identified by Shenkman et al [21] but note the major influence in characteristics of a sit-to-stand action in a range of studies where chair heights and types varied.

Decision trees have the advantage of giving an intuitive explanation of the reason for a prediction model, the most important attribute being at the top of the tree. However, decision tree models may vary widely with different data sets so random forests are appropriate particularly with small data sets. Random forests are formed by applying decision trees on bootstrapped examples chosen from the data pool. Averaging the results leads to lower variance and this method known as bagging maximises the use of the available data for producing a model and validating it. Random forests can be configured in various ways, using randomly chosen subsets of attributes up to the full complement and can be averaged over any number of sample selections without causing over-fitting.

The data sets available in this project are limited and any prediction method should be tested using data examples and comparing real outcomes with predictions. This may be achieved using cross-validation or bagging methods. The advantage of bagging is that the model can be averaged over a large number of random samples thus making the most of labelled data and also reducing variance.

4 Evaluation

4.1 Extracting Action Data

The Data collected includes in most cases standing in front of the sensor while waiting for Kinect to recognize the body joints, then assuming the seated position before carrying out the action of standing up. There may then be trailing actions which, along with any data recorded before the action, should be trimmed from the data. Functions were written in order to programmatically retrieve the required data. Doing this programmatically means that some rules need to be decided which may be fairly complicated in order to be applied correctly to all data sets. However, it is useful to be able to prepare the data automatically so that it is somewhat uniform. The two data measurement techniques used, namely with subject face-on to sensor and 45^{o} angled, are considered completely separately because the 3-d data measurements produced are unrelated. The idea is to produce two sets of results for comparison.

In order to remove any preliminary movement from the data sets, the variation of values of the distance of the head from the sensor is used. In normal circumstances, the seated position places the head at its furthest distance from the sensor during the sensor-facing experiments and furthest to the right of the sensor during the angled experiments. Before searching for the start point, a minimum point is verified in the majority of cases where the participant started from a standing position. In the 3 cases where the participant started seated, this minimum value is ignored.

Similarly to the determination of starting, the end point is determined programmatically but using the height of the sternum which is expected to increase following the start of the action and reach a steady value at the end of the action. When height values stay within a small distance of each other for more than 10 readings (1/3 of a second) this is taken as the end of the movement. Typical plots of the sternum height are shown below before and after data excision.

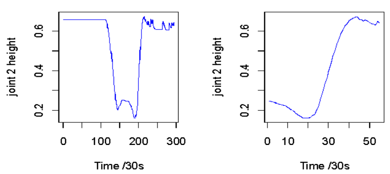

Fig. 7. A sternum plot before and after start and finish indexing

4.2 Warp Distance Matrix

Using one of the data sets as a reference, time warping was obtained based on the height of the sternum joint during the action. The reference example used was example 20 which corresponds to User2-2 and User2-5 in the face-on and 45^{o} aspects and is compatible with all other examples for the purpose of warping. The sternum is chosen as the reference joint for the warp because it is central and most representative of the time sequence of the whole body movement.

Having calculated the warp indices for each example data set, the sternum-based warp is used to map the time series data from the remaining joints with their corresponding joint from the reference example. By this method, difference measurements are calculated between the mapped measurements and summed over the full period. The normalized distance for each joint and axis is calculated by averaging with respect to time elapsed. Thus a 3-dimensional matrix of distances is created with a single value for each joint in each of 3 axes. These values are used as attributes in building decision trees and the other predictive algorithms

4.3 Classification

For the purpose of this study, all actions performed by normal participants were initially labelled as not faulty. The examples 1-12 corresponding to the 3 participants who performed faulty standing actions are labelled as FALSE, and examples 13-105 are labelled as TRUE. Although some of the normal participants were assessed with minor faults, the training algorithms will use the more exaggerated faults in the creation of models with the hope of assessing the other examples.

4.4 Decision Trees

The decision trees built with all of the data from each of the two data sets give only one classification error for the face-on data tree and none for the other tree. These trees show the important attributes used to match the trees to the data and distinguish actions labelled faulty. The values however are only relevant to the data set produced against the reference example and have no meaning as absolute values. The closer the value to 0, the more alike that attribute is to that of the reference example after normalization. The figure below shows the trees built from the two sets of data with attribute values at decision points described by axis and joint location. These trees model the full data sets but are not validated and it can be assumed that that the data is over-fitted.

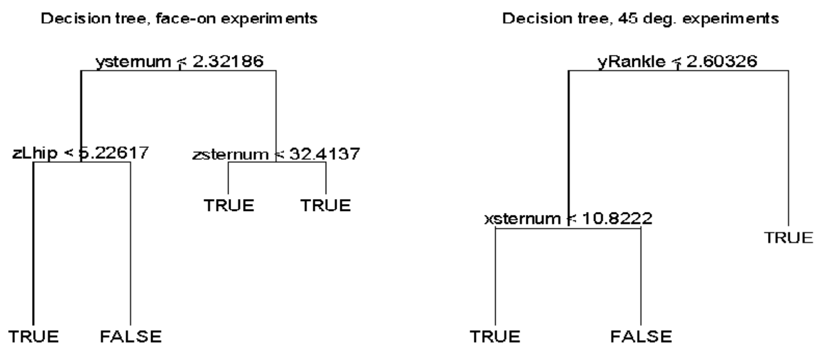

Fig. 8. Decision tree plots showing dimension and joint of decision splits

Rather than using k-fold cross-validation and averaging error estimation, the random forest algorithm is a better method for prediction and evaluation using decision trees. Random forest results follow this sub-section.

4.5 Random Forests

Random forests were grown with the parameters: maximum number of nodes, number of attributes for decision trees, and number of trees to grow. Best results were obtained with trees of size 3. All attributes from the frames were used which includes 10 joints in x and y axes for face-on data and all 3 axes for 45^{o} data.

The confusion matrices show out-of-bag prediction error rates of 6.7% and 4.8% respectively for the two data sets, almost all of the errors coming from the FALSE classifications. In this case, error rates for false positive classification (classifying a FALSE result as TRUE) were 58% and 33% respectively. These results may vary using different seeds as the process is stochastic. In fact the 45^{o} data set consistently showed better classification of faulty actions. Given the high rate of correct classification of non-faulty actions, greater than 50% correct classification of faulty actions results in the low overall out-of-bag prediction error rates stated (see tables below).

Table 1. Face-on random forest results

	FALSE	**TRUE**	**error**
FALSE	5	7	0.58333
TRUE	0	92	0

45^{o} random forest results

	FALSE	**TRUE**	**error**
FALSE	8	4	0.33333
TRUE	1	91	0.01087

Using proximity matrices produced by the random forest algorithm which is a square matrix of distances between every pair of examples, a multi-dimensional scaling plot can be produced which graphically demonstrates distances of examples where most are clustered closely. This shows a trend in the faulty examples because the random forest was able to distinguish them successfully; but also some examples of the non-faulty trials which lie outside of the main cluster. The 45^{o} data shows good clustering of a group of faulty trials some distance from the main cluster of non-faulty examples.

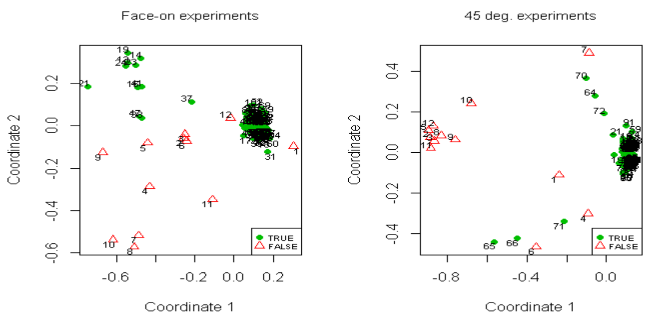

Fig. 9. Labelled Multi-Dimensional Scaling plots of Random Forest distances

In the face-on plot in **Fig. 9** the 11 examples in the top left are outside of the normal cluster, identified in the plot by their data example numbers. In the 45^{o} plot the 3 examples in the bottom centre are outside of the normal cluster. These represent examples labelled as normal which show some abnormalities according to the algorithm.

4.6 K-Nearest Neighbours with bagging

The k-Nearest Neighbours algorithm was used on the scaled attribute data with a bagged sample over 500 sample selections and the out-of-bag predictions averaged to produce a confusion matrix. The implementation stores predicted results for out-of-bag samples on each bag selection. After 500 selections each example is awarded a classification based on which classification was predicted more often for that example. The following error rates are produced with k=2 giving 1.9% and 0 out-of-bag prediction error rates respectively.

Table 2. Face-on kNN results**,** k=2

	FALSE	**TRUE**	**error**
FALSE	11	1	0.083333
TRUE	1	91	0.010870

45^{o} kNN results, k=2

	FALSE	**TRUE**	**error**
FALSE	12	0	0
TRUE	0	92	0

The actions of a single user may be expected to be close to one another so using a higher value of k may give a more realistic error rate. Using k=5 implies a majority of 3 neighbours in the training set determine the result of the test set. Table 3 shows the results with k=5, giving 4.8% and 0.96% error rates.

Table 3. Face-on kNN results, k=5

	FALSE	**TRUE**	**error**
FALSE	7	5	0.41667
TRUE	0	92	0

45^{o} kNN results, k=5

	FALSE	**TRUE**	**error**
FALSE	11	1	0.083333
TRUE	0	92	0

kNN uses a distance matrix between every example set of attributes and every other to calculate the nearest neighbours. This can be plotted using multi-dimensional scaling

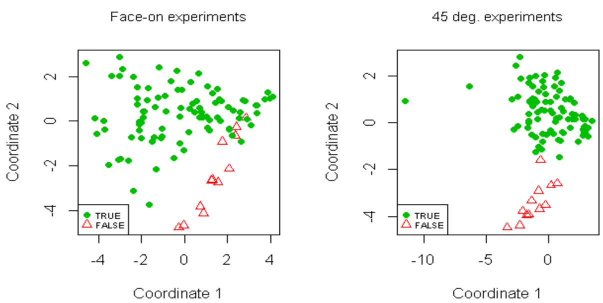

Fig. 10. Labelled Multi-Dimensional Scaling plot of kNN distances

Dynamic time warp distances are used as the attributes and each example has a TRUE or FALSE label. The plot shown in **Fig. 10** shows why the results for kNN bagging prediction error are so low, particularly for the 45^o experiments. The clusters for faulty and non-faulty actions are well defined.

5 Discussion and Conclusions

We have presented an experiment for collecting human movement data using the Kinect depth sensor motion camera with emphasis on variation in movement due to back pain. Various machine-learning techniques were used to analyse the data and effectively applied to identify anomalies. A general method of comparing data sets acquired from experiments was devised. Results show that attribute sets created using a technique based on dynamic time warping can be successfully used in machine-learning algorithms to identify anomalous actions. Results using random forests and using k-Nearest Neighbours with bagging showed good clustering and good out-of-bag validation results with overall error rates below 7%. Best results were obtained on the 45^o experiments which when analysed using kNN with k=5 give an overall error rate of <1% and false positive rate of only 8% meaning 92% of the anomalous actions were identified. Classification predictions could be made using the models created against new examples of the same action. Multi-dimensional scaling techniques were used to visualize distance matrices produced by the two methods.

Further work could use machine-learning methods to detect the action within a full data set. This is an important part of the overall process. It is important to build a larger database of experimental data including clinical participants with back pain. In such a case the heuristic approach we used to identify the offset of the action within a data set becomes less feasible. While matching a part of the overall action using machine-learning, the same process based on dynamic time warping could be used to identify anomalies. In this case, the dynamic time warp would have to be free to match the start rather than anchored to the start of the data. Using a non-anchored configuration with dynamic time warping was not found to be satisfactory so a new approach to the DTW algorithm could be considered. Other models such as Hidden Markov Models could also be considered for this approach.

References

1. "Natural User Interface for Kinect" Microsoft http://msdn.microsoft.com/en-us/library/hh855352.aspx
2. S. Bragaglia, S. Di Monte, P. Mello "A Distributed System Using MS Kinect and Event Calculus for Adaptive Physiotherapist Rehabilitation" Eighth International Conference on Complex, Intelligent and Software Intensive Systems 2014
3. F, Balagué et al. Non-specific low back pain. The Lancet , Volume 379 , Issue 9814 , 482 – 491, 2012.
4. A. Cappozzo, F. Catani, A. Leardini, U. Croce "Position and orientation in space of bones during movement: anatomical frame definition and determination" Clinical Biomechanics 1995

5. S. Han, M. Achar, S. Lee, F. Pena-Mora "Empirical assessment of RGB-D sensor on motion capture and action recognition for construction worker monitoring" Visualization in Engineering June 2014
6. A. Shingade, A. Ghotkar "Animation of 3D Human Model Using Markerless Motion Capture Applied to Sports" International Journal of Computer Graphics & Animation Jan. 2014
7. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake "Real-time human pose recognition in parts from single depth images" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011
8. C. Sinthanayothin, N. Wongwaen, W. Bholsithi "Skeleton Tracking using Kinect Sensor & Displaying in 3D Virtual Scene" International Journal of Advancements in Computing Technology (IJACT) June 2013
9. M. Elgendi, F. Picon, N. Magnenat-Thalmann "Real-Time Speed Detection of Hand Gesture using Kinect" Institute for Media Innovation paper
10. C. Schonauer, T. Pintaric, H. Kaufmann "Chronic Pain Rehabilitation with a Serious Game using Multimodal Input" Proceedings of International Conference on Virtual Rehabilitation June 2011
11. B. Bonnechere, B. Jansen, P. Silva, H. Bouzahouene, V. Sholukha, J. Cornelius, M. Rooze, S. Van Sint Jan " Determination of the precision and accuracy of morphological measurements using the Kinect sensor: comparison with standard stereophotogrammetry" Ergonomics Apr. 2014
12. B. Bonnechere, B. Jansen, P. Silva, H. Bouzahouene, L. Omelina, J. Cornelius, M. Rooze, S. Van Sint Jan "Can the Kinect sensors be used for Motion Analysis?" Transaction on Electrical and Electronic Circuits and Systems Jan. 2013
13. S. Okada & O. Hasegawa "Motion recognition based on dynamic time warping method with self-organizing incremental neural network" Proceeding of 19th International Conference on Pattern Recognition (ICPR) 2008
14. C. Ratanamahatana, E. Keogh "Everything you know about Dynamic Time Warping is Wrong" University of California paper
15. D. Herzog, V. Kruger, D. Grest "Parametric Hidden Markov Models for Recognition and Synthesis of Movements" Aalborg University Copenhagen paper
16. V. Chandola, A. Banerjee, v. Kumar "Anomaly Detection: a Survey" ACM Computing Surveys Sep. 2009
17. A. Brandao, L. Fernandes, E. Clua "A comparative Analysis of Classification Algorithms Applied to M5AIE-Extracted Human Poses" Proceedings of SBGames Oct. 2013
18. J. Shotton, R.Girshick, A.Firzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, A. Blake "Efficient human pose estimation from single depth images" Advances in Computer Vision and Pattern Recognition 2013
19. G. Rogez, J.Rihan, S. Ramalingam, C. Orrite, P.H. Tor "Randomized trees for human pose detection" Computer Vision and Pattern Recognition IEEE Computer Society Conference 2008
20. W. Janssen, H. Bussmann, H. Stam "Determinants of the Sit-To-Stand Movement: A review" Physical Therapy Journal 2002
21. M. Shenkman, R. Berger, P. Riley et al "Whole-body movements during rising to standing from sitting" Physical Therapy Journal 1990