# Walking Cycle Analysis

Team members:
Yucheng Wang <yuchengw@andrew.cmu.edu>
Ligen Peng <ligenp@andrew.cmu.edu>

18-798 Image, Video and Multimedia
Instructor: Yang Cai

## I. Problem Statement and Background

Human motion analysis is constantly receiving increasing attention from researchers from different fields such as computer vision, human-computer interaction, artificial intelligence, medical care and social security. This interest is motivated by a large quantity of potential applications such as remote automatic monitoring, surveillance, disabled auxiliary system, man-machine interface and content-based image storage and retrieval [1].

There are several problems that researchers are trying to solve at present: 1) How to extract human features efficiently using the equipment such as web cameras, Kinect, etc. 2) How to determine what those important features [2] are we need to care about? 3) How to combine all these features together to realize some automatic application aided by computer?

If all these problems have been solved with reasonable time complexity and space complexity, then we can efficiently develop a lot of useful applications to help us in real world while saving a large amount of time and human resources. For example, we can analyze the intent of those people with disabilities and then robots can be utilized to help these people. We can also realize remote diagnosing by analyzing the gait features and some other motion features. Also, this can be used in the field of surveillance [3] by doing the automatic tracking of thefts and robbers when no other witnesses are on the scene. In general, the achievement of our project would be greatly helpful in future intelligent life.

Figure 1: Robot Aided System

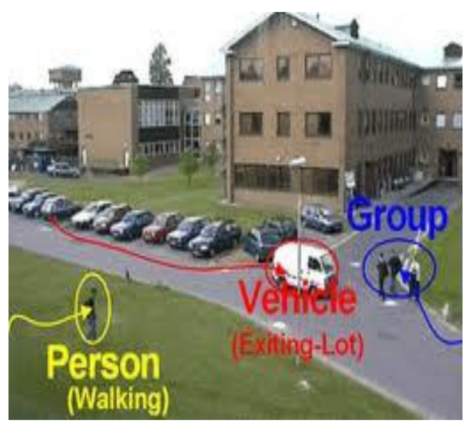



Figure 2: Intelligent Surveillance


Figure 3: Intelligent Vehicle

Figure 4: Remote Diagnostic

In our project, we are making effort to extract human motion features from the depth stream and the skeleton stream from Kinect. And then we define several features for us to do the classification. By making a great amount of experiments, we are adjusting our parameters to realize accurate classification. Additionally, we built our own mathematical model for confidence calculation.

## II. ALGORITHM SURVEY

There are several human motion analysis algorithms and the algorithm is different because they are based on the different mathematical models.

Conventionally, human bodies are represented as volumetric models, 2D contours, or stick figures [4]. Because of these different human graphical models, the body segments can be approximated as lines, 2D ribbons, and elliptical cylinders, accordingly, Figures, 5, 6, 7 show examples of the stick figure, 2D contour, and volumetric representations of the human body, respectively.

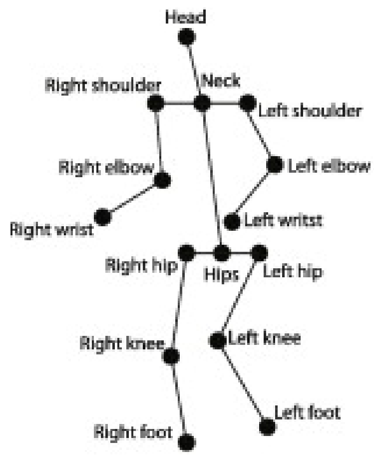



Figure 5: A stick-figure human model

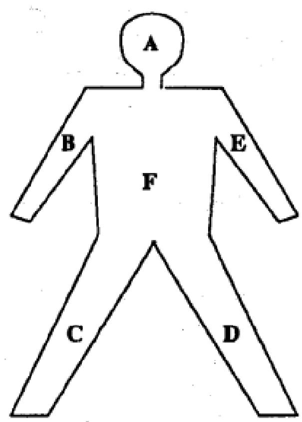



Figure 6: A 2D contour human model(similar to Leung and Yang's model [5]).

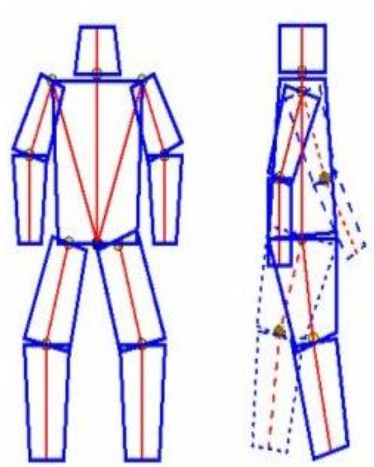


Figure 7: A volumetric human model

In traditionl human motion or walking cycle analysis. All methodologies follow the general framework of: 1) feature extraction, 2) feature correspondence, and 3) high-level processing. Most approaches for 2d or 3D interpretation of human body structure focus on motion estimation of the joints of body sementes between consecutive frames. When noa priori shape models are assumed, heuristic assumptions are usually imposed to establish the correspondence of joints in an image sequence. There assumptions define the constraints on feature corrrespondence, decrease the search space, and, eventually, result in a unique match.

The simplest representation of a human body is the stick figure, which constists of line sements linked by joints the motion of joints provides the key to motion estimation and recognition of the whole figure, think marked joints as moving light displays(MLD) . and using axis difference to define applicable algorithm and our project is focus on this model because of the Kinect model using for describing human body is stick figure.

Another way to describe the human body is by using 2D contours. Under such descriptions, the human body segments are analogous to 2D ribbons or blobs. For example, Shio and Sklansky [6] focused their work on 2D translational motion of human blobs. The blobs were grouped based on the magnitude and direction of the pixel velocity, which was obtaied using techniques similar to the optical flow method(OFM). The velocity of each part was considered to converge to a global average value over several frames This average velocity corresponded to the motion of the whole human body This observation led to identification of the whole subject via region grouping of blobs with a similar smoothed.

There are also several aglorithms that are not based on the human body model. They are based on successfully tracking a human human image from image sequences, some researchers compute the optical flow fields between consecutive frames and divide each flow frame into a spatial grid in both X and Y directions. The motion magnitude in each cell is summed, forming a high dimensional feature vector using for recognition. To normalize the duration of the movement, they assume that human motion is periodic and divide the entire sequence into a number of cycles of the activity. Motion in a single cycle is averaged throughout the number of cycles and differentiated into a fixed number of termporal divisoins. Finally, activity recognition is processed using the nearest neighbor algorithm.

Moreover, Hidden Markov Model, which is a probabilistic technique for the study of discrete times series, has been adopted for recognition of human motion sequences in computer version. Its model structure could be summarized as a hidden Markov chain and a finite set of ouput probablility dstributions [7]. The main tool in HMM people motion recgontion problem is the Baum-Welch(forward-backward) algorithm for maximum likelihood estimation of the model parameters. Features to be recognized in each state vary from points and lines to 2D blobs, obtained from any homogeneous regioins based on motion, color, texture, etc The work by Yamato et al. [8] is perhaps the first one on recognition of human action in this category. Mesh features of binary moving human blobs are used as the low-level feature for learning and recognition. Learning was implemented by traning HMMs to generate symbol patterns for each clas. Optimization of the modle parameters is achieved using the Baum-Welch algorithm, as metioned above. Finally, recognition is based on the ouput of the given image sequence using forward calcualtion. They tested sequences with six tennis strokes and achieved recognition rates ranging from 70% to 100%, depending on the number of training patterns. There are several advanced work related to HMM, most of them are add or modify the features in feature vector, the base aglorithm is similar and most them has already extened to human acitivity recognition.

## III. ALGORITHM DESCRIPTION

As for our application algorithm, we designed our algorithm based on relative difference on 2D axis because most features are extracted from stick figure obtained from Kinect API. Figure 8 has showed a base model of our algorithm.

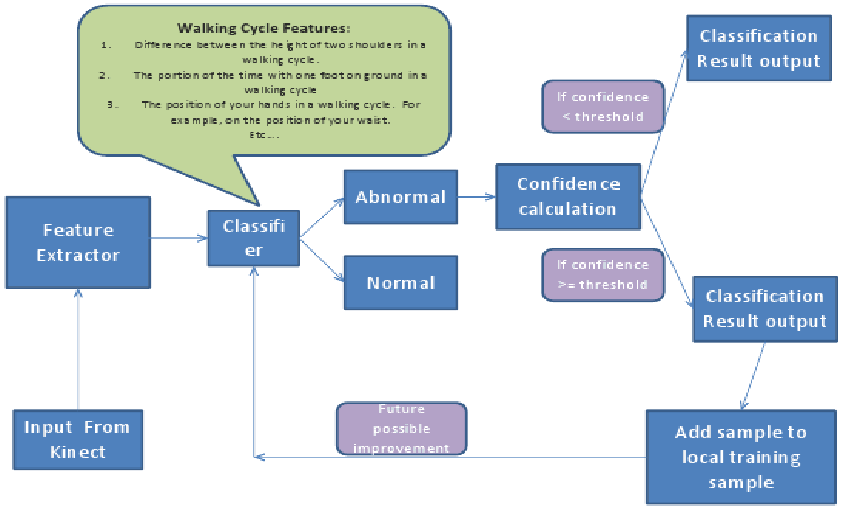



Figure 8: algorithm base model

For the input, we use the Kinect API, collect the image frames in certain time and Kinect provide two model for frames rate per second: 30 frames/sec and 45 frames/sec which could be set in its xml configuration file. Since our application is not data decided, we use low frame rate as data resource, it also could give us higher real time performance than higher frame rate.

Because our current algorithm is focused on six problems: humpback, neck issues, lame, people walking with cane, people walking with crutches, and pregnant femme. So we divide the whole recognition problem into sub problems descripted above, and for each sub problem, we define features we need to extract from stick figure model, and then test features for each sub problems. For example, for simplest people with neck problem, we use the x-axis difference in certain confidence time to identify if the person has neck problem, as figure 9 shows.

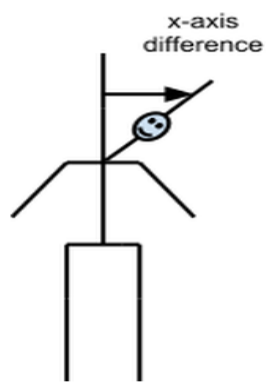



Figure 9: neck issue feature extraction using x-axis difference

After obtaining the x-axis difference in each separate frame, we first add them up in ceratin time interval, then divide the summation by frame number, finally we use normal people x-axis difference as threashold (obatined from experiments from different people) to identify if the people has neck problem, and we consecutively use confidence modual to give it a confidence. A general sub problem algorithm described as below:

First obtain particular feature values from feature vector
If the value is large than certain problem threshold, use confidence model to give it a confidence value

If confidence is low, test other features
Else show the confidence of particular problem.

Else test other problem.

As general sub problem shows, we manually arrange the sub problems in particular sequence, the sequence is tuned in the experiment to ensure that we could identify the significant problem of the people, and in the meantime we could decrease the co-occurrence of those sub problems.

In higher level, we structure our application software architecture in three main module Feature extraction module, Classification module and Confidence test module as Figure 8 shows. Feature extraction module is responsible for extracting features from each sub problems. Table 1 shows the selected feature for each six sub problems.

| Problems: | Feature extracted |
|---|---|
| Neck issues | x-axis diff between head and spine |
| Humpback | Depth diff between head and pelvis |
| People with right cane | Humpback feature plus left right hand y-axis diff and x-axis diff between hand and elbow |
| lame | Accumulated value about left and right y-axis diff. if left foot's height is larger than right foot's height in a range, plus 1, vice versa |
| People with crutches | Lame feature plus x-axis diff between hand and elbow and y-axis diff between hand and foot |
| Pregnant femme | Depth diff between head and pelvis |

Table 1: feature vector for classification

Classification module is used for classifying each sub problems described above, we use priority to decide the sequence of problem identification, first we examine if people has head problem, then whether people is humpback, lame, people with canes or crutches, finally we test pregnant femme, there are two main reasons for this sequence, first, we want to give a severity rank for each sub problems, like people with lame often has neck issues according to the human morphology, secondly, there are dependencies between different sub problems, for example, most people with canes are likely has humpback, so we test if people is humpback first, then go further to test humpback problem. Figure 10 shows the high level algorithm.

Confidence test module is used for calculate the confidence of each sub problem, and show the value on interface that exposed to users. In this module, we define several principle for calculation which will be shown in mathematical articulation section, the base idea we use is to set up a normal value for each sub problem, most are extreme value of each sub problem's features, we use the experimented value obtained from real time to calculate correspondent confidence value, for multiple features, we use normalized value from different values as the final values for this problem. If the confidence value is higher enough (higher than 60% in our application), we saved the corresponding images with particular problem in database respectively for further research work.

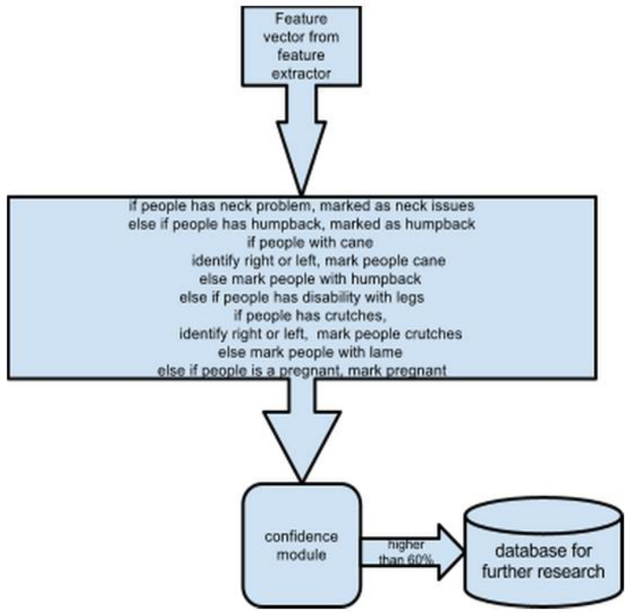



Figure 10: detailed algorithm in high level

The algorithm has limitations: The body segment angle and joint displacement were measured and smoothed from real image sequences, and then a common kinematic pattern was detected for each walking cycle, which was done by Kinect platform and application embedded into it. The drawback to this model is that it is view-based and sensitive to changes of the perspective angle at which the images are captures. So our algorithm is try to diminish this drawback but with little improvement.

## IV. Rapid Prototype

In our project, we apply both depth image stream and skeleton image stream from Kinect to extract features of walking cycle, position of hand, relative depth distance between head and hip center and so on. Then by utilizing all these features, we generate our own human body model, mathematical model for classification confidence, feature extractor and classifiers. We then use our own classifiers to classify people into the following categories: humpback person, lame person, pregnant woman, elderly walking with cane on the right, elderly walking with cane on the left, people walking crutches and normal person. In addition, for each classification, we give out our own confidence in the classification result. If the confidence is above a threshold, we would store the data in our local database for future use.

Generally speaking, our application can be divided into the following modules: feature extractor, classifier, confidence calculator and local data sampling storage. The work flow of our application has been illustrated in detail in the "**ALGORITHM Description"** section.

## V. EXPERIMENTS

Through our experiments, we get many useful parameters as the following three tables.

| | Depth difference between hip center and head per frame |
|---|---|
| Humpback Person | <-0.15 |
| Pregnant Women | >0.09 |

Table 2

| | Difference between the height of the left shoulders and the right shoulders per frame | Portion of time the body keep balanced |
|---|---|---|
| Normal Person | <2 | <36% |

Table 3

| | Difference between the height of the left hand and the right hand per frame |
|---|---|
| People Walking with canes | >15 |

Table4

The following table shows the accuracy for our test results:

| Human Forms | Identification Accuracy |
|---|---|
| Humpback | 90% |
| Neck Issues | 80% |
| Pregnant Woman | 80% |
| Walking with canes | 80% |
| People with crutches | 60% |
| Lame | 90% |

Table 5

## VI. Mathematical Articulation

The followings are the important features we extract from the depth image stream and the skeleton stream:

1) Features for pregnant women and humpback person:

$$\sum_{i=1}^{n} D_i/T$$

(Di stands for the depth difference value per frame, T stands for the number of frames we get)

2) Features for people walking with cane:

$$\sum_{i=1}^{n} |\alpha_i - \beta_i|/T$$

( $\alpha$ stands for the value of hand's X-axis in the coordinate, $\beta$ stands for the value of elbow's X-axis in the coordinate )

$$\sum_{i=1}^{n} |L_i - R_i|/T$$

(L stands for the value of left hand's Y-axis in the coordinate, R stands for the value of right hand's Y-axis in the coordinate )

3) People walking with neck problem:

$$\sum_{i=1}^{n} |S_i - h_i|/T$$

(S stands for the value of spine's X-axis in the coordinate, h stands for the value of head's X-axis in the coordinate )

4) Lame:

$$\sum_{i=1}^{n}((-1)^{LS_i} + (1)^{RS_i})$$

(LS stands for the left feet's state, RS stands for the right feet's state)

The following is how we calcuate the confidence:
If we use three features to make the decision and the corresponding probability for each feature is P1, P2 and P3.
Then the confidence is P = (P1+P2+P3)/3 + P1*P2*P3. If P >= 1, we set P = 1.

If we use two features to make the decision and the corresponding probability for each feature is P1 and P2.
Then the confidence is P = (P1+P2)/2 + P1*P2. If P >= 1, we set P = 1.

If we use one feature to make the decision and the corresponding probability for each feature is P1
Then the confidence is P = P1

## VII. ALGORITHMIC COMPARISONS

As stated above, our algorithm is based on the stick figure streaming from Kinect application, for different model, the algorithm is totally different, for example, some algorithm is based on multi camera system which could be 3D space, and then the feature vector could be totally different.

Our algorithm is a simple version compared to other 2D model based algorithm which has implemented machine learning techniques, however, our algorithm works perfectly if the user's frame images are in the correct fault range, and it is high extensible for other sub problems, finally, it could give the robust and reliable result in real time which is much better than complicated algorithm, especially, those with machine algorithms.

Moreover, our software architecture is high extensible which could be extended to other modules and divide into more detailed modules, especially, like classification module, actually, with small modification, our algorithm could be extended to identify several sub problems at the same time. For example, we could test if a people has hump back with right cane and has neck problem. :-)

## REFERENCES


[1] A. F. Bobick and A. D. Wilson. "A state-based technique for the summarization and recognition of gesture". In Proc. of 5th Intl. Con) on Computer Vision, pages 382-388, 1995.

[2] A. F. Bobick and J. Davis. "Real-time recognition of activity using temporal templates". In Pmc. of IEEE Computer Society Workshop Applications on Computer Vision, pages 39-42, Sarasota, FL, 1996.

[3] J. K. Aggarwal and N. Nandhakumar. "On the computation of motion of sequences of images" - a review. Proc. of the IEEE, 76(8):917-934, 1988.

[4] J.K. Aggarwal, Q. Cai, W. Liao, and B Sabata. Non-rigid motion analysis: Articulated & elastic motion. Accepted by CVGIP: Image Understanding.\

[5] I.A Kakadiaris, D. Metaxas, and R. Bajcsy Active part-decomposition, shape and motion estimations of articulated objects: A physics-based approach. In Proc. CVPR, pages 980-984, Seattle, WA, 1994.

[6] A. Shio and J. Sklansky. Segmentation of people in motion. In Proc. of IEEE Workshop on Visual Motion. IEEE Computer Society, pages 325-332, October 1991.

[7] A. B. Poritz. Hidden Markov Models: A guided tour. In Proc. IEEE Intl Conf. on Acoust. Speech and Singal Proc., pages 7-13, 1988

[8] J, Yamat, J. Ohya, and K Ishii. Recognizing human action in time-sequential images using Hidden Markov Model. In Proc. IEEE conf. CVPR, pages 379-385, Champaign, IL, June 1992

## APPENDIX

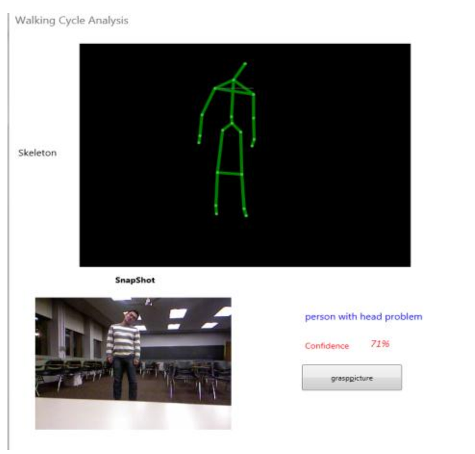
Data sources: Real-time input from Kinect

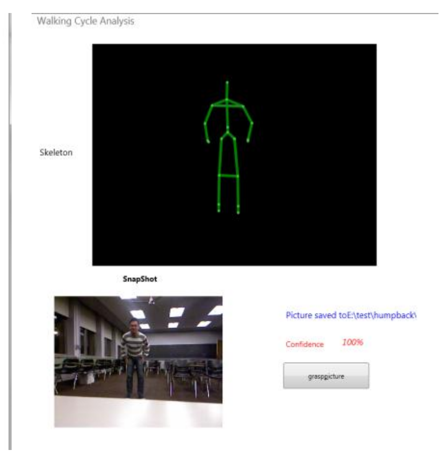
File paths:
E:\test\head_problem***.jpg
E:\test\humpback***.jpg
E:\test\humpback_with_left_cane***.jpg
E:\test\humpback_with_right_cane***.jpg
E:\test\lame***.jpg
E:\test\pregnant***.jpg
E:\test***.jpg

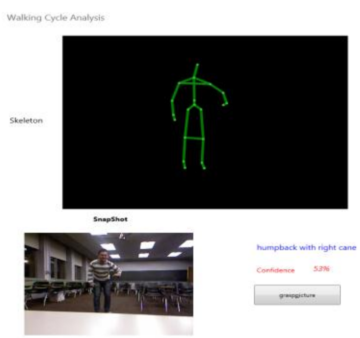
Source code:
Please see the attached source code

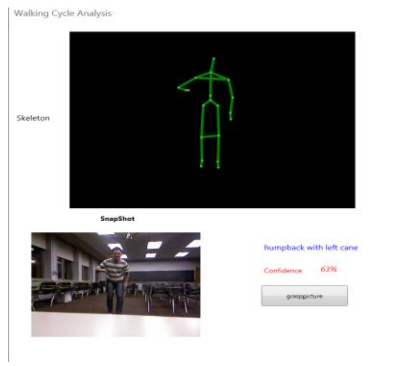
User manual:
Create the file paths stated above before you run the application. Then open our project in VS2010 and click on run. The application will begin to work.

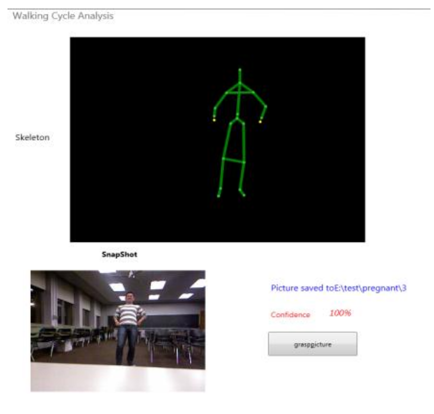
Screenshots of our application:




Walking Cycle Analysis
Skeleton
SnapShot
humpback with right cane
Confidence
53%


Walking Cycle Analysis
Skeleton
SnapShot
humpback with left cane
Confidence
63%


Walking Cycle Analysis
Skeleton
SnapShot
Picture saved toE:\test\pregnant\3
Confidence
100%
graspgicture