HUMAN POSE RECOVERY AND ACTION RECOGNITION

Khandesh Bhange (11196) & Piyush Kumar(11496)
Advisor – Prof. Amitabha Mukerjee
Dept .of Computer Science and Engineering, IIT Kanpur

ABSTRACT

This work explains the implementation of a model which recognizes an action of a human in a video by pose estimation of Articulated Human in static 2-D image. In our project the video is segmented into frames and then features are extracted from each frames. And finally using ***Modified LCS approach*** *we output the action performed by the individual.*

Introduction :

Human pose recovery and action recognition is one of the most fascinating topic in Computer vision. Several methods have been proposed for human pose recovery which uses pictorial structural framework[1] which revolutionized this area of research. For human pose recovery we used local part based models[2] and constrained are imposed on the joints geometrically. This model help us deal with the problems of foreshortening and rotation of limbs.

So Where is action recognition required ?

Action recognition is one of the important topic in computer vision. It is important for the applications such as video surveillance, content-based video search. There are lots of other fields as well.

Why it is challenging?

In real time tracking or just in a static image action recognition is a non-trivial job because of high variability in appearance produced by changes in the point of view, lighting conditions, and number of articulations of the human body.

How to detect these actions??

Fig 1- Bicycle Kick (Ref : http://www.livesportsandnews.com/best-bicycle-kicks/)

Our project has two parts, first part is labelling of the different body parts of human in given image or frame. Second part is to determine the action done by human in a given video. For the second part we used the result obtained in first part.

Related Work:

Pose estimation is very much interesting area in the field of computer vision. Most recent and popular work on pose estimation is by Deva Ramanan and Yi Yang.

Articulated pose Estimation With Flexible Mixtures of parts (By - Yi Yang, Deva Ramanan)[1] :

This paper describes a method for pose estimation in stationary images based on part models. In this method they have used a spring model as a human model and calculated a contextual correlation between the model parts. One way to visualize the model is a configuration of body parts interconnected by springs. The spring like connections allow for the variations in relative positions of parts with respect to each other. The amount of deformation in the springs acts as penalty (Cost of deformation).

Most of the work done on action recognition from video requires RGB as well as Depth data to recognize the action.

An Approach to Pose based Action Recognition (Chunyu Wang, Yizhou Wang and Alan L. Yuille) [2] :

For representing human actions, it first group the estimated joints into five body parts namely Head, L/R Arm, L/R Leg. A dictionary of possible pose templates for each body parts is formed by clustering the poses of training data. For every Action class we distinguish some part sets (Temporal and Spatial) for representing the given action and then find the maximum intersection out of it.

Approach and Algorithm:

Part 1: Labelling different body parts

For this part we used code provided by D. Ramanan[3]. The code basically gives the 26 rectangular boxes which means there is one body part in each of the box. Deva Ramanan model represents the human body as a deformable configuration (like spring is attached between parts) of individual parts which are in turn are modelled separately in a recursive manner. Then for all the configurations of human body parts, score is calculated as follows:

$$\text{score}_i(t_i, p_i) = b_i^{t_i} + w_{t_i}^i \cdot \phi(I, p_i) + \sum_{k \in \text{kids}(i)} m_k(t_i, p_i)$$

Fig 2 : Equation from [1]

Informally we can think of this equation as having two parts, one its local score (the 1st part of rhs) which means how much this part fits in the given position. And a pair-wise score which finds its score when their kids are at a fixed position. Now this score is propagated till its root (head) and the configuration which achieves the highest score is being selected.

So, from this we get 26 parts of human body. But our main goal is to recognize actions based on above data, so all the 26 parts individually doesn't make much sense and increases the computation overhead as well. So to minimize the cost we clustered the 26 body parts into 11 parts which are : Left

Hand/Arm/Torso/Thigh/Leg, Right Hand/Arm/Torso/Thigh/Leg and Head. For clustering this part we first normalized the skeleton w.r.t Head-Neck length. This method helps us deal with the cases of difference in height, length of detected skeleton caused by the different shapes and sizes of individuals. After normalization we used '**Linear Regression'** method to estimate the 11 body parts described above. r

After this we get a model which shows labels for each part of human body. Results obtained on some of the input images is as shown in the following figure:

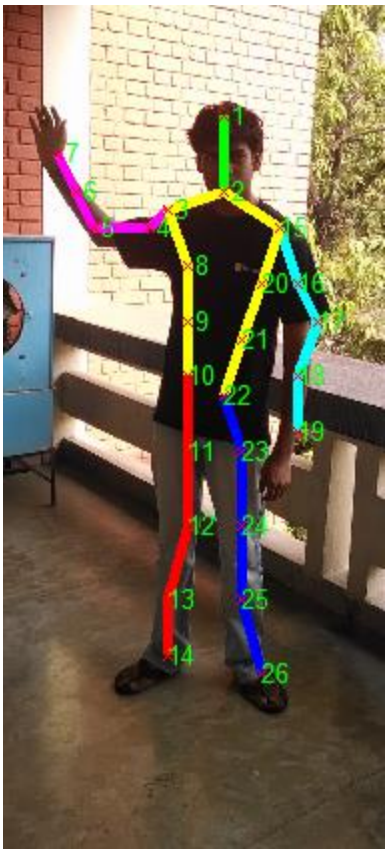


Fig 3 : Labelled 26 body parts

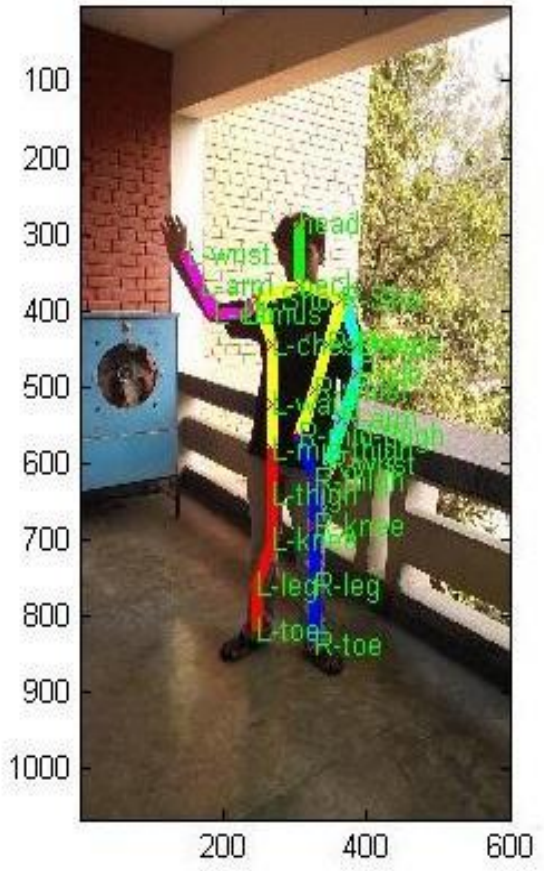


Fig 4 : After clustering Labelled 11 body part.

Part 2: Action recognition from video

Feature Extraction: From above 11 parts, we use the set of 8 angles as the feature vector of one image/frame. These angles are:

Angles between: Left Hand and Arm, Left Arm and Torso, Left Torso and Left Thigh, Left Thigh and Left leg and similarly for Right body parts. For training we used '**i3DPost Multi-view Human Action Datasets [4].**

This provides a dataset of HD image sequences of 8 person (different camera view) doing 13 actions namely Walk, Run, Jump, Bend, Hand-wave, Jump-in-place, Sit-Stand Up, Run-fall, Walk-sit, Run-jump-walk, Handshake, Pull, Facial-expressions.
Currently we have trained our model for actions Walk, Jump, Bend, Hand-wave, and Walk-sit. For training we used idea very similar to codebook, extracted feature vector for each frame of a every training video and stored it.Now when new video(input) comes, we take some frames of that video at some interval and apply modified LCS algorithm to check which action it belongs to. The flowchart given below is a high-level overview of what the whole algorithm is:

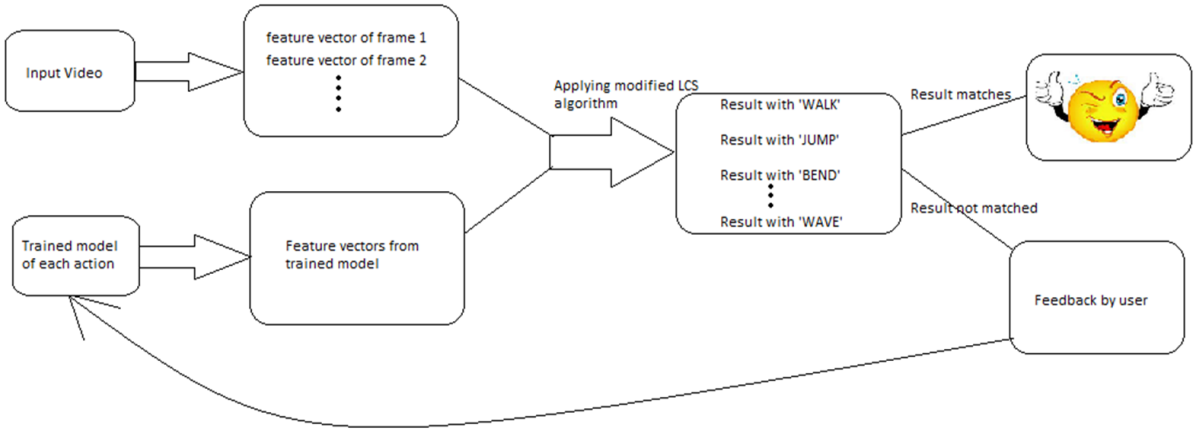


Fig 5 : Our approach

Modified LCS Algorithm: In this algorithm it calculates the maximum number of frames in subsequence matching with that on training sets. This is implemented as a dynamic programming fashion.
Now this modified algorithm will give number of frame-sequences which are matched (within a particular threshold) with trained videos for each action. The more the matched number more the probability of that action. For finding when two frames matches we have used the concept of star distance [5] which is the similarity between two feature vector and is defined as the Euclidean distance between 8 sub-vector as follows :

$$\textbf{Star-Distance} = \Sigma (Si - Ti) \qquad \text{for all } i = 1 \text{ to } 8$$

We used two kind of threshold to deal with the problem of occlusion and noise in human skeleton detection. The first one is local threshold which takes care of small changes on local level on every sub-vector of feature vector and one penalty which deals with high changes in one or two sub-vector in feature vector and ignores them. So if they satisfies the local and global threshold then they are considered to be equivalent.

Feedback System: At the end of test video we will ask the user whether the result is consistent with the provided video or not, if not we ask user the correct action of that video and train our model on this new video accordingly.

IMPLEMENTATION AND EXPERIMENT: To evaluate the performance of the above approach we implemented a model to classify actions in 5 categories basically- Bend, Jump, Walk, Wave and Sit. This system was evaluated on real human videos from the i3D dataset [4] and random videos taken from our normal cameras. A histogram was used to present the classification result. A less number of videos were taken for training and testing due to its high time cost. So the difference is not very substantial but correctness is ensured. The duration of video for training ranged from 60-110 frames as short video doesn't provide much information for action classification and longer video was very time consuming for training and testing.

Examples of 5 action classes :

Fig1. Walk

Fig2. Bend

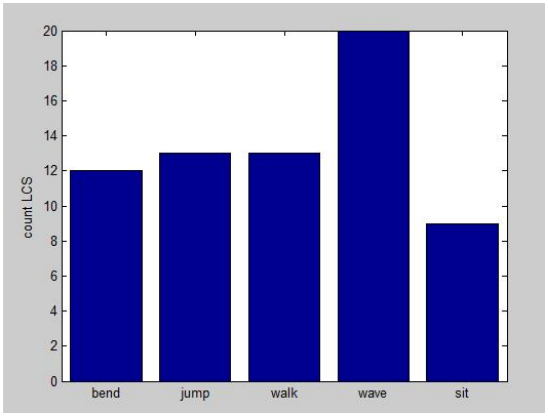
Fig3. Wave

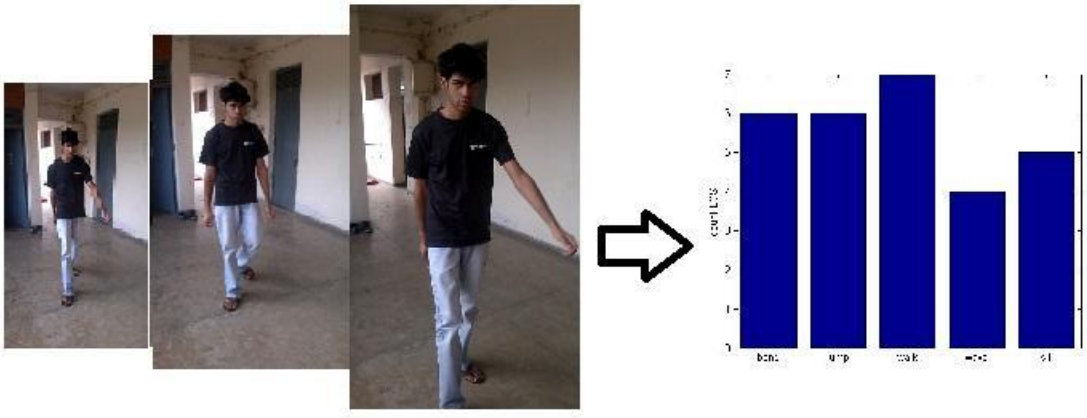
Fig4: Jump

Ref : Above images taken from i3D dataset.

Fig5 : Sit

Results :





Some of the screenshots are shown above for results and there was high accuracy in actions like wave, walk and jump but some action classes like bend and sit were not much accurate. The reason behind this was that, in bend and sit the skeleton was distorted and hence the extracted feature vector was not always correct and there were lots of occlusions which lead to high penalty and lesser **countLCS** in our implementation.

Confusion-Matrix:

	Bend	**Jump**	**Walk**	**Wave**	**Sit**
Bend	**5**	**0**	**2**	**0**	**1**
Jump	**1**	**7**	**1**	**0**	**1**
Walk	**0**	**2**	**9**	**0**	**1**
Wave	**0**	**0**	**2**	**9**	**0**
Sit	**2**	**1**	**1**	**0**	**4**

Accuracy we obtained:

Action	**Wave**	**Bend**	**Sit**	**Walk**	**Jump**
Accuracy	**81.8%**	**62.5%**	**50%**	**75%**	**70%**

Conclusion and future improvements:

We tried to implement a good model for human action recognition based skeleton information human body represented by a 8 dimensional feature vector. We implemented our own algorithm to classify action into 5 categories and were also successful in it. But there are lots of restrictions on our model, like time for classifying one action is quite large. For training itself it takes a lot of time. We have done this classification for single-action recognition and we can use this idea to implement it on a series of actions. For more robust approach we also have to take the depth data in consideration and this can be

implemented as a real time action classifier because of lesser computation cost in second part of our algorithm.

Datasets and code :

We have used '**i3DPost Multi-view Human Action Datasets'** for training and testing purposes. This provides a dataset of HD image sequences of 8 person (different camera view) doing 13 actions namely Walk, Run, Jump, Bend, Hand- wave, Jump-in-place, Sit-Stand Up, Run-fall, Walk-sit, Run-jump-walk, Handshake, Pull, Facial-expression.

In Human pose recovery, they have used Image Buffy dataset and Parse dataset. This Parse dataset contains 305 pose-annotated images full body images of human poses.

References :

1. "**Articulated pose estimation with flexible mixtures-of-parts**"
Y Yang, D Ramanan - Computer Vision and Pattern Recognition (CVPR), 2011

2. "**An approach to pose –based action recognition"**
Chunyu Wang, Yizhou Wang, and Alan L. Yuille (CVPR), 2013

3. Code provided by D. Ramanan :
http://www.ics.uci.edu/~dramanan/software/pose/

4. i3DPost Multi-view Human Action Datasets:
http://kahlan.eps.surrey.ac.uk/i3dpost_action/

5. Human Action Recognition Using Star Skeleton
Hsuan-Sheng Chen, Hua-Tsung Chen, Yi-Wen Chen and Suh-Yin Lee VSSN '06