

HUMAN BODY POSTURE RECOGNITION USING A KINECT SENSOR

Catalina Mocanu ^{1*}

Irina Mocanu ²

ABSTRACT

This paper presents a new hierarchical approach used for human body posture recognition based on histograms of angles and voting schemes. Our approach uses the 3D skeleton information from a Kinect sensor to compute features for classification that are represented by the angles between different body parts. The posture recognition is performed in two steps: first the major posture is obtained (defined by the lower limbs' position) using a two voting scheme process and after that the minor posture is computed (defined by the upper limbs' position) that is performed using a weighted sum of the votes from the involved features.

1. Introduction

Human posture and activity recognition are important fields of computer vision research with applications in many domains. Computer vision is a scientific discipline concerned with developing artificial systems that enable a computer to extract information about the world and its environment from images. The data processed can take different forms: video sequences, 3D views from stereo cameras, depth images or even a combination of RGB and depth images.

Human activity recognition has several domains of application. From sign-language to smart human-computer interaction in ambient intelligence rooms, the number of application areas knows a continuum growth and the solutions proposed are more and more efficient. In a world where tracking and monitoring people is an integral part of everyday life, human activity recognition is a key instrument in the future surveillance methods. In order to emphasize more on the importance of this research field, we will review some of the most important applications of human activity recognition in the following paragraphs.

Firstly, human posture and activity recognition opens new doors in the field of human-computer interaction. Even today, the most common means of interaction with a computer are considered to be the mouse and the keyboard. Despite being everyday objects to most of us, they limit the speed and naturalness of the human brain and body [1]. As a result, gesture and posture recognition play a major part in achieving ease in usability and independence from keyboards and mice by creating systems able to recognize our gestures as commands and react to them accordingly. The goal of combining natural

^{1*} Corresponding author., *Software Engineer, Ixia, Bucharest, Romania*, catalina.mocanu@gmail.com

² *Associate Professor, University POLITEHNICA of Bucharest, Splaiul Independentei No. 313, Bucharest 060042, Romania*, irina.mocanu@cs.pub.ro

gestures to operate the technology around us is to create user friendly technologies that understand the behavior of humans and can be controlled from distance, using no devices, just a simple movement of a body part.

This paper purposes a hierarchical approach on the task of human posture recognition. The configuration of the human body has many degrees of freedom in its joints and the overall shape of a skeleton can vary greatly from a posture to another. We chose this hierarchical approach in order to increase the speed of recognition, since the recognition of the whole-body in a single step depends on a numerous features and thus needs a greater computational power. The proposed method uses a Kinect sensor to retrieve information about the 3D joint positions of the subject. We use a supervised learning algorithm based on probability distribution of features to learn four major postures: standing, sitting on the chair, sitting on the floor and couching.

The system consists of two levels. First, the major posture of the body is recognized using information about the torso and the lower limbs. Then, based on the result of the first classification, we stop or continue. We consider there are no minor postures for the sitting on the floor and crouching major postures, so we continue with the second step of classification only when the major posture detected is standing or sitting on the chair. The second recognition step performs the classification using information about the upper limbs and then the entire classification is presented.

The rest of the paper is organized as follows. Section 2 presents some existing methods for body posture and activity recognition. Section 3 describes the proposed approach, while details about the experimental results of the proposed method can be found in Section 4. Conclusions and future work are presented in sections 5.

2. Related Works

According to [2], activities can be divided in four categories depending on their complexity: gesture, actions, interactions and group activities. Gestures are elementary movements of a person's body part. A gesture is a non-vocal means of communication; it may be created from the movement of the head, hands, arms or body. Examples of gestures are "*stretching an arm*", "*raising a leg*", "*pointing in one direction*". However, there is no universal language of gestures, and that is acceptable in some areas of the world, might be considered offensive in others. Such an example is "*pointing*" which is common in Europe and the United States, but is considered rude in Asia. Actions are one person activities composed of a temporal sequence of gestures. Such activities include "*walking*", "*waving*" and "*punching*." Interactions are activities involving two or more persons and/or an object. For instance, "*two people fighting*" is an interaction between two subjects, while "*stealing a bag*" is an interaction between two subjects and an object. Lastly, group activities are actions performed in a group of people which can interact with objects too. Some examples include "*a group marching*", "*a grouping participating in a meeting*". We are interested in our research only in the types of activities involving a single person: gestures and actions. Postures enter in the first category, being an atomic constituent of any action. Next, we briefly present some single layer approaches, which rely on the sequential characteristic of an activity, and hierarchical approaches, which deal with high complexity activities that are described in the form of sub-events [2].

The article [3] proposes a human activity recognition system based on body joint angles that are mapped on codewords in order to generate discrete symbols for a hidden markov model (HMM) [4] for every activity. After being trained on an activity, all HMMs are used for activity recognition. A stereo camera was used to capture a pair of stereo RGB images at the same time in a manner similar to human eyes. By using the depth information recovered from the stereo images, 3D coordinates of every joint could be recovered. The human body model is made up of 14 body segments, 9 joints (2 knees, 2 hips, 2 elbows, 2 shoulders, 1 neck) and 24 degrees of freedom (2 DOF to the horizontal and vertical direction, respectively, at each joint and 6 DOF for the transformation from the global coordinate system to the local coordinate system at the body's hip). Each segment is represented by an ellipsoid and ellipsoids are joined by kinematic parameters. For the joint angle estimation, the ellipsoids presented above are used to reconstruct the human pose from the stereo images. First, a tracking algorithm is used to locate the position of the moving subject: horizontal movement like walking, or vertical movement like sitting.

The subject's location is used for computing the six parameters of the transformation from the global coordinate system to the hips' local system. Then, by using an Expectation Maximization [5] framework based on several probability relationships and parameters computed from Gaussian distribution, each point in the image is given a label that represents the ellipsoid it belongs to. This process starts with a face detection algorithm to detect the head and torso areas and then moves down to labeling the other body segments.

The features used for detecting the body posture are the joint angles kept in the form of a DOF (degrees of freedom) vector - 6 for the hips (transformations from the global to the local coordinate system) and 2 DOFs (vertical and horizontal) for the other joints. For each activity consisting of several frames a set of feature vectors is created. For robustness of the feature vector space, LDA (Linear discriminant analysis) [6] is applied. LDA is used to compute the best discrimination among different classes by maximizing the ratio of between-class scatter matrix and within class scatter matrix. The HMM requires generation of the codebook of codewords. A vector quantization algorithm (Linde, Buzo, Grey clustering algorithm) is used to generate an efficient codebook of vectors from the training vectors. It starts with a codeword of size one and recursively splits into two codewords, until a convergence criteria is met. Then, in order to obtain a symbol for a sample feature vector, the vector is compared to all codewords and the index of the codeword with the minimum distance to the feature vector is chosen as symbol. So, the index numbers of the codewords are used as symbols for the HMMs, each activity frame being represented by a symbol. The HMMs are trained using these symbol sequences for each of the detected activities.

For the recognition process, the feature set firstly undergoes a process of quantization and then passes through all the HMMs to decide the most likely activity. The recognised activities are: left hand up-down, right hand up-down, both hands up-down, boxing, left leg up-down, right leg up-down, walking, and sitting.

In this article [7], an activity is modeled as a set of poses and velocity vectors for the major body parts that are stored in a set of multi dimensional hash vectors. For each pose an 18 dimensional vector is kept. The vector contains the angles and the velocity angles for 9 body parts: the torso, the upper arms, the lower arms (forearms + hands), the upper legs, the lower legs (calves + feet). The angles are computed based on the 2D projection of the real 3D angles which makes this approach view based. However, good results are obtained even in the case where a subject's position varies with more than 30°.

There are five hash tables used: one for the torso, two for the legs and two for the arms. The hash table for the torso is two dimensional: it contains 1) the angle between the positive x-axis and the major axis of the torso; 2) the angular velocity of the torso. The hash tables for the limbs are four dimensional and consist of: 1) the angle between the positive x-axis and the upper arm/ the thigh; 2) the angle between the positive x-axis and the forearm/the calf; 3) the angular velocity of the upper arm/thigh; 4) the angular velocity of the forearm/calf. The angular velocity is computed as the distance between the angles of two consecutive frames. The entries of the hash tables are represented by pairs of the activity model number and the time instant of that activity model. Each body part may correspond to more than one activity model.

The recognition process consists of three stages: 1) voting for the individual body parts; 2) combining the individual body parts votes for each pose; 3) combining the votes of all the test frames. Since the voting is done for each of the body parts, the system is less susceptible to errors due to occlusions of limbs.

In the first stage, for each body part and for each model activity, a 1D array containing the votes for each k test frame is created. Items in the same hash table bin that correspond to the same pose index may correspond to different activity models or time instants. A vote for a specific body part pose relatively to a index is computed using a logarithmic Gaussian approach [8]. In order to tolerate slight pose variations that may occur for the same activity, it is crucial to consider the neighboring pose bins of the index derived from the poses of the test activity. For this reason, all the frames used for testing a specific activity are analyzed to extract a variance of the angular poses. In the second step, votes from the limbs and torso are added up to obtain a general vote for each frame k . The third step consists of combining the votes for all the test frames. This can be done using temporal or sequential correlation. However, only sequential correlation works for activities performed at different speeds. The activities recognized are: jumping, kneeling, picking up an object, putting down an object, running, sitting down, standing up and walking.

3. System Description

The system aims to perform a hierarchical classification of the human body based on a hierarchy of postures. The postures are organized in two levels. First, there are “*major postures*”: sitting on a chair, sitting on the floor, crouching, standing, standing with one leg raised frontal or to the side. *Major postures* are defined by the position of the legs between themselves and with the torso and they constitute the first level of classification of a human body posture. Then, from two of the major postures, namely for standing and sitting on a chair, we have 16 possible detailed postures with respect to the position of the

arms. These are the “*minor postures*”. We consider that each arm can be positioned in one of the following four positions: up, down, lateral, frontal. So the all the combinations of these 4 positions for each arm create a number of 16 minor postures for the standing and for the sitting on a chair main postures. This is the second level of classification.

The idea for creating a hierarchical body posture recognition method is inspired by the posture hierarchy. Since the human body has a great mobility from its points, and positions of body parts can vary greatly, creating a single recognition layer system would have been more difficult due to the great number of features involved. By using an hierarchical approach, the number of features used per recognition module decreases. Also, deciding between sitting and standing postures, the system is less prone to errors when not taking into account the positions of the arms which may be the same for both.

The system consists of three modules: *the acquisitions module*, *the learning module* and *the recognition module*, as described in Figure 1.

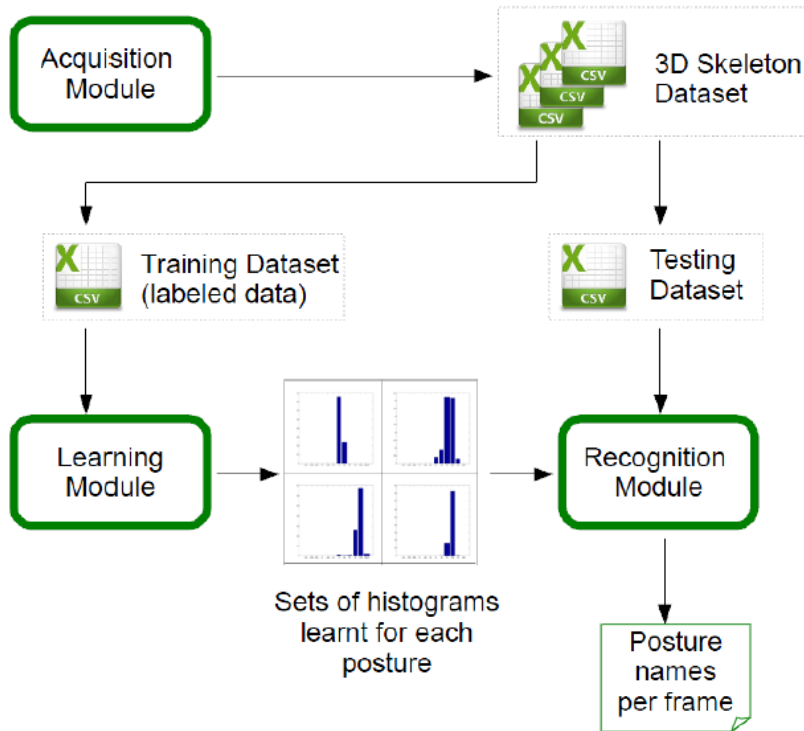


Figure 1. System description

The acquisition module is responsible for gathering skeleton information from the Kinect sensor and preprocessing it in CVS files that are used for training and testing. The CSV files contain the 3D coordinates of 15 joints. *The learning module* uses the training dataset obtained from the acquisition module, processes it in two submodules and outputs a set of histograms for the major and the minor postures' features. The submodules deal with

learning the characteristics of the angles of the upper body and the lower body parts for a specific posture and represent them as a probability distribution. *The recognition module* uses the sets of histograms previously learnt in a two step process of voting. The first step performs the recognition based on the lower body features, while the second step uses upper body features.

The acquisition module (described in Figure 2) consists of four elements: a human subject willing to pose in all the postures we intend to classify, a Microsoft Kinect sensor [9], a computer with Microsoft Kinect SDK [10] software installed and a program for preprocessing the raw data. We will discuss these elements and the way they are linked in the next paragraphs.

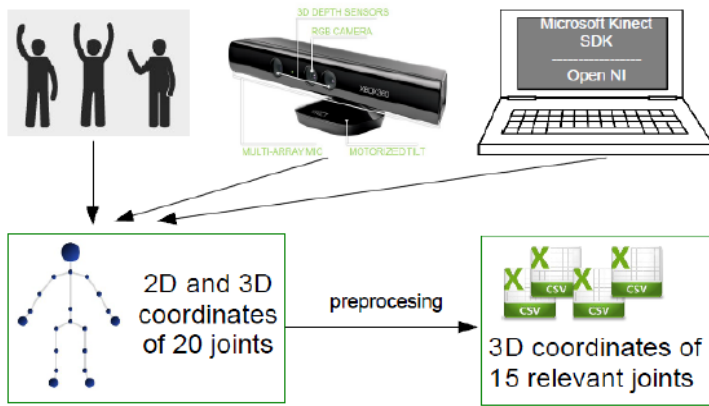


Figure 2. The acquisition module

The acquisition module consists of four components: a human subject willing to pose in all the postures we intend to classify, a Microsoft Kinect sensor, a computer with Microsoft Kinect SDK and a program for preprocessing the raw data.

Our subject had to pose in four major postures: standing, sitting on a chair, sitting on the floor and crouching. For the standing and sitting on a chair postures, we recorded different postures of the arms (up/down/frontal/lateral), while for the standing positions, we also varied the position of one leg at a time (down/raised frontal/raised lateral). In this way, we obtained a great variety of postures. For instance, for the standing posture with both legs down, we have $4 * 4 = 16$ upper body postures, since each arm can take any of the 4 postures mentioned earlier. In the same manner, for the sitting on a chair posture we have 16 variations of the posture. Thus, the number of postures to recognize is close to 40.

The learning module is responsible for processing labeled training data in order to discover some distinctive characteristics of the classes learnt which will later be used in the recognition process. The insides of the learning module are presented in Figure 3.

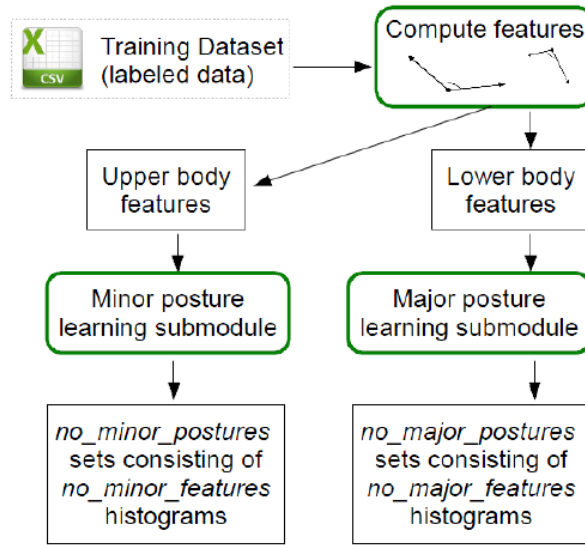


Figure 3.The learning module

The four major positions: standing, sitting on the floor, sitting on a chair and crouching are very easy to describe using only information about the lower body. This gave us the idea of elaborating a recognition system that works in two steps: first, by using information about the lower body parts, we device upon a major posture. Then, by taking into account information about the upper body parts, we are able to a better posture understanding by generating a detailed minor posture.

Because we choose to implement a two layer hierarchical recognition system, the learning phase must be performed for each recognition part separately. Thus, the learning module consists of one learning component for the major posture and one learning component for the minor posture. Because the two learning components use different sets of features for training and the result of one does not depend on the result of the other, the training of both components can be performed in parallel.

The used features consist of angles between body parts. The angles are obtained from 3D joint coordinate information such that they ensure independence of the distance to the camera. Also, by using only angles between body parts, and no angles with the horizontal plane XOZ or the axis, the system is robust to changes of the viewpoint of the Kinect camera.

For the major posture, we use ten features, five for each leg and another feature to characterize the relationship between the two legs:

- the angle between the upper and the lower leg;
- the angle between the upper leg and the torso plane defined by the hips and the spine joints;

- the angle between the lower leg and the torso plane, the angle between the upper leg and the hip line;
- the angle between the upper leg and the hip-shoulder line and the angle between the left ankle, the spine and the right ankle.

It can be noticed that there are no angles with the any of the x, y, z axis, so all features are relative to the body. Thus, independence from the camera position is obtained.

The major posture learning component computes histograms for the following postures: standing (both feet down), standing (left/right leg frontal), standing (left/right leg lateral), sitting on a chair, sitting on the floor and crouching. The minor posture learning component is responsible for learning four possible positions of the hands: up, down, frontal, lateral. The learning is performed for both arms at the same time, and the recognition step is applied on an arm at a time.

The features used for arms are:

- the angle between the upper and the lower arm;
- the angle between the lower and the plane of the torso defined by the shoulders and the spine joints;
- the angle between the lower arm and the torso plane;
- the angle between the upper arm and the shoulder line;
- the angle between the upper arm and the shoulder-hip line.

The learning module transforms the input data represented by 3D coordinates of joints into a set of histograms for each of the learnt posture class. The results of the learning process consist of a number of CSV files equal to the number of different postures learnt by that submodule. The CVS files contain the probability of distribution in the form of histograms for each of the features involved. In other words, for the major postures (we have eight different postures), the learning module outputs eight files containing histograms for each of the ten features involved. For the case of the minor postures, the output consists of four files (up/down/frontal/lateral positions of the arm) with five lines containing the histogram values.

The recognition module is organized in two layers as presented in Figure 4. The decision to use a hierarchical approach has several reasons. First, the number of postures is very high if we take into account all possible combination of the arms' positions. Secondly, using both the features related to the upper body and those related to the lower body in a single recognition step is computationally expensive. Last, two postures may differ based on the position of a single body part and the vote of the body part might be lost if the other features vote for another posture. The classification is performed hierarchically. At the first level, we are interested in classifying in: standing, sitting on a chair, sitting on the floor or crouching. We use only the angles regarding the core and the legs for this classification. At the second level, we are interested in giving a more detailed description of the posture with information regarding the arms. We perform the second classification only if the major posture is standing or sitting on a chair. The minor posture obtained after the second classification provides information about the position of the arms: up, down,

frontal, lateral. The major posture recognition module sums up the votes from all the lower body features, for both legs. We have implemented two approaches for this: a weighted sum of all votes and a two step voting process where each leg votes on itself and then the information is combined. The process is straightforward: for each major posture, each feature gives a vote, representing the likelihood that the feature can have the given value in that posture. A vote is actually the likelihood that a certain angle can be considered natural for a certain posture. From the learning step, the algorithm knows a probability distribution of values of an angle that are considered “normal” for a specific posture. Depending on how much an angle varies from what the system knows to be “normal”, we determine the likelihood that the current feature is representative for each one of the possible posture classes. If the detected posture in the major posture recognition component is standing or sitting on a chair, the minor posture recognition component is applied. The recognition process is performed separately on the left arm and on the right arm based on the weighted voting scheme.

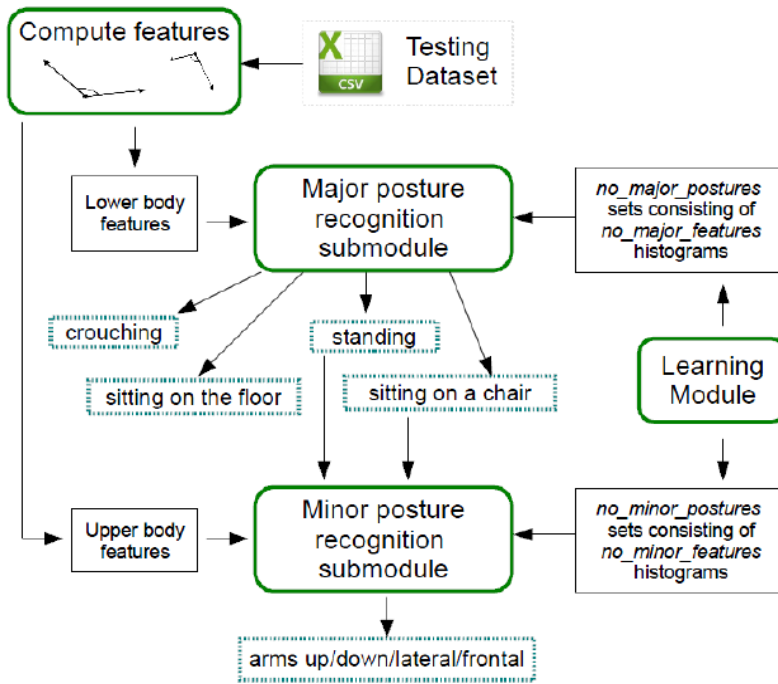


Figure 4. The recognition module

The features for classification are represented by angles between body parts. Angles are obtained from 3D joint coordinate information such that they ensure independence of the distance to the camera. Also, by using only angles between body parts, and no angles with the horizontal plane XOZ or the axis, the system is robust to changes of the viewpoint of the Kinect camera.

For each position, we compute a probability distribution for each feature. We consider that the features can take any value between -180° and 180° . Actually, only the angles

with the torso plane can take negative values due to the orientation of the normal to the plane. All other features have a value between 0° and 180° . We compute the probability distribution using a histogram. Histograms are used to roughly estimate the probability distribution of continuous data by depicting the frequencies of data occurring in a certain range. Because in our training data, the number of training frames differs from a posture to another, we choose to use normalized histograms in the learning process. Thus, the height of a bin shows the likelihood that the feature which fell in that bin is representative of a certain posture.

The body posture classification is a two step process:

- *Major posture recognition*: standing or sitting (either on the chair or on the floor). This first step involves features computing regarding the torso and leg coordinates. There are four possible positions for the arms: frontal, lateral, up, down. The features used for classification in this step use the joints of the arms, the position of the torso and the hips. The recognition is performed using a voting system based on the probability distribution of the features presented above. For the recognition process, two elements are needed: the features from the tested frame and the probability distribution obtained from the histograms. We use two different voting schemes: one method for the minor posture recognition and another for the major posture recognition. The minor posture recognition performs voting on each arm at a time, we use a weighted voting system to determine the position of arms. A new voting scheme, called two-step voting is used for the major posture recognition.

- *Minor posture recognition*: based on a voting process. The first step of the voting process is the computation of the features. Let $no_features$ be the total number of features which take part in the voting at the current level. On the one hand, for the major posture module, we are interested in the lower body features involving the angles made by the lower and upper leg with the plane of the torso. On the other hand, the minor posture recognition module uses upper body features regarding the upper and lower arm in relation to the torso. Let's say the number of possible posture classes is $no_postures$. Then, the system has determined in the learning process a set of histograms for each of these $no_postures$ possible postures. Actually we will have $no_postures$ sets containing $no_features$ histograms, one histogram per feature, per posture. As we know from the previous section, a histogram is defined by the start value, the end value of the data interval and the width of the bin. In our case, features' value vary between -180° and 180° . Based on the learnt histograms, each feature gives a vote to each posture. The vote is a value between 0 and 1 representing the likelihood that, being given a position, the voting feature would have its current value. Let's consider the following notations: f - a feature, v - the value of feature f , p - the posture which receives the vote, $vote_p^f$ - the vote given by feature f to posture p , $histo_p^f$ - the histogram learnt for feature f from all the training data covering posture p .

Then the vote of feature f is: $vote_p^f = histo_p^f[no_bin]$

where no_bin_f is the number of the bin in the histogram to which feature f belongs based its value v . The bin number is computed as follows:

$$no_bin_f = \frac{180 + v}{bin_width} + 1$$

where bin_width is the dimension of the interval.

So, each posture will receive $no_features$ votes computed as described above. For each feature, we create an array containing $no_postures$ elements: the votes given by feature f to all the postures. At this point, the maximum of each vote array may point to a different position. We need to sum up all the votes for a position before deciding upon a posture. The final vote obtained as a weighted sum:

$$vote_p^{final} = \sum_{i=1}^{no_features} vote_p^{f_i} * w_{f_i}$$

where w_{f_i} represents the weight of feature f_i . We chose to consider the weights equal to 1 in our first tests and because we obtained very satisfying results, we kept the weights 1 for the minor posture recognition module. After the final vote is computed for each of the $no_postures$ classes is computed, they are sorted in descending order and the posture with the highest score is chosen.

The voting scheme performed very well on the dataset for minor postures. The minor recognition submodule detects separately the positions of the hands because each arm can be found in one of the next four positions: up, down, frontal, lateral, independently of the position of the other arm. So the weighted voting is applied on the features of the left arm to obtain the position of the left arm, and on the right arm features to detect the position of the right arm. However, this approach was not as good for the major posture recognition submodule. For determining the characteristics of the major posture of the body, we need the features related to solely to the left leg, the features related solely to the right leg and the features that link them. By using the weighted approach, the system's accuracy in detecting the "standing" posture was small due to the fact that there were many false positives in the classes of "standing with left leg frontal", "standing with left leg to side", "standing with right leg frontal", "standing with right leg to side". We tried several weights (equal weights, a double value for the features regarding both legs, a value of 1 for features regarding a single body part, a value of 2 for features regarding two body parts, a value of 3 for features regarding 3 body parts). However, the issue persisted. We then realized that for postures such as "standing with left leg frontal", "standing with left leg to side", the features of the left leg should have a higher weight in the voting process, while the right leg's votes should be more important in postures like "standing with right leg frontal", "standing with right leg to side".

Based on this reason we propose a two step voting process. First, the left and the right leg give their votes to six possible postures: crouching, sitting on the floor, sitting on a chair,

standing, leg raised frontal and leg raised to the side. It should be emphasized that the vote of the left leg for “*leg raised frontal*” or “*leg raised to the side*” is actually the probability for the left leg to be in that position. The left leg cannot vote for the right leg. This is why some features related to both legs should be used to decide upon the best solution. Let f_{left} be a feature concerning just the left leg (the angle between the upper and lower leg, or the angle between the upper leg and the torso plane), f_{right} a feature concerning only the right leg and both a feature containing information about both legs such as the angle between the left ankle, torso, right ankle. For each of these set of features, we obtain a voting array with $no_leg_postures$ columns where $no_postures$ is the number of postures possible per leg.

In the first step of the voting, each of the three sets of features votes on the $no_leg_postures$ possible postures and each returns an array containing their votes. Let $vote_{left}$, $vote_{right}$ and $vote_{both}$ be those vectors. Because we want to eliminate the votes for postures which are very unlikely, we create some new vectors $vote_{left_podium}$, $vote_{right_podium}$ which contain 0 for all the postures that were not voted in top three and keep the other vote values. Below is shown how $vote_{left_podium}$ is computed:

$$vote_{left_podium} = \begin{cases} vote_{left}(i), & \text{if } vote_{left}(i) \text{ in top 3 biggest values in the array} \\ 0, & \text{otherwise} \end{cases}$$

We also want to eliminate all posture which are considered impossible for both legs, but might have received a vote from one leg. In this way, we create a mask, containing values of 1 for “possible” and 0 for “impossible.”

$$mask(i) = \begin{cases} 1, & \text{if } vote_{both}(i) > 0 \\ 0, & \text{otherwise} \end{cases}$$

In the second step, we compute a $vote_{final}$ array:

$$vote_{final}(i) = vote_{left_podium}(i) * mask(i) + vote_{right_podium}(i) * mask(i) + vote_{both}(i)$$

From this new voting array, the posture with the greatest value is considered the winner. If the posture detected is “*leg raised frontal*” or “*leg raised to the side*”, we are interested in finding out the specific leg. This is done easily by summing up the votes for these two postures for each of the legs. If we have a result of “*leg raised frontal*” and the sum computed above is higher for the left leg, then the major posture is “*standing with left leg frontal*”, otherwise is “*standing with right leg frontal*”.

The number of training frames differs from a posture to another, we chose to use normalized histograms in the learning process. Thus, the height of a bin shows the likelihood that the feature which fell in that bin is representative of a certain posture. The probability distribution for the upper body features is computed using a histogram with a

width of 30° and the bins centres ranging from -165° to 165° . The histogram is a distribution of frequencies for some values for a feature when the arm is in a certain position. In order to ensure independence from the number of data in the training set, the histogram is normalized by dividing the frequencies to the total number of training data labeled to that class. The histogram is computed for a position p and all the values of feature f for all skeletons in the training set. For a test frame, the value v of feature f will be mapped to a specific bin of the histogram. The normalized frequency of this bin represents the probability that for position p there may be a skeleton for which feature f is equal to v . We notice that for the "arm down" position, most frequent values of the angle between upper and shoulder-hip line are close to 0° . Also, for the "arm up" position, most frequently the values tend to be close to 135° . Although the values from table in fig. 6 are not strictly obeyed, the difference between positions is noticeable. The probability distribution for the upper body features is computed using a histogram with a width of 30° and the bins centres ranging from -165° to 165° .

There are subtle differences in the values of the angle between the lower and the upper leg for postures such as crouching, sitting on the floor and sitting on a chair. The 30° bin width used for the histograms of upper body feature don't offer the needed degree of detail. So, for the lower body feature histograms we decided upon a bin width of 10° . The first bin's center is -175° and the last bin's center is 175° . For the two-step voting approach, the system is trained for only six positions regarding the position of a leg: crouching, sitting on the floor, sitting on a chair, standing, leg frontal, left leg side. Because the left and the right leg vote independently, in order to obtain greater variety of the training data and more flexibility of the recognition module, we compute the histograms for a posture taking into account the features values of both legs. For instance, for the "leg side" posture, the histogram contains information from the left leg features of the "standing with left leg side" training set and the right leg features of the "standing with right leg side" training set.

4. Experimental Results

The data set consists of a series of CSV files containing the 3D coordinates of 15 Kinect joints (head, neck, left/right shoulder, left/right elbow, left/right wrist, torso, left/right hip, left/right knee, left/right ankle). The files are named using the following format: standing_frontal_lateral.csv which means that: a) the major posture is standing; b) the detail posture indicates a frontal position for the left arm and a lateral position for the right arm. The data set is divided in two parts: 75% of the data represents the training set, while the rest of 25% is the test set. For the detailed classification, there are 16 possible combinations for the arms' position, each arm falling in one of the classes: frontal, lateral, up, down.

As the confusion matrices show, the results between the left arm and right arm recognition are similar. Both perform the worst in the case of "arm frontal" recognition, while classifying very well "arm down" and "arm lateral". For the second case, when common histograms were computed for left and right arm positions, the results seem to be a bit better than those for the right arm confusion matrix, without being too different from the left arm confusion matrix (see Figure 5). So, the best approach is to treat both left arm and right arm cases together in the learning stage, since it will provide a more accurate recognition basis.

Predicted ----- Actual	Left arm frontal	Left arm lateral	Left arm up	Left arm down
Left arm frontal	76.02	2.06	15.07	6.85
Left arm lateral	0	94.90	4.38	0.72
Left arm up	2.36	3.94	89.76	3.94
Left arm down	1.82	0	0	98.18

Predicted ----- Actual	Right arm frontal	Right arm lateral	Right arm up	Right arm down
Right arm frontal	86.03	2.21	8.82	2.94
Right arm lateral	2.13	93.62	4.23	0
Right arm up	8.77	5.26	81.58	4.39
Right arm down	0	0	0	100

Figure 5. Confusion matrices

We tested the eight postures four different weights and confusion matrix for each case is given in Figure 6):

- weights equal to 1 for all features involved in the computation. We will name this weight system w_1 ;
- weights equal to 1 for the features regarding the leg or its relative position to the torso and weight equal to 2 for the features regarding both legs. We will name this weight system w_2 .
- weights equal to the number of participant body parts: e.g. weight equal to 1 for the upper-lower leg, a weight equal to 2 for the upper leg - torso plane angle, a weight of 3 for the feature depicted by the angle made by the left ankle, spine and right ankle. We will name this weight system w_3 ;
- weights equal to the number of participant body parts, except for weight of the features which are related to a single body part. For this, the weight is 4. We will name this weight system w_4 .

Predicted ----- Actual	Crouching	Sitting on a chair	Sitting on the floor	Standing	Standing left leg frontal	Standing left leg side	Standing right leg frontal	Standing right leg side
Crouching	100	0	0	0	0	0	0	0
Sitting on a chair	1.36	97.83	0	0	0	0	0.82	0
Sitting on the floor	11.11	3.03	85.86	0	0	0	0	0
Standing	0	1.84	0.10	73.51	1.73	4.22	1.19	17.41
Standing left leg frontal	0	0	0	0	90.24	9.76	0	0
Standing left leg side	0	0	0	0	0	100	0	0
Standing right leg frontal	0	0	0	0	0	0	100	0
Standing right leg side	0	0	0	0	0	0	0	100

Weight set used: w_1

Predicted ----- Actual	Crouching	Sitting on a chair	Sitting on the floor	Standing	Standing left leg frontal	Standing left leg side	Standing right leg frontal	Standing right leg side
Crouching	100	0	0	0	0	0	0	0
Sitting on a chair	2.44	96.48	0	0	0	0	0.81	0.27
Sitting on the floor	14.14	3.03	82.83	0	0	0	0	0
Standing	0.21	2.59	0	82.60	0.76	2.27	0.54	11.03
Standing left leg frontal	0	0	0	0	90.24	9.76	0	0
Standing left leg side	0	0	0	0	0	100	0	0
Standing right leg frontal	0	0	0	0	0	0	100	0
Standing right leg side	0	0	0	0	0	0	0	100

Weight set used: w_3

Predicted ----- Actual	Crouching	Sitting on a chair	Sitting on the floor	Standing	Standing left leg frontal	Standing left leg side	Standing right leg frontal	Standing right leg side
Crouching	100	0	0	0	0	0	0	0
Sitting on a chair	1.90	97.29	0	0	0	0	0.81	0
Sitting on the floor	15.15	3.03	81.82	0	0	0	0	0
Standing	0.32	2.05	0.11	85.95	0.54	2.59	0.22	8.22
Standing left leg frontal	0	0	0	0	90.24	9.76	0	0
Standing left leg side	0	0	0	0	0	100	0	0
Standing right leg frontal	0	0	0	0	0	0	100	0
Standing right leg side	0	0	0	0	0	0	0	100

Weight set used: w_2

Predicted ----- Actual	Crouching	Sitting on a chair	Sitting on the floor	Standing	Standing left leg frontal	Standing left leg side	Standing right leg frontal	Standing right leg side
Crouching	100	0	0	0	0	0	0	0
Sitting on a chair	0.81	98.37	0	0	0	0	0.82	0
Sitting on the floor	13.13	5.05	81.82	0	0	0	0	0
Standing	0	1.73	0.10	70.28	1.19	9.62	2.59	14.49
Standing left leg frontal	0	0	0	0	85.37	12.19	0	2.44
Standing left leg side	0	0	0	0	0	100	0	0
Standing right leg frontal	0	0	0	0	0	0	92.31	7.69
Standing right leg side	0	0	0	0	0	0	0	100

Weight set used: w_4

Figure 6. confusion matrix using the weighted voting process

It can be observed that for all sets of weights, the recognition of the “standing” posture has the lowest precision. If we consider the precision of a method as the minimum precision in recognizing any of the postures, then the descending order of these approaches’ performance is: w_3 (82.60%), w_2 (81.82%), w_1 (73.51%), w_4 (70.28%). So, the best approach, offering a precision of at least 82% for all postures is the approach where the votes are computed depending on the number of body postures involved in the computation of the feature. The fourth weighting set performs worse than the identity weight vector w_1 . From the confusion matrices, we see that the majority of the misclassified “standing” frames are recognized as “standing with right leg to the side” and “standing with left leg to the side”. This means that the voting system performs poorly on the eight posture set due to postures like “standing with left/right leg to the side”.

The confusion matrix for recognition using a two-step voting scheme shows a minimum accuracy of 82.83% in choosing correctly the posture for a test frame. This assesses that the two step voting scheme outperforms even the best of the four approaches based on weighted voting presented in the previous section. By applying this method, the percent of correctly identified “standing” postures increased from 73.51% (when using weighted voting with weight equal to 1) to 96.42%. Also, a small improvement in the classification of the “sitting on a chair” posture is detectable in the bar chart in Figure 7. However, postures such as “standing with right leg frontal” and “standing with the right leg to the side” suffer a decline in the correctness of the recognition. The two step voting scheme has its advantages and its drawbacks, but, overall, is more robust.

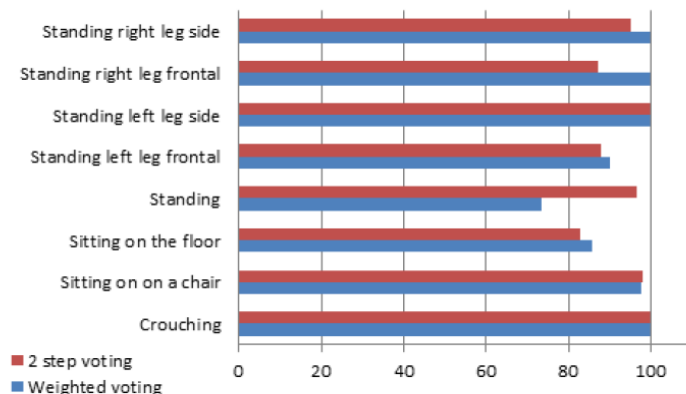


Figure 7. Recognition accuracy between the weighted voting approach and the two-step voting approach

5. Conclusions and Future Work

In this thesis, we propose a hierarchical approach on the task of human posture recognition based on a voting scheme. The postures we intend to classify are also organized hierarchically. The first layer consists of “major postures” - postures defined only by the lower body members and their relation to the torso. Our system is able to classify eight major postures: standing, sitting on a chair, sitting on the floor, crouching, standing with left leg raised to the front, standing with left leg raised to the side, standing with the right leg raised to front, standing with the right leg raised to the side. The second layer is composed of “minor postures” which are in fact variations of the arms’ position.

Each arm can take any of the following positions: up, down, frontal, lateral, thus, resulting in a number of 16 minor postures for each major posture.

Our solution uses the 3D coordinates of the human body joints from the Kinect camera to compute the features for classification. The features consist of the 3D angles made between two body parts: e.g. upper and lower leg, upper leg and torso plane. The system learns through histograms a discrete estimation of the probability of distribution of each feature in each posture. The human body has numerous degrees of freedom in its joints, thus resulting in a high number of features for the entire body. Our hierarchical approach computes and utilizes these features separately, increasing the speed of the process.

As future work The system should be able to work with information from two kinect sensors simultaneously. One of the problems with data from cameras is occlusion of body parts. The kinect camera does not detect a person if the entire body is in the frame, but the head is not. Also, for postures like sitting and like crouching, a view of 45% to the left or to the right increases the chances of good joint position computation. However, in real environments, people will not face the sensor in the “right” manner. By using two kinect cameras positioned at a specific angle, more information could be obtained. We would like to extend our system to use information from more than one kinect in the voting process. Also New postures should be learnt by the system. Human body language is very suggestive of our emotions. Although most emotions are represented through gestures, postures also reveal the state of mind of a person. For ubiquitous computing environments detecting the mood of the subject may be needed.

Acknowledge: The research presented in this paper is co-funded by the national project Mobile@Old, PN-II-PT-PCCA-2013-4-2241 No 315/2014, under the Partnerships Program PN II, powered by MEN – UEFISCDI.

REFERENCES:

1. Pallavi Halarnkar, Sahil Shah, Harsh Shah, Hardik Shah, and Jay Shah. Gesture recognition technology: A review. *International Journal of Engineering Science and Technology*, Vol. 4, Issue 11, pp. 4648 – 4654, 2012.
2. J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, Vol. 43, Issue 3, Article 16, 2011.
3. Md. Zia Uddin, Nguyen Duc Thang, Jeong Tai Kim, and Tae-Seong Kim. Human activity recognition using body joint-angle features and hidden markov model. *ETRI Journal*, Vol. 33, No. 4, pp. 569-579, August 2011.
4. Lawrence R. Rabiner – A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257-286, 1989.
5. M. R. Gupta and Y. Chen - Theory and Use of the EM Algorithm, *Signal Processing*, Vol. 4, No. 3, pp. 223-296, 2010.
6. Yu, H.; Yang, J. - A direct LDA algorithm for high-dimensional data — with application to face recognition, *Pattern Recognition*, Vol. 34, Issue 10, pp. 2067–2069, 2001.
7. Jezekiel Ben-Arie, Zhiqian Wang, Purvin Pandit, and Shyamsundar Rajaram - Human Activity Recognition using Multidimensional Indexing. *IEEE*, pp. 1091-1104, August 2002.
8. Crow, Edwin L.; Shimizu, Kunio (Editors) - *Lognormal Distributions, Theory and Applications*, Statistics: Textbooks and Monographs, 1988.
9. Microsoft Kinect sensor, <https://dev.windows.com/en-us/kinect/develop>, accessed November 2015
10. Microsoft Kinect SDK, <http://www.microsoft.com/en-us/download/details.aspx?id=44561>, accessed November 2015