Recognition of Static Gestures using Three-Dimensional Chain Codes using Dominant Direction Vectors

Jose FIGUEROA [b], Jesus SAVAGE [a] and Ernesto BRIBIESCA [b]
Enrique SUCCAR [c]

[a] *Universidad Nacional Autonoma de Mexico, Department of Electrical Engineering*
[b] *Universidad Nacional Autonoma de Mexico, Instituto de Investigaciones en Matematicas Aplicadas y Sistemas, Department of Computer Science*
[c] *Instituto Nacional de Astrofisica, Optica y Electronica, Department of Computer Science*

Abstract. This paper presents a novel method for static gesture recognition based on three-dimensional chain codes which are computed from three-dimensional skeletons acquired by three-dimensional vision sensors, such as Microsoft Kinect^{TM}. The method has two stages: a digitization stage and a recognition stage. The digitization stage is based on the orthogonal direction change chain code, which represents the changes of direction on segments of a three-dimensional curve which was fitted in a three-dimensional grid; these changes of direction are invariant to rotation and translation. The recognition stage is based in the detection of dominant changes of direction on segments of a three-dimensional curve which was fitted in a three-dimensional grid. The experiments for testing this method of static gesture recognition involved recording the pose of the arms of a subject who was standing at increasing distances and the body was oriented in frontal and three-quarters angles; the poses were matched against a set of reference arm poses which were taken from a frontal body pose at an specific distance. The results show that the generation and matching of three-dimensional chain codes of arm poses captured on varying configurations are reliable enough for being used for gesture recognition. The gesture recognition system will be used on the robots of the Robocup@Home teams Markovito and Pumas from Mexico.

Keywords. Feature extraction, Chain code, Machine vision, Three-dimensional Curve, People detection, Pattern recognition, Gesture recognition

Introduction

Human beings have several means of communicating ideas to their peers: speech, which relies on emitting sounds with the vocal organs; writing, which relies on

drawing symbols on a surface; and gesturing, which relies on the presentation or movement of body parts.

In daily life, people use gestures in order to give more information when a person is speaking or for transmitting ideas when the situation does not allow to speak, either because loud noises overwhelm speech loudness, silence is required to perform a task, the interlocutors are too far from each other, or any of the interlocutors is unable of speaking.

The interaction between a human being and a computer usually is done by devices which must be manipulated directly, such as keyboards, mice, touch screens, digitizing tablets, trackballs, joysticks, just to name a few. Another way of interacting with a computer without relying on manipulable input devices is by using voice commands, which are recorded by a microphone and the computer interprets the sound information in order to get commands from it.

The advantage of using gestures is that they work in environments where voice acquisition for verbal commands is complicated or infeasible, as well as gestures provide a more universal way of inputting commands because voice input is too dependent on both the accent and the pronunciation of the person who is speaking.

The scope of this article is focused in presenting a novel way of representing and recognizing human pose data from vision and motion capture systems.

The representation of human pose data depends on the source data which is being used for tracking the human being. For the case of bi-dimensional image data, the ways of acquiring human pose are: binary silhouettes obtained either by background removal, color segmentation or depth segmentation [8,13,10,9,1]; invariant features extracted from image gradient [11], or three-dimensional joint data acquired through motion capture [15,14,6,7] or by probabilistic recognition of the position of the body parts using depth data [12]. According to the source data and the scope of the work, the representation of the pose data has varied formats, such as directed histograms of pixels [8], mathematical models for body parts [13], edge oriented histograms [10], sets of images containing key poses [9], residual vectors [1], SIFT-like features [11], differences of joint angles [15], quaternions of each joint [14], statistical data from acceleration and angular data [6], or vectors of angle features [7]. All these representations of pose data are used as input for algorithms of temporal pattern recognition, such as Markov Models, State Machines, Neural Networks, or Support Vector Machines.

The advantages of this method are the following: 1. the chain codes represent three-dimensional orthogonal direction change, which makes the representation of a three-dimensional curve invariant to translation and rotation; 2. the chain codes can represent any three-dimensional curve in a compact fashion, as a string of chain codes; 3. the computation of the chain codes and the computation of the dominant direction changes use vector math and basic math, so they can be optimized easily for computer architectures which lack of floating-point units.

This work was supported by PAPIIT-DGAPA UNAM under Grant IN117612.

1. Proposed Approach

The proposed method for static gesture recognition consists of two stages, a digitization stage, where the three-dimensional joint data is converted to a discrete three-dimensional curve, and a recognition stage, where the measure of similarity between two discrete three-dimensional curves is computed in order to know if two discrete three-dimensional curve have similar shapes.

1.1. Orthogonal Direction Change Chain Code

The digitization stage is based on the orthogonal direction change chain code [2,5], which digitizes three-dimensional curves into a set of codes which represent orthogonal direction changes between three constant length segments of a three-dimensional curve (u, v, w) (Equation 2), which is aligned to the vertices of a three-dimensional grid of a constant cell size.

In order to convert a set of three-dimensional lines into a set of constant length segments, the first step consists in aligning the vertices of each line, p and q, to the corners of the three-dimensional grid, by rounding the values of p' and q', according to the smallest distance of each axis and two neighbouring vertices of the grid (Equation 1), getting the vertices p' and q' as a result.

Most of the time, the length of the line is longer than the size of the cell of the grid, additional points are added to the line by means of linear interpolation, whose amount is the Chebyshev distance of p' and q', $D_{Chebyshev}(p', q') = \max(|p' - q'|)$. The interpolated points are adjusted to the grid using the Equation 1.

$$v'_x = \begin{cases} g_x, & if\ g_x \leq vx < \frac{g_x + g_{x+1}}{2} < g_{x+1}; \\ g_{x+1}, & if\ g_x < \frac{g_x + g_{x+1}}{2} \leq vx \leq g_{x+1}; \end{cases} \tag{1a}$$

$$v'_y = \begin{cases} g_y, & if\ g_y \leq vy < \frac{g_y + g_{y+1}}{2} < g_{y+1}; \\ g_{y+1}, & if\ g_y < \frac{g_y + g_{y+1}}{2} \leq vy \leq g_{y+1}; \end{cases} \tag{1b}$$

$$v'_z = \begin{cases} g_z, & if\ g_z \leq vz < \frac{g_z + g_{z+1}}{2} < g_{z+1}; \\ g_{z+1}, & if\ g_z < \frac{g_z + g_{z+1}}{2} \leq vz \leq g_{z+1}; \end{cases} \tag{1c}$$

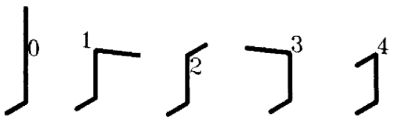
Once all the points are adjusted to the grid, the next step is to split the coordinates of the lines between two consecutive points to make a set of single line segments. And, the final step is to assign chain codes by taking three consecutive single line segments, starting from the first line segment, and apply the rules of orthogonal direction changes to compute the corresponding chain element (Figure 1b).

There are five different orthogonal direction changes for representing any three-dimensional curve (Figure 1) [4]:

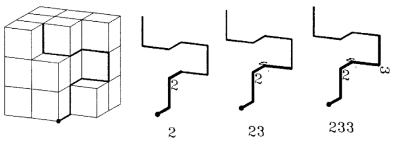
- The Chain Element "0" represents a direction change which *goes straight* through the contiguous straight-line segments following the direction of the last segment.

- The Chain Element "1" represents a direction change to the *right*.
- The Chain Element "2" represents a direction change *upward* (stair-case fashion).
- The Chain Element "3" represents a direction change to the *left*.
- The Chain Element "4" represents a direction change which is *going back*.

$$chain\,element(u,v,w) = \begin{cases} 0, & if\, w = v; \\ 1, & if\, w = u \times v; \\ 2, & if\, w = u; \\ 3, & if\, w = -(u \times v); \\ 4, & if\, w = -u \end{cases} \tag{2}$$



(a) Chain Elements, by number



(b) Example of Chain Code Sequence

Figure 1. Orthogonal Direction Change Chain Codes

1.2. Digitization of Three-dimensional joint data

The three-dimensional joint data is captured by a three-dimensional vision system, such as the Microsoft Kinect^{TM}, which acquires the joint data by analysing the depth map captured by the sensor. The joint data is organized in a humanoid skeleton hierarchy.

For purposes of this work, the joints are numbered in this way (Figure 2):

0) Head (0); *1*) Neck (1); *2*) Left Shoulder (2); *3*) Left Elbow (3); *4*) Left Hand (4); *5*) Right Shoulder (5); *6*) Right Elbow (6); *7*) Right Hand (7); *8*) Torso (8) *9*) Left Hip (9); *10*) Left Knee (A); *11*) Left Foot (B); *12*) Right Hip (C); *13*) Right Knee (D); *14*) Right Hip (E).

The skeleton data is converted to a series of chain codes, which are merged in a tree structure (Figure 3), using the notation described in [3].

1.3. Dominant Direction Changes

The first attempts of recognizing static gestures were based on the work of [4], which describes an algorithm to compute the measure of dissimilarity between two discrete three-dimensional curves (A and B), which are composed of a series of orthogonal direction change chain codes, based on the quantification of the differences between partial common couples contained in two curves. An implementation of the former algorithm was made in C++ and tested with sets of chain codes, the results of the matching had large amounts of false positives, specially between the poses for the arm being extended to the front and the arm being lowered down, whose reference poses which had a matching rate close to 90% between both poses.

Due to these problems, alternative approaches for analyzing the chain codes to determine poses are researched. Through visual analysis of the chain codes, it is pointed out that the chain codes of the three-dimensional curves for each reference pose have different distributions of orthogonal segments. As an example, the chain code for the left arm when is extended to the side has a large proportion of orthogonal segments whose directions are parallel to the positive horizontal axis $+X$, while the chain code for the right arm when is extended to the side has a large proportion of orthogonal segments whose directions are parallel to the negative horizontal axis $-X$ (Table 1).

Direction	Dominant Axis
Left	$+X$
Right	$-X$
Up	$+Y$
Down	$-Y$
Back	$+Z$
Front	$-Z$

Table 1. Dominant Axis for each Orthogonal Direction

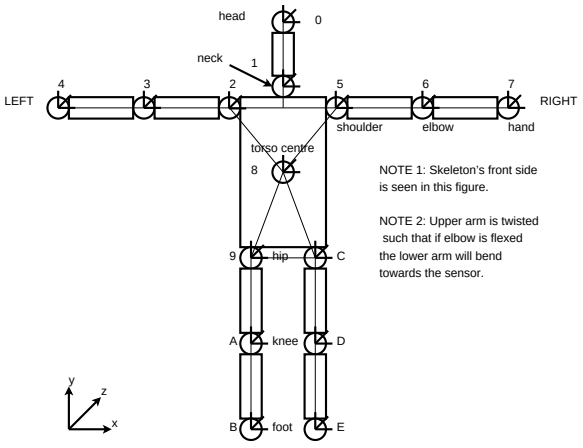
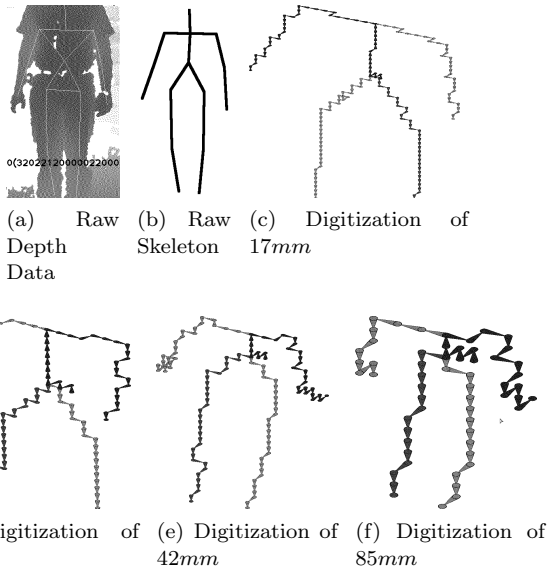


Figure 2. Three-dimensional Joint Data Hierarchical Structure



(a) Raw Depth Data (b) Raw Skeleton (c) Digitization of $17mm$ (d) Digitization of $29mm$ (e) Digitization of $42mm$ (f) Digitization of $85mm$

Figure 3. Skeleton Processing. Figures 3c–3f shows the trees of chain codes at different resolutions, which are computed from raw skeleton data (Figure 3a, 3b).

In order to determine the Dominant Direction Vectors of a chain code, the chain code needs to be converted to a set of orthogonal segments with a known direction. In order to do that, two direction vectors are given arbitrarily, whose only restriction is that they must be orthogonal to each other. For example, $u = (0, 1, 0)$ and $v = (0, 0, 1)$. The next direction vector, w, is computed by applying the rules for chain codes to each character of the chain code (Equation 2). Each direction vector, including the arbitrary direction vectors used at the beginning, is stored in a list of vectors V. Every time that the direction vector w is stored in the list of vectors, the direction vectors u and v have their values reassigned to $u = v$ and $v = w$.

Once all the direction vectors are stored in the list of vectors, the next step is to compute the percentages of each direction, P, by dividing the count of the vectors on each direction contained in the list of vectors, $\{V_{+X}, V_{-X}, V_{+Y}, V_{-Y}, V_{+Z}, V_{-Z}\}$, between the total amount of vectors in the list of vectors, $|V|$ (Equation 3). The Dominant Direction Vectors, D, are obtained from the two largest percentages on P (Equation 4).

$$P = \begin{cases} P_{+X} & = \frac{|V_{+X}|}{|V|} \\ P_{-X} & = \frac{|V_{-X}|}{|V|} \\ P_{+Y} & = \frac{|V_{+Y}|}{|V|} \\ P_{-Y} & = \frac{|V_{-Y}|}{|V|} \\ P_{+Z} & = \frac{|V_{+z}|}{|V|} \\ P_{-Z} & = \frac{|V_{-z}|}{|V|} \end{cases} \tag{3}$$

Figure 4. Skeleton Joint Tree structure used in this work.

$$D = \begin{cases} D_1 & = \max(P) \\ D_2 & = \max(P \setminus D_1) \end{cases} \tag{4}$$

2. Experiments

The purpose of the experiments is to test the reliability of computing the Dominant Direction Vectors of three-dimensional chain codes for recognizing pose data from body parts, specifically both arms, captured by three-dimensional vision sensors and stored as video files.

2.1. Configuration

The vision sensor is set to a height of $1000mm$ above the ground, later the test subject is set at a certain distance from the vision sensor and is asked to perform a pose with the arm extended, in a determinate body orientation angle. The set of distances are $\{2000mm, 2500mm\}$, the set of poses for each arm is *lowered down, extended to the front, and extended to the side*; and the orientation angles set are $0°, 45°, -45°$, where $0°$ is the frontal angle camera of the subject and $45°$ is the subject set on a three-quarters orientation. The height of the vision sensor does not allow to capture poses where the arms are raised above the head.

The joints of the skeleton are used in relative coordinates, using the joint of the torso as reference. The reference size for the resolution of the three-dimensional grid for computing the three-dimensional chain codes is set to $170mm$, because is the length of the vector which is formed by the coordinates of both head and neck joints. By using this reference size, other resolutions, which are fractions of the reference size, are selected for the experiments, the resolution set is: $\{17mm, 29mm, 42mm, 85mm\}$; the purpose of testing multiple resolutions for the three-dimensional grid is to figure out at which resolutions the computation of the Dominant Direction Vectors becomes unreliable while keeping close to real-time performance. A subset of the joints of the skeleton is used to determine the pose of each arm, without the forearm: torso, neck, shoulder and elbow.

The tree which describes the skeleton has the structure shown in Figure 4. The tree is formed by four sequences of joints which have the torso centre as its root. This configuration was chosen because the height of the vision sensor does not allow acquiring the joint of the head when a person is at a close distance ($<= 1500mm$).

2.2. Orientation Angle of the Body

An issue with the matching of chain codes from skeletons captured by three-dimensional vision sensors is the orientation angle of the torso of the subject: the visual analysis of skeletons made of orthogonal direction vectors shows that the same pose has varying proportions of orthogonal direction vectors, according to the orientation angle of the subject to the camera, which changes the apparent pose of each limb. For example, when the subject is in front of the camera and has both arms extended to the side, each arm has a large percentage of orthogonal direction vectors on the X axis and a small percentage of orthogonal direction vectors in the Z axis, which would match to a pose where a limb is extended to the side; however, when the subject is in a three-quarters angle, the percentage of orthogonal direction vectors on the X axis decreases and the percentage of orthogonal direction vectors on the Z axis increases, which would match to a pose where the limb is extended to the front or it would match to a pose where the limb is extended to the back.

In order to avoid that, the orientation angle of the subject is computed from the normal vector N of the triangle formed by the joints of the torso, the left shoulder and the right shoulder, $p1, p2, p3$. From that vector, the orientation angle θ is computed using N_x and N_z (Equation 8). The numerical value of the normal vector of the torso, when the subject is in front of the camera, is 90°, therefore, when the subject is not oriented towards the camera, the complementary angle ϕ must be computed to rotate each joint of the skeleton, whose coordinates are set relative to the torso joint, in order to have the angle of the normal vector in the numerical value of the frontal camera angle (Equation 9).

$$U = p3 - p1 \tag{5}$$

$$V = p2 - p1 \tag{6}$$

$$N = U \times V \tag{7}$$

$$\theta = \tan(\frac{N_z}{N_x}) \tag{8}$$

$$\phi = 90° - \theta \tag{9}$$

As the chain codes for each arm are start at the joint of the torso, a reference set of direction vectors can be added to reduce the ambiguity of the chain codes, in this case the following sequence of direction vectors was used $\{(0,1,0),(0,0,1),(0,1,0),(0,0,1)\}$, which is converted to the chain code sequence 22. With this sequence, the line between the torso joint and the neck joint can be formed by a set of orthogonal direction vectors with the values $(0,1,0)$ if the resolution is high enough.

2.3. Arm Pose Matching

The matching process consists in computing the Dominant Direction Vector of the chain code sequence for each arm, using the algorithm described in the Subsection

1.3. The algorithm was applied with a variation in the way the direction vectors are counted: at high resolutions, the line between the torso joint and the neck joint can be formed by a set of positive vertical orthogonal direction vectors with the values $(0, 1, 0)$, which can be used as a reference to find the origin of the shoulders by finding any orthogonal direction vector whose value is different to $(0, 1, 0)$.

That sequence of consecutive positive vertical orthogonal direction vectors can be ignored in order to limit the proportion of positive vertical orthogonal direction vectors to those vectors which belong to poses where the arm is raised above the level of the shoulders. Once an orthogonal direction vector which is different to a positive vertical vector is found, at the level of each shoulder, the count for computing the Dominant Direction Vectors is started.

For each pose there is a set of Dominant Direction Vector which are expected (Table 2).

2.4. Results

In the testing stage, the subject is set in the distance and angle sets which are mentioned in the section 2.1 and 200 skeleton samples of each arm pose are captured, whose chain codes are computed later and matched against the reference chain codes.

Both accuracy and running time are assessed for each configuration of the arm pose tests. In the matching accuracy tests (Table 3–8), the overall best results were achieved when the resolution became higher (Table 3, 8). As for the average running time of the matching, this value increases with the resolution, the tests performed with the highest resolution show an average time of 0.003 milliseconds, which is good enough for real-time applications. These results can be compared against a set of results from control data, which was obtained from the Dominant Direction Vectors of raw join data captured by the Kinect (Tables 9 and 10).

A problem with the generation of chain codes for the skeleton is that either by the noise on the depth map or by the lack of complete information in the skeleton, it is not possible to generate a root node for the skeleton tree, which is used as reference during the matching process. The lack of a root node reduces the length of the chain code and increases the probability of a mismatch.

Another problem with the generation of the skeleton comes when the subject has the arms extended towards the camera, as the visibility of the elbow is reduced, OpenNI is unable of computing the coordinates of the joint of the elbow correctly, which alters the proportion of orthogonal direction vectors over the $-Z$

Pose	First Dominant Vector	Second Dominant Vector
Left Arm Side	+X	+Y or -Y
Right Arm Side	-X	+Y or -Y
Left Arm Down	-Y	+X
Right Arm Down	-Y	-X
Left Arm Front	-Z	+X
Right Arm Front	-Z	-X

Table 2. Expected Dominant Axis for each Arm Pose

axis, resulting in a set of Dominant Direction Vectors which does not identify the pose for arms extended towards the camera (Tables 5 , 6).

Pose	Arm	Accuracy (%)				Average Running Time (ms)			
		17mm	29mm	42mm	85mm	17mm	29mm	42mm	85mm
Down (0°)	Left	99.50	99.50	99.50	8.50	0.0032	0.0027	0.0031	0.0019
	Right	99.50	99.50	99.50	21.00	0.0033	0.0027	0.0028	0.0020
Down (45°)	Left	99.50	99.50	99.50	15.50	0.0033	0.0026	0.0026	0.0020
	Right	99.00	79.50	99.50	2.00	0.002609	0.002178	0.002024	0.001532
Down (−45°)	Left	96.50	96.50	95.50	96.50	0.002509	0.002055	0.002063	0.001504
	Right	98.00	88.50	99.50	96.00	0.002763	0.002201	0.002094	0.001609

Table 3. Matching Accuracy of Down Pose (2000mm Away, Relative Coordinates)

Pose	Arm	Accuracy (%)				Average Running Time (ms)			
		17mm	29mm	42mm	85mm	17mm	29mm	42mm	85mm
Down (0°)	Left	99.50	98.00	99.50	89.00	0.0032	0.0027	0.0024	0.0020
	Right	99.50	99.50	99.50	98.50	0.0036	0.0029	0.0025	0.0020
Down (45°)	Left	99.50	99.50	99.50	99.50	0.0041	0.0027	0.0025	0.0020
	Right	95.00	94.50	94.50	86.00	0.002658	0.002189	0.001855	0.001675
Down (−45°)	Left	99.50	99.50	99.50	99.50	0.002614	0.002142	0.001996	0.001539
	Right	96.50	94.00	95.50	94.00	0.002938	0.002240	0.002022	0.001568

Table 4. Matching Accuracy of Down Pose (2500mm Away, Relative Coordinates)

Pose	Arm	Accuracy (%)				Average Running Time (ms)			
		17mm	29mm	42mm	85mm	17mm	29mm	42mm	85mm
Front (0°)	Left	48.00	49.50	48.00	48.00	0.0034	0.0026	0.0025	0.0022
	Right	10.00	9.00	2.00	0.00	0.0031	0.0025	0.0026	0.0020
Front (45°)	Left	95.00	95.00	95.00	94.00	0.0032	0.0027	0.0025	0.0020
	Right	93.00	94.00	91.00	95.00	0.002712	0.002112	0.001945	0.001611
Front (−45°)	Left	94.00	94.00	94.00	94.00	0.002425	0.002004	0.002022	0.001568
	Right	9.00	7.00	10.00	0.00	0.002591	0.002194	0.001996	0.001596

Table 5. Matching Accuracy of Front Pose (2000mm Away, Relative Coordinates)

Pose	Arm	Accuracy (%)				Average Running Time (ms)			
		17mm	29mm	42mm	85mm	17mm	29mm	42mm	85mm
Front (0°)	Left	7.00	4.00	4.00	4.50	0.0036	0.0026	0.0024	0.0020
	Right	59.00	58.00	54.50	37.00	0.0026	0.0021	0.0019	0.0018
Front (45°)	Left	80.50	81.50	76.50	78.00	0.0033	0.0026	0.0024	0.0020
	Right	95.50	96.00	95.50	56.00	0.002656	0.002068	0.001906	0.001539
Front (−45°)	Left	74.00	74.00	74.00	74.00	0.002045	0.001673	0.001521	0.001311
	Right	25.50	26.50	21.50	17.00	0.002951	0.002296	0.002073	0.001678

Table 6. Matching Accuracy of Front Pose (2500mm Away, Relative Coordinates)

Pose	Arm	Accuracy (%)				Average Running Time (ms)			
		17mm	29mm	42mm	85mm	17mm	29mm	42mm	85mm
Side (0°)	Left	99.50	99.50	99.50	99.50	0.0033	0.0026	0.0025	0.0020
	Right	99.50	99.50	99.50	99.50	0.0036	0.0030	0.0025	0.0021
Side (45°)	Left	99.50	99.50	99.50	99.50	0.0034	0.0028	0.0025	0.0020
	Right	99.50	99.50	99.50	99.50	0.002543	0.002012	0.001858	0.001557
Side (−45°)	Left	99.50	99.50	99.50	99.50	0.002799	0.002337	0.002001	0.001578
	Right	99.00	99.50	99.50	99.50	0.002779	0.002217	0.001947	0.001516

Table 7. Matching Accuracy of Side Pose (2000mm Away, Relative Coordinates)

Pose	Arm	Accuracy (%)				Average Running Time (ms)			
		17mm	29mm	42mm	85mm	17mm	29mm	42mm	85mm
Side (0°)	Left	99.50	99.50	99.50	99.50	0.0035	0.0027	0.0026	0.0020
	Right	99.00	99.00	99.00	99.00	0.0035	0.0028	0.0028	0.0021
Side (45°)	Left	99.50	99.50	99.50	99.50	0.0034	0.0027	0.0024	0.0020
	Right	99.50	99.50	99.50	99.50	0.002473	0.002009	0.001911	0.001529
Side (−45°)	Left	99.00	99.00	99.00	99.00	0.002838	0.002281	0.002258	0.001645
	Right	99.50	99.50	99.50	99.50	0.002658	0.002101	0.001876	0.001529

Table 8. Matching Accuracy of Side Pose (2500mm Away, Relative Coordinates)

Pose	Arm	Accuracy (%)			Average Running Time (ms)		
		0°	45°	−45°	0°	45°	−45°
Down	Left	99.50	99.50	99.50	0.00682752	0.00801804	0.00938817
	Right	99.50	99.50	99.50	0.00494681	0.00566779	0.00618352
Front	Left	95.00	94.00	94.50	0.00858765	0.00839521	0.00764601
	Right	95.00	88.00	95.00	0.00609884	0.00608858	0.00538813
Side	Left	99.50	99.50	98.00	0.00745101	0.00843626	0.0073612
	Right	98.50	98.50	98.50	0.00538813	0.00630154	0.00556003

Table 9. Stats of Control Group (2000mm Away, Relative Coordinates)

Pose	Arm	Accuracy (%)			Average Running Time (ms)		
		0°	45°	−45°	0°	45°	−45°
Down	Left	99.50	99.50	99.50	0.00714568	0.00681213	0.00861843
	Right	99.50	99.50	99.50	0.00526497	0.00509563	0.00664279
Front	Left	96.50	74.00	84.00	0.00859021	0.00659147	0.00799238
	Right	96.50	73.00	83.50	0.00606036	0.00482879	0.00537273
Side	Left	98.50	99.00	99.50	0.00916751	0.00706357	0.00788206
	Right	99.00	99.50	99.50	0.00611937	0.00560108	0.00540865

Table 10. Stats of Control Group (2500mm Away, Relative Coordinates)

3. Conclusions

The tests show that the matching of skeletons generated by three-dimensional vision sensor, by means of a set of chain codes generated by high-resolution three-dimensional resolution grid, using only the chain codes of arm poses when the subject is in front of the camera can be used for gesture recognition. The matching accuracy is reliable enough to use it as input for gesture recognition using probabilistic signal analysis models, such as Hidden Markov Models; and the matching running time is fast enough to use the chain code matching algorithm for real-time applications. This gesture recognition system will be used on the robots of the Mexican RobocupHome teams Markovito (INAOE) and Pumas (UNAM).

References

[1] S. Aslam, C. F. Barnes, and A. F. Bobick. Video action recognition using residual vector quantization and hidden markov models. In *IPCV'10*, pages 659–666, 2010.

[2] E. Bribiesca. A chain code for representing 3d curves. *Pattern Recognition*, 33(5):755 – 765, 2000.

[3] E. Bribiesca. A method for representing 3d tree objects using chain coding. *J. Vis. Comun. Image Represent.*, 19:184–198, April 2008.

[4] E. Bribiesca and W. Aguilar. A measure of shape dissimilarity for 3d curves. 2008.

[5] E. Bribiesca and C. Velarde. A formal language approach for a 3d curve representation. *Computers & Mathematics with Applications*, 42(12):1571 – 1584, 2001.

[6] E. Guenterberg, H. Ghasemzadeh, V. Loseu, and R. Jafari. Distributed continuous action recognition using a hidden markov model in body sensor networks. In *Proceedings of the 5th IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS '09, pages 145–158, Berlin, Heidelberg, 2009. Springer-Verlag.

[7] A. K. Koc. Body posture analysis in the context of shopping. Master's thesis, TU Delft, June 2011.

[8] S. Koceski and N. Koceska. Vision-based gesture recognition for human-computer interaction and mobile robot's freight ramp control. In *Proc. 32nd Int Information Technology Interfaces (ITI) Conf*, pages 289–294, 2010.

[9] C.-M. Oh, M. Z. Islam, and C.-W. Lee. Pictorial structures-based upper body tracking and gesture recognition. In *Proc. 17th Korea-Japan Joint Workshop Frontiers of Computer Vision (FCV)*, pages 1–6, 2011.

[10] J.-W. Park, Y.-C. Lee, B.-S. Jo, and C.-W. Lee. Virtual playing ground interface using upper-body gesture recognition. In *Proc. 17th Korea-Japan Joint Workshop Frontiers of Computer Vision (FCV)*, pages 1–5, 2011.

[11] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-markov model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008.

[12] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297 –1304, june 2011.

[13] M. Sigalas, H. Baltzakis, and P. Trahanias. Gesture recognition based on arm tracking for human-robot interaction. In *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, pages 5424–5429, 2010.

[14] J. Sung, C. Ponce, B. Selman, and A. Saxena. Human activity detection from rgbd images. Technical report, Department of Computer Science, Cornell University, Ithaca, NY 14850.

[15] J. K. T. Tang, J. C. P. Chan, and H. Leung. Interactive dancing game with real-time recognition of continuous dance moves from 3d human motion capture. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '11, pages 50:1–50:9, New York, NY, USA, 2011. ACM.