The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

3D Hand Trajectory Segmentation by Curvatures and Hand Orientation for Classification through a Probabilistic Approach

Diego R. Faria and Jorge Dias

Abstract — **In this work we present the segmentation and classification of 3D hand trajectory. Curvatures features are acquired by (*r, θ, h*) and the hand orientation is acquired by approximating the hand plane in 3D space. The 3D positions of the hand movement are acquired by markers of a magnetic tracking system [6]. Observing humans movements we perform a learning phase using histogram techniques. Based on the learning phase is possible classify reach-to-grasp movements applying Bayes rule to recognize the way that a human grasps an object by continuous classification based on multiplicative updates of beliefs. We are classifying the hand trajectory by its curvatures and by hand orientation along the trajectory individually. Both results are compared after some trials to verify the best classification between these two kinds of segmentation. Using entropy as confidence level, we can give weights for each kind of classification to combine both, acquiring a new classification for results comparison. Using these techniques we developed an application to estimate and classify two possible types of grasping by the reach-to-grasp movements performed by humans. These reported steps are important to understand some human behaviors before the object manipulation and can be used to endow a robot with autonomous capabilities (e.g. reaching objects for handling).**

I. Introduction

Robotics is moving towards to the research and development of technologies that permit the introduction of the robots in our daily life. To create such applications some problems need to be solved, including grasp strategies. Applications of service robots will require advanced capabilities of grasping and skills that allow a robot to grasp different types of objects in different ways. Some of the most performed actions by humans in their daily activities involve the handling of objects for a specific task. The study of human reach-to-grasp movements is important for researches of different areas. In computer science field, hand trajectories segmentation and classification are useful for human-machine interaction using gestures to interact with machines, e.g. the hand can be used as computer mouse. Various theories have been proposed for predicting hand trajectories. Hand trajectory segmentation and classification are useful also in the robotics field for imitation learning for human-robot interaction. Typically, the global hand's trajectory during a manipulation task can be segmented in different stages: reach, lift, transport and release [1]. We focus our attention in the reach stage (reach-to-grasp movement). Our intention is developing an automated system for trajectories segmentation and classification by a probabilistic approach. In this work we want to show the estimation and classification of reach-to-grasp movements when someone is performing the grasping. Analyzing these movements we can be able to understand some human behaviors during the hand journey to reach and grasp an object. This information can be used to endow robots using the movements before the object manipulation, i.e. using it as capability of a robot recognizing how a human grasp an object to imitate his action. This methodology can also be applied for gesture recognition tasks.

This work is partially supported by the European project HANDLE ICT-23-16-40. Diego Faria is supported by Portuguese Foundation for Science and Technology. Diego Faria and Jorge Dias are with Institute of Systems and Robotics – University of Coimbra – Polo II, 3030-290 Coimbra, Portugal (diego, jorge)@isr.uc.pt.

II. Related Work

Hand trajectories have been studied in different areas such as neuroscience, robotics, ergonomics, etc. In [2] is described a modelling approach for 3D hand trajectories in reaching movements. The authors use Bézier curves for geometrical interpretation. Their purpose is to describe a modelling approach to show how the trajectories depend on some predictors and how they vary from repetition of the trajectories. Bayesian models have been used in [3] to classify gestures from images sequences. Tracking of human hands and face are used based on skin-color features. The application is for human-robot interaction. The human actions are interpreted and mapped to the robot actions. They have contributed also with Laban Movement Analysis that helps to identify useful low-level features and to develop a classifier of expressive actions. Images sequence were used in [4] for hand tracking and hand shape representation when a person is gripping a mug. They proposed a method for hand shape representation that characterises the finger-only topology of the hand using cepstral coefficients. Techniques of speech signal processing were used for that. The work shows hand shape recognition classified as top-grab, side-grab, flat-hand and handle-grab when the hand is close to object. In our previous work [5] we developed an application to segment a trajectory to find features like up, down and line for its classification. We have used second order derivative to analyze the evolution of the trajectory finding features using just the x and y axis of a 3D trajectory ignoring other features like diagonal, forward and backward directions. The classification results were satisfactory but we reached undesired results as false negative and classification of the trajectory with low probability.

III. Experimental Setup and Context

It was used the Polhemus Liberty tracker [6] to track the humans hands trajectories. It was attached five sensors on a glove to acquire the 3D hand trajectory. Another sensor was placed on the object to have a priori knowledge of the object position. The setup for the experiments is composed of a wooden table, without any metallic parts, since the magnetic tracker is sensitive to nearby ferromagnetic materials. The experiments are executed by a subject standing in front of the table for the reaching tasks. The tabletop is 50cm by75cm and is placed at a height of 100cm. The object is placed on the center of the tabletop in a marked region for all experiments having the object in the same position. The magnetic tracker emitter unit that determines the frame of reference for the motion tracking system is placed on another table near to the object table. There is no any specific area for a subject starts the trajectory to the target. Usually the subject is positioned close to the table varying the distance until one meter far from the object. Fig.1 shows our current scenario and configuration for this application. Two reach-to-grasp movements were defined for this application: Top-Grasp and Side-Grasp (Fig.2). The Side-Grasp happens when a person wants to grasp the object by its side or by its handle. The Top-Grasp usually happens when someone want to grip the object by its top just to displace it

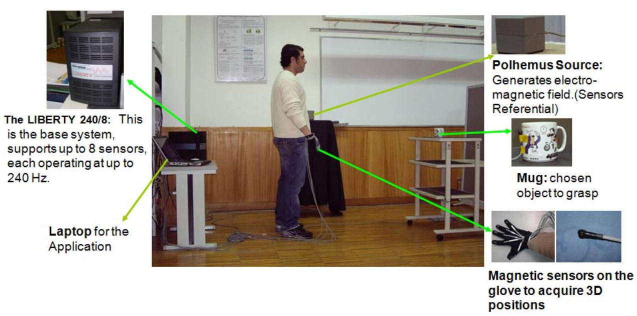

Fig.1. Scenario for our application: environment setup.

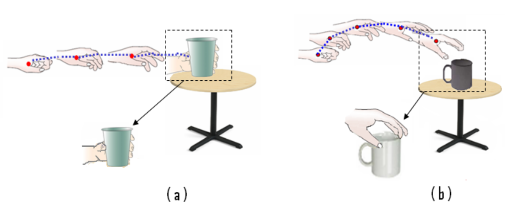

Fig.2. (a) – Side-grasp; (b) Top-grasp.

IV. Trajectories Segmentation

A. Pre-Processing step

We are not considering temporal analysis of the trajectory to avoid some problems. For example, if a movement was learned with trajectories performed in 10 seconds, and when a movement of same type is performed slowly, in 20 seconds, then this movement will not be considered as the same of the learned one, due the features do not correspond. We are considering the spatial information. Even considering the spatial information we can find some difficulties to classify the same type of movement with different distances. The subjects can start the trajectories in different places reaching different sizes of trajectories yielding different scales which can harm the results. To solve the problem, we are normalizing all trajectories to have the size 1. To extract the features we are splitting the trajectory in 8 similar parts (each one representing a hand displacement) to detect the features and then for each part we can characterize the movement due the types and amount of detected features. The division of the trajectory in 8 parts was chosen empirically, it could be 10 or 12 parts that would have the same effect in the classification. This way, the movements can be initialized from different positions with different velocities without harm the results. To achieve the trajectory normalization, for all points of each axis (x, y, z) is applied the following equation to rescale it:

$$R = \left(\frac{X}{\max - \min}\right)(cur - \min) \tag{1}$$

Where ***R*** is the rescaled point; ***X*** is the new size of the trajectory (in our case the size of the trajectory is 1); ***max*** represents maximum value of the raw data found in the current axis, ***min*** the minimum value found and ***cur*** is the current value that is being normalized.

A trajectory smoothing is also necessary. For each point of each axis is calculated the mean value among its previous four neighbours and its four forward four points. Fig.3 shows an example of smoothing.

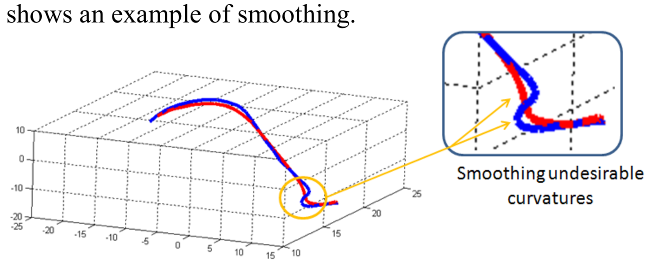

Fig3. Smoothed trajectory: Blue color – raw data; Red color – result.

B. Segmentation by Trajectory Curvatures

As long as the trajectory is in 3D space, for better curvature detection we can work in cylindrical **(*r, θ, h*)** or spherical coordinate system **(*r, θ, φ*)**. Using two points of the trajectory we have the vectors representation in 3D space. The angle formed between these two vectors by the projection on (*x, y*) plane we achieve the ***θ*** angle which give us the pan information, if the angle is increasing, we have the curvature *left,* or if it is decreasing we reach the curvature *right*. The same 2 vectors and their formed angles by the projection on (*z, y*) plane, we can achieve ***φ*** angle for tilt information. In a 3D space we can make some combinations of the possible directions, for example, we have *up* and *down* reached by ***h***, *left* and *right* reached by ***θ*** and *further* and *closer* reached by ***r***, so that we can have several combinations of features. For our application we intend just to detect features like *up, down, left, right, up-left, up-right*, *down-left, down-right* and *no-movement*, restricting others information like *closer* and *further*. This information could be used in other goals, e.g. analyzing if a subject is displacing the object closer or further after the grasping. We can reach the height information (***h***) in a simpler way using the cylindrical coordinate system, calculating the difference between the z axis values from both points. In spherical coordinate system just the φ angle can not give us the height or diagonals movements, being necessary verify also the radius (r), if it is increasing or decreasing and φ angle did not change, this way, we reach this information. To know *up* or *down*, φ and r change and θ remains the same. In cylindrical coordinate system we need to combine r, θ and h to know features like *up-right*, *up-left*, *down-right* and *down-left*.

The curvature segmentation is performed at each two points of the trajectory. The next steps demonstrated by the following equations show us how to reach (r, θ, φ) in spherical coordinate system:

$$r1 = \sqrt{x_1^2 + y_1^2 + z_1^2} \tag{02}$$

$$\sin\varphi = \frac{\sqrt{x_1^2 + y_1^2}}{r1} \tag{03}$$

$$\cos\varphi = \frac{z_1}{r1} \tag{04}$$

$$\varphi_1 = \arctan 2(\sin\varphi, \cos\varphi) \tag{05}$$

$$\cos\theta = \frac{x_1}{\sqrt{x_1^2 + y_1^2}} \tag{06}$$

$$\sin\theta = \frac{y_1}{\sqrt{x_1^2 + y_1^2}} \tag{07}$$

$$\theta_1 = \arctan 2(\sin\theta, \cos\theta) \tag{08}$$

Then with the second vector acquired by the second 3D point we follow the same steps that are used in equation 02 until equation 08 achieving r_2, φ_2 *and* θ_2. After that, we reach θ angle and tilt information (height) given by φ angles as follows:

$$h = r_2 \cos\varphi_2 - r_1 \cos\varphi_1 \tag{9}$$

$$\theta = \theta_2 - \theta_1 \tag{10}$$

In the cylindrical coordinate system, to find the height information, we can simplify eliminating the equations (02) to (06) and we can replace the equation (9) by:

$$h = z_2 - z_1 \tag{11}$$

To find the feature c we use the following rules:

$$c = \begin{cases} height > 0 & \text{and} & \theta \sim 0 & \text{and} & r_{(x,y)} \sim 0 & \text{Up} \\ height < 0 & \text{and} & \theta \sim 0 & \text{and} & r_{(x,y)} \sim 0 & \text{Down} \\ height = 0 & \text{and} & \theta > 0 & \text{and} & r_{(x,y)} = 0 & \text{Right} \\ height = 0 & \text{and} & \theta < 0 & \text{and} & r_{(x,y)} = 0 & \text{Left} \\ height > 0 & \text{and} & \theta > 0 & \text{and} & r_{(x,y)} \sim 0 & \text{UR} \\ height > 0 & \text{and} & \theta < 0 & \text{and} & r_{(x,y)} \sim 0 & \text{UL} \\ height < 0 & \text{and} & \theta > 0 & \text{and} & r_{(x,y)} \sim 0 & \text{DR} \\ height < 0 & \text{and} & \theta < 0 & \text{and} & r_{(x,y)} \sim 0 & \text{DL} \end{cases} \tag{12}$$

Where $r_{(x,y)}$ is the radius in cylindrical coordinate system represented in (x, y) plane. It is reached as follows:

$$r_{(x,y)} = r_{2(x,y)} - r_{1(x,y)} \tag{13}$$

Where r_1 and r_2 is given by:

$$r_{1(x,y)} = \sqrt{x_1^2 + y_1^2} \tag{14}$$

$$r_{2(x,y)} = \sqrt{x_2^2 + y_2^2} \tag{15}$$

If h, θ, and r are equal to zero, then there is no movement. Splitting the trajectory we can characterize the trajectory so that each part can differentiate the type of grasp. After curvatures detection is computed the probability distribution of these features in each part of the trajectory. For each feature is computed its probability as follows:

$$P(c_i) = \frac{c_i}{C} \tag{16}$$

Where c_i represents the amount of a specific curvature in a specific hand displacement (trajectory part) and C is the total of curvatures found in each part, i.e., the summation of the total of occurrence of all c_i.

C. *Segmentation by Hand Orientation*

Using three sensors on three fingertips we can approximate the hand plane computing its orientation to find out if it represents top or side-grasp (Fig.4). We have used the three parallel fingers (index, middle and ring) that usually remain parallel in the most part of hand shape for grasping. These three 3D points form the hand plane and after computing the normal of the hand plane we compare it with the z axis of the Polhemus referential to know the hand orientation. For the hand orientation the strategy of splitting trajectory in 8 parts is kept. At each 3 points in each part of the trajectory we can compute the hand orientation. In each part of the trajectory is found the amount of hand orientation for side and top-grasp. The probability of each one is computed as follows:

$$P(o_i) = \frac{o_i}{O} \tag{17}$$

Where o_i represents the amount of a type of hand orientation (side or top grasp) in a specific trajectory part and O represents the total of occurrences of all hands orientation found in a specific trajectory part.

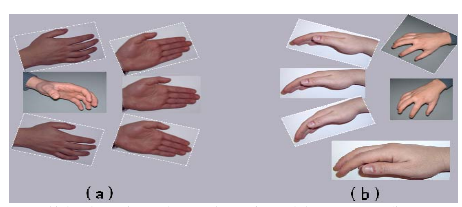

Fig.4. (a) Possible hands orientation for side-grasp; (b) Possible hands orientation for top-grasp.

D. *Experimental Results of the Segmentation Step*

For each observation of our dataset were created xml files that stores the characterization of the trajectory, i.e. segmentation information: features amount and their probabilities in each part. It was created 2 xml files for each trajectory, one with curvatures segmentation and another with hand orientation segmentation. This information is useful to perform the histogram learning that will be used in the classification step. Fig. 5 shows an example of top-grasp trajectory. Table 1 shows the result of trajectory segmentation by hand orientation acquired from the trajectory shown in Fig. 5. The same process shown in table 1 is done for the segmentation by curvatures.

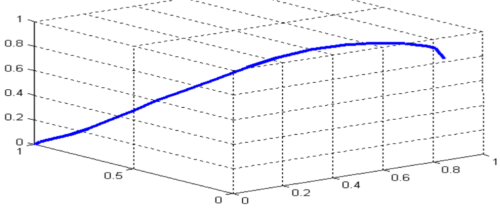

Fig.5. Top-grasp trajectory after smoothing and normalization.

Trajectory Parts	Hand Orientation Side - Top	Hand Orientation. Probab. Side - Top
1	5 - 4	0.56 – 0.44
2	3 - 8	0.28 – 0.72
3	4 - 7	0.37 – 0.63
4	3 - 8	0.28 – 0.72
5	2 - 10	0.17 – 0.83
6	1 - 11	0.08 – 0.92
7	1 - 13	0.07 – 0.93
8	1 - 16	0.06 – 0.94

Tab.1. Trajectory Segmentation by Hand Orientation: Result of our application for the trajectory shown in fig.5. The second column is the amount of features found in each part; the third column is the correspondent probability of each features.

V. LEARNING AND CLASSIFICATION

The learning phase is based on histogram of the segmented features. Some studies have motivated us to apply Bayesian method to classify human movements. Computational models for human perception and action has been explored by researches. Some studies about human brain reports that Bayesian methods have achieved success in creating computational theories for perception and sensorimotor control [7].

A. *Grasping Learning Table*

In the learning phase is analyzed all trajectories of our dataset. Given a set of observations to represent a type of Grasping ***G***, at some displacement ***D***, we have the probability of each type of curvature ***C*** in each part of a trajectory represented as ***P(C | G D)***. The same rule is used for hand orientation learning, so that we have ***P(O| G D)*** where ***O*** represent all possible hand orientation. The learned table is a mean histogram calculated from all top grasp and all side grasp probability tables acquired in the segmentation process. Each type of grasping has its specific learning table. Fig.6 shows 2 examples of the Grasping Learning Tables obtained after analysing all trajectories of our dataset. Due the learning be achieved through histogram is possible some features might have zero probability, because they never have been observed. Whenever these features with zero probability occur in the classification step, the correspondent hypothesis(es) will receive also a zero probability. Our classifier is continuous, based on multiplicative update of beliefs and this situation leads to definite out-rule of the hypothesis. To avoid this problem we are using the Laplace Succession Law, i.e., producing a minimum probability for non-observed evidences by the equation below:

$$P(F=i)=\frac{n_i+1}{n+[F]} \tag{18}$$

Where ***F*** represents the features (e.g. curvatures = 9, orientation = 2); $\boldsymbol{n_i}$ represents total of occurrence of this feature; ***n*** represents the total of all features occurrence.

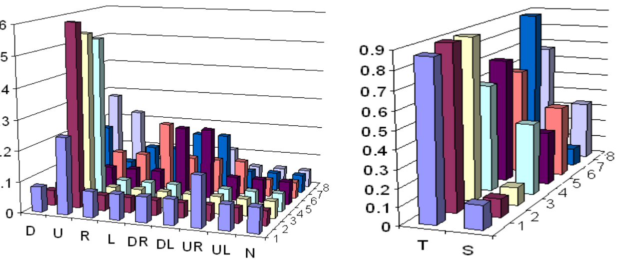

Fig.6. Left image represents the top-grasp curvatures learning table and right image top-grasp hand orientation learning table.

B. *Classification Model Using Bayesian Techniques*

Bayesian classification models have already proven their usability in gesture recognition systems as we can see in [3]. Based on this study we present a Bayesian classification of grasp types analyzing reach-to-grasp movements. The estimation and classification of a type of grasp happens along of a trajectory that is being performed by a subject. In each determined hand displacement is updated the probability of each type of grasp, i.e. the application informs us which grasping is more probable to happen by the higher probability between top and side grasp variables. Assuming the trajectory that is being performed has size 1 due we know the trajectory size a priori, i.e. we have the initial hand position and the mug position given by the sensors, then at each hand displacement corresponding by 1/8 of the trajectory is shown the probability of each grasp type. To understand the General Grasping Classification Model some definitions are done as follows: ***g*** is a known grasp from all possible ***G*** (Grasp types); ***c*** is a certain value of feature ***C*** (Curvature types); ***i*** is a given index from all possible hand displacement composed of *a distance* ***D*** *(* 1/8 of a trajectory) of the learned table.

The probability **P(*c* | *g i*)** that a feature ***C*** has certain value ***c*** can be defined by learning the probability distribution **P(*C* | *G D*)**. Knowing **P(*c* | *G i*)** and the prior **P(*G*)** we are able to apply Bayes rule and compute the probability distribution for ***G*** given the hand displacement ***i*** of the learned table and the feature ***c***. Initially, the grasp variables (priors) ***G*** are a uniform distribution and during the classification their values is updated applying Bayes rule shown in equation below:

$$\mathrm{P}(\mathrm{G}_{k+1} \mid \mathrm{c}_{k+1}, \mathrm{i}) \,\alpha\, \mathrm{P}(\mathrm{c}_{k+1} \mid \mathrm{G}, \mathrm{i})\, \mathrm{P}(\mathrm{G}) \tag{19}$$

We assume the same model of classification for hand orientation which is differentiated just by segmentation information, that is, the hand orientation instead of curvatures, where ***o*** is a certain value of feature ***O*** (hand orientation for side and top grasp). Knowing **P(*o* | *G* *i*)** and the prior **P(*G*)** we apply Bayes rule as follows.

$$P(G_{k+1} \,|\, o_{k+1}, i) \,\alpha\, P(o_{k+1} \,|\, G, i)\, P(G) \qquad (20)$$

We formulate the equation as recursive way. The posterior probability of a previous trajectory part becomes the prior for the next trajectory part (next hand displacement). Assuming that each hand displacement we can find new curvatures and new hand orientation then we can express the online behaviour by using the index *k* that represents a certain displacement performed by the person in the reach-to-grasp movement. The rule for classification is based on the highest probability value being necessary reaching a certain threshold (e.g. 0.7). We expect that a reach-to-grasp movement that is being performed by a subject to grasp the mug by top or side grasp will produce a grasp hypothesis with a significant probability.

C. Entropy as Confidence Level for Classification Fusion

The Shannon entropy ***H*** [8] as a measure of the uncertainty associated with a random variable is used in several works; we can see examples in [9] and [10]. In this work is used entropy as confidence level to try to improve and reach a better classification based on results of previous classification. After analyzing the classifications results of trajectories by hand orientation and by curvatures, we can apply entropy to verify the best classification between both. For that, a confidence variable will be used as weight ***w* Є {w1 , …, *w_N*}** for each model of classification. The weight ***w*** can be expressed as a prior ***P(w)*** in the Bayesian rule. For each model of classification we can compute the entropy of the posterior probabilities as follows:

$$H(P(G \,|\, F\, D)) = -\sum_i P(G_i \,|\, F\, D) \log(P(G_i \,|\, F\, D)) \qquad (21)$$

Where ***P(G| F D)*** represents the posterior probability of each model of classification. The variable ***i*** represents the index of each classification results, that is, after ***n*** trials of one model of classification, we can apply the equation (21) for these results. Through the entropy *H* we can achieve the probability distribution of the weights of each classification (e.g. by curvatures and hand orientation). The weights are computed as follows:

$$w = 1 - \left(\frac{Hc}{\sum_{i=0}^{n} H_i} \right) \qquad (22)$$

Where ***w*** is the weight result; H_C is the current value of entropy that is being transformed in a weight***; i*** represents the index for each entropy value.

Given the confidence of classification we can fuse the classification belief using the weights reached by the entropy. For each part of a trajectory we can compute the equation for the classification fusion achieving a new form of classification:

$$P(G \,|\, F\, D) = \sum_{j=1}^{n} P(w_j) P(g_j \,|\, f\, i) \qquad (23)$$

Where ***P(g | f i)*** represents the classification result of each hand displacement computed as shown in equation 19 and 20. Then at each part of a trajectory that is being classified is computed first the equations (19) and (20) and then the equation (23) which we already got the weight value for each model of classification based on previous results. The index ***j*** represents each type of classification, in this case for hand orientation and trajectory curvatures. Each kind of classification is multiplied for its correspondent weight.

D. Experimental Results of Learning and Classification

Fig. 7 shows a side grasp trajectory performed by a subject and table 2 shows the answer of our application along this trajectory classifying it by curvatures detection and by hand orientation. It shows the probability updated by Bayes rule for both variables (top and side) in each part of the trajectory. The final probability in the last part of the trajectory (in the 8^{th} part) is the result of the classification. Comparing this case of Fig.7, we can see that both classifications reached good performance classifying correctly the trajectory. In this experiment the classification by curvatures was better than by hand orientation.

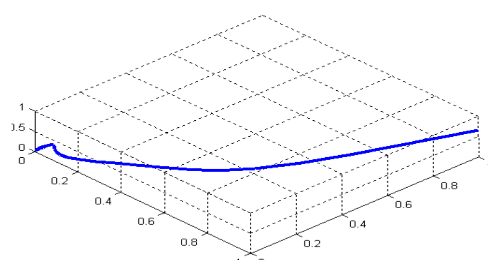

Fig.7. Side-grasp trajectory (after smoothing and rescale).

Trajectory Part	Top% (***C***)	Side% (***C***)	Top% (***O***)	Side% (***O***)
1	34	66	19.10	80.90
2	34	66	4.76	95.24
3	34	66	4.76	95.24
4	0.68	99.32	4.76	95.24
5	0.68	99.32	4.76	95.24
6	0.68	99.32	8.25	91.75
7	0.68	99.32	10.83	89.17
8	**1.68**	**98.32**	**8.00**	**92.00**

Tab.2. Classification using Curvatures (***C***) and Hand Orientation (***O***) for the trajectory shown in figure 7. It was classified as side grasp with 98.32% using curvatures and 92% using hand orientation (***O***). It is shown the probability of the trajectory being top or side grasp in each part of the trajectory.

Following the protocol (section III), two subjects have done some reach-to-grasp movements to test our application. Table 3 shows the results of the classification of 10 trials of side grasp using curvatures features, using hand orientation features and combining them using entropy as confidence level. The false negative values in the classification using curvatures features happened due the side-grasp trajectory are similar to the top-grasp. The classification using curvatures features when positive reached higher values than the classification using hand orientation features, but by other hand, using hand orientation features we did not have false negative values. Using the entropy $\boldsymbol{H}$, in these trials to reach an uncertainty measurement and give weights for each type of classification, we reached the following weights: $\boldsymbol{P(w_{curv})}$ = 0.61 and $\boldsymbol{P(w_{h\_or})}$= 0.38. Fig. 8 shows a comparison graphic among these 3 methods. The results show us that the result reached by the entropy belief is a kind of balance between both methods for the trials shown in table 3. The application was developed using the language C++. It was used a laptop HP Pavilion dv5000, AMD Turion 64, 2.0Ghz, 1Gb of RAM. The processing time for the segmentation process and classification are in real time.

Trial	1 – Classification using Curvatures	2 - Classification using Hand Orientation	3 – Entropy to combine both features
1	**98.32** %	**92.00** %	**95.85**%
2	**86.63** %	**76.93** %	**82.86**%
3	**21.67** %	**91.53** %	**48.81**%
4	**84.69** %	**61.12** %	**75.52**%
5	**5.78** %	**82.53** %	**67.41**%
6	**99.33** %	**51.22** %	**80.63**%
7	**99.68** %	**90.43** %	**96.08**%
8	**99.97** %	**91.53** %	**96.68**%
9	**88.98** %	**95.69** %	**91.58**%
10	**78.67** %	**55.98** %	**69.85**%

Tab.3. Result of 10 trials of Side-grasp. Two false Negative (less than 50%) on trial 3 and 5 using curvatures. The trials 4, 6 and 10 in hand orientation were considered as side grasp but with low probability, less than the threshold of 70%. It was reached just one false negative (trial 3) using entropy to combining both classifications. The trials 5 and 10 were considered side-grasp with low probability.

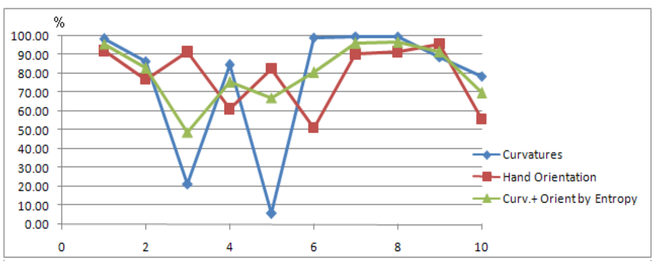

Fig.8. Comparison graphic. The 2 methods of classification using two different kind of features and the third one using weights reached by entropy.

To test the efficiency of the proposed method, we also have done some gesture recognition, i.e. we have learned more movements (bye-bye and circle), with 30 observation for both. Table 4 shows in the two first rows the classification of two circles movements and the last two rows show the classification of the by-bye movement. The results are similar to the top and side-grasp classification. In 10 trials we reached 1 false negative for both movements.

Trial	1 – Classification using Curvatures	2 - Classification using Hand Orientation	3 – Entropy to combine both features
1	**95.30** %	**80.65** %	**89.40** %
2	**86.53** %	**78.95** %	**83.54** %
3	**87.59** %	**76.52** %	**82.90** %
4	**89.84** %	**82.92** %	**86.91** %

Tab.4. Classification of Circle (2 first rows) and bye-bye (2 last rows) movements.

VI. Conclusion

We have developed an application for segmentation and classification of reach-to-grasp movements. Two different methods of segmentation in 3D space were used, by curvatures and by hand orientation. A dataset of reach-to-grasp movements were created to be used in a learning phase based on histogram techniques. Applying these two methods of segmentation we are able to classify the trajectories using Bayesian techniques. Entropy as uncertainty measurement was applied to reach a confidence level giving weights for both classifications for their fusion. The results have shown that using the weights reached from entropy for a joint classification has balanced the results, improving some classification when its probability is too low. The proposed method can also be used for gesture recognition tasks (e.g. for human-robot interaction), reaching similar results of reach-to-grasp movements classification.

References

[1] J. R. Flanagan, M. C. Bowman, and R. S. Johansson, “Control strategies in object manipulation tasks”. Current Opinion in Neurobiology, 16(6), 2006, pp.650–659.

[2] J. Faraway, M. Reed and J. Wang, "Modelling 3D trajectories using Bézier curves with application to hand motion". Journal of the Royal Statistical Society: Applied Statistics 56, 2007, pp. 571-585.

[3] J. Rett and J. Dias, “Human-robot interface with anticipatory characteristics based on Laban Movement Analysis and Bayesian models”. *Proceedings of the IEEE 10th International Conference on Rehabilitation Robotics* (ICORR), 2007, pp. 257-268.

[4] E.-J Holden and R. Owens, “Representing the finger-only topology for hand shape recognition*”, In Machine Graphics & Vision International Journal*, Volume 12 , Issue 2, 2003, pp. 187 – 202.

[5] D.R. Faria and J. Dias, "Hand Trajectory Segmentation and Classification Using Bayesian Techniques", *Workshop on Grasp and Task Learning by Imitation 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France - Sept, 22-26, 2008, pp. 44-49.

[6] Polhemus Liberty Electromagnetic Motion Tracking System. Available: http://www.polhemus.com/?page=Motion_Liberty.

[7] D. C. Knill and A. Pouget, “The Bayesian brain: the role of uncertainty in neural coding and computation,” *TRENDS in Neurosciences*, vol. 27, 2004, pp. 712–719.

[8] T. M. Cover and J. A. Thomas, “Elements of Information.Theory”, *Wiley & Sons*, 1991.

[9] T. Arbel and F. P. Ferrie, "Entropy-based Gaze Planning", Image and Vision Computing, Elsevier Science, vol. 19, issue 11, Sept. 2001, pp. 779-786.

[10] O. Ludwig Jr., U. Nunes, L. Schnitman and H. Lepikson, “Applications of information theory, genetic algorithms, and neural models to predict oil flow”, Communications in Nonlinear Science & Numerical Simulation, v. 14, p. 52-67, 2009.