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 computing (*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]. By observing human 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 types of features extraction. Using entropy as confidence level, weights for each classification model are assigned, allowing the combination of both types of features in a mixture model, acquiring a new classification model for results comparison. Using these techniques we developed an application to estimate and classify two possible types of human movements (reaching for top or side grasp). 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 the research and development of technologies that permit the introduction of the robots in our daily lives. 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. Many approaches have been proposed for predicting hand trajectories. Hand trajectory segmentation and classification are useful also in robotics field for imitation learning for human-robot interaction. Typically, the global hand's trajectory during a manipulation task can be segmented into 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, classification of reach-to-grasp movements when someone is performing the grasping is presented. 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 approach 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], a modelling approach for 3D hand trajectories in reaching movements is described. 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 towards human-robot interaction. The human actions are interpreted and mapped to the robot actions. The authors have contributed also with Laban Movement Analysis that assist identifying useful low-level features 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 characterizes the finger-only topology of the hand using cepstral coefficients. Techniques of speech signal processing were used for that. This 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 obtained undesired results as false negative, and classification of the trajectory with low probability.

III. Experimental Setup and Context

Polhemus Liberty tracker [6] is used to track the human hand trajectories. Five sensors were attached to a glove in order to grab 3D hand trajectories. Oner sensor was also attached to the object to have a prior knowledge of the object position. The setup for the experiments is comprised 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 the experimental area setup. Two reach-to-grasp movements were defined for this work: 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 wants to grip the object by its top just to displace it

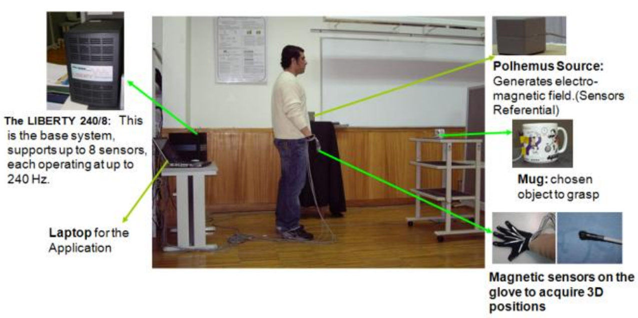

Fig.1. Experimental setup used for this work.

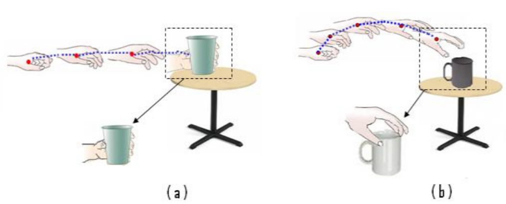

Fig.2. (a) – Side-grasp; (b) Top-grasp.

IV. Segmentation and Features Extraction

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, more than 20 seconds, then this movement will not be considered as the same of the learned one, because the features will not correspond to the learned ones. We are considering the spatial information instead. 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 resulting different scales which can harm the results. To solve this 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 by these phases. The division of the trajectory in 8 parts was chosen empirically. However a more sophisticated approach can be applied, e.g., segmenting the trajectories by events that characterize a manipulation tasks (e.g., reaching, lift, transport, and release phases). Thus, the movements can be initialized from different positions with different velocities without influencing the results.

To obtain the normalization of a trajectory, for all points of each axis (x, y, z), the following equation is applied 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* is the minimum value found; and *cur* is the current value of the trajectory 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 neighbours. Fig.3 shows an example of smoothing.

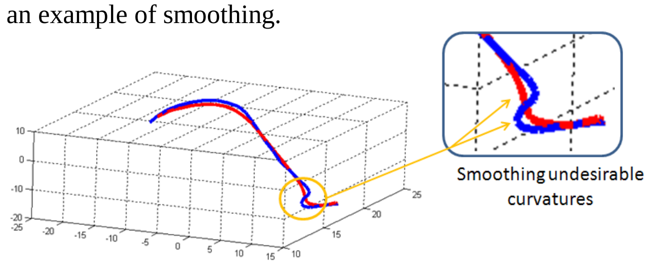

Fig3. Smoothed trajectory: Blue color – raw data; Red color – result.

B. Trajectory Curvature Features

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 have the curvature *right*. The same 2 vectors and their formed angles by the projection on (z, y) plane, we obtain the φ angle for tilt information. In 3D space we can make some combinations of the possible directions, e.g., we have *up* and *down* obtained by h, *left* and *right* obtained by θ and *further* and *closer* obtained by r. In this work we detect the following discretized features: *up*, *down*, *left*, *right*, *up-left*, *up-right*, *down-left*, *down-right* and *no-movement*, restricting other information, such as *closer* and *further*. We can obtain the height information (h) in a simpler way using the cylindrical coordinate system, calculating the difference between the z axis values using two points. In spherical coordinate system just the φ angle cannot inform us the height or diagonal movements, being also necessary verify the radius changes (r). 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 discretized features like *up-right*, *up-left*, *down-right* and *down-left*.

The curvature extraction is performed at each two points observed from the trajectory. The next steps show us how to obtain **(**r, θ, φ) in spherical coordinate system. Given two 3D points, for the first point we compute:

$$r_1 = \sqrt{x_1^2 + y_1^2 + z_1^2}\ , \tag{2}$$

$$\sin\varphi = \frac{\sqrt{x_1^2 + y_1^2}}{r_1}\ , \tag{3}$$

$$\cos\varphi = \frac{z_1}{r_1}\ , \tag{4}$$

$$\varphi_1 = \arctan\ 2(\sin\varphi, \cos\varphi)\ , \tag{5}$$

$$\cos\theta\ = \frac{x_1}{\sqrt{x_1^2 + y_1^2}}\ , \tag{6}$$

$$\sin\theta = \frac{y_1}{\sqrt{x_1^2 + y_1^2}}\ , \tag{7}$$

$$\theta_1 = \arctan\ 2(\sin\theta, \cos\theta)\ . \tag{8}$$

Then with the second vector acquired (i.e., the second 3D point) we follow the same steps presented by equations (2)-(8), obtaining then r_2, φ_2 and θ_2. After that, we obtain the 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 ignoring the equations (2) to (6), and we can rewrite equation (9) as follows:

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

To find a 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 obtined as follows:

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

where r_1 and r_2 are 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 could distinguish a type of grasp. After curvatures detection, the probability distribution of these features in each part of the trajectory is computed. For each feature is computed the probability distribution as follows:

$$P(c_i) = \frac{c_{i,k}}{\sum_q^k c_{i,q}}\ , \tag{16}$$

where $c_{i,k}$ represents the occurrence of a specific curvature in a specific hand displacement (trajectory part) and the denominator in (16) is the total of occurrences of all curvatures types found in each segment of the trajectory.

C. *Hand Orientation Features*

Using three sensors on three fingertips we can approximate the hand plane, allowing computing its orientation to find out if the hand is in top or side orientation (Fig.4). We have used the three fingers (index, middle and ring fingers) that usually remain parallel in the most part of hand configuration 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 frame of reference to know the hand orientation. At each 3 points in each part of the trajectory, the hand orientation is computed. In each part of the trajectory is found the occurrences of each type of hand orientation. Thus, probability distribution is given by:

$$P(o_i) = \frac{o_{i,k}}{\sum_q^k o_{i,q}}\ , \tag{17}$$

where $o_{i,k}$ represents the occurrences of hand orientation (side or top grasp) in a specific trajectory part and the denominator in (17) sums the total of occurrences of all hand orientation features found in a specific trajectory part.

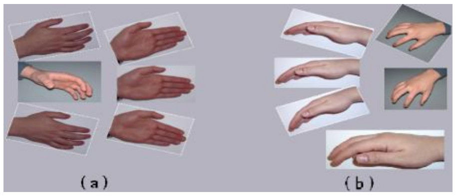

Fig.4. (a) Possible hands orientation for side-grasp; (b) Possible hands orientation for top-grasp.

D. *Results of the Segmentation Step*

For each observation of our dataset xml files that stores the characterization each the trajectory were created, i.e. segmentation information: features amount and their probability distribution. Two xml files for each trajectory was generated, one with curvatures and another with hand orientation information. This information is useful to for learning using histogram techniques that will be used in the classification step as likelihoods. 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 of table 1 is done for the segmentation by curvatures.

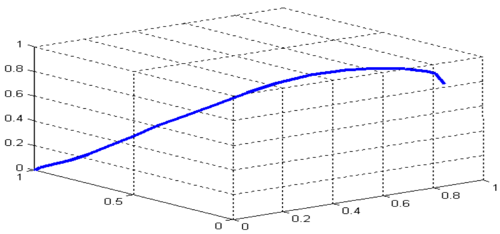

Fig.5. Top-grasp trajectory after smoothing and normalization.

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

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

V. Learning and Classification

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 [7]. Based on these studies, we follow Bayesian techniques to classify hand trajectories. The learning phase is based on histogram techniques given segmented features.

A. *Grasping Learning Table*

During the learning phase, all trajectories of our dataset are analyzed. 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. Since each trajectory generates a histogram with probability distributions for each class of feature, then we built learned tables computing an averaged histogram for top and side grasp trajectories. Fig.6 shows 2 examples of the Grasping Learning Tables obtained after analysing all trajectories of our dataset. Due to the learning being achieved through histogram techniques, some features might have zero probability, because they never have been observed. Whenever these features with zero probability occur in a classification step, the corresponding hypothesis will also receive 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:

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

where *F* represents types of the features (e.g. curvatures = 9, orientation = 2); n_i represents total of occurrence a specific feature type; *N* represents the total of all feature occurrences.

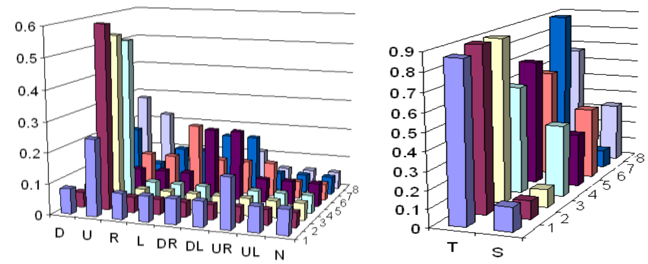

Fig.6. Left image represents the top-grasp curvatures learning table and right image top-grasp hand orientation learning table.

B. *Classification Model using Bayesian Technique*

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. We adopt a simple dynamic Bayesian network, using a Naïve Bayes classifier with probability transition (i.e., last posterior becomes the current prior – probabilistic loop). Assuming the trajectory that is being performed has size 1 (i.e., we know the trajectory size a priori since 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 the posterior is updated. 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*) as explained in section IV and V. Knowing P(*c* | *G i*) and the prior P(*G*) information of given trajectory represent a top or side grasp, 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 as follows:

$$P(G|c\ \ i) \propto P(c|G\ \ i)P(G). \tag{19}$$

We assume the same model of classification for hand orientation features, where *o* is a certain value of feature *O* (hand orientation for side and top grasp). Knowing P(*o* | , *i*) and the prior P(*G*) we apply Bayes rule as follows.

$$P(G|o\ \ i) \propto P(o|G\ \ i)P(G) \tag{20}$$

We formulate the equation in a recursive way. The posterior probability of a previous instant (trajectory part) becomes the prior for the next instant (next hand displacement). The rule for classification is based on the highest probability value, taking into consideration a certain confidence (e.g., probability of 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 Classifiers 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 entropy is used as confidence level to try to improve and obtain 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 models in a learning stage. For that, a confidence variable will be used as weight $w_i = \{w_1, \ldots, w_N\}$ for each classification model. The weight *w* will be used for a mixture model. For each model of classification we can compute the entropy of the posterior probabilities (outputs of the classifiers) as follows:

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

where *P(G| F D)* represents the posterior probability of each classification model. The variable *i* represents the index of each classification results, i.e., after *n* outputs of one model, we can apply (21). 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:

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

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 obtained by the entropy - this is known as a mixture model. Then the new model of classification is given by:

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

where $P(g_j \mid f\ i)$, represents the posterior of each Bayesian model (19) and (20). Each posterior is multiplied for its correspondent weight, thus, the new classifier is a weighted sum of both models, having this way a mixture model as an ensemble of classifiers.

D. Learning and Classification Results

Fig. 7 shows a side grasp trajectory performed by a subject and table 2 shows the answer of our approach along this trajectory, classifying it by using curvature and hand orientation features individually. The probability of the trajectory being top or side grasp is updated by Bayes rule in each part of the trajectory. The probability shown in table 2 represents the confidence of classification for each part of the trajectory. Comparing this case of Fig.7, we can see that both classifications obtained suitable performance, correctly classifying the trajectory. In this experiment the classification confidence at trajectory part by using curvatures was better than by hand orientation.

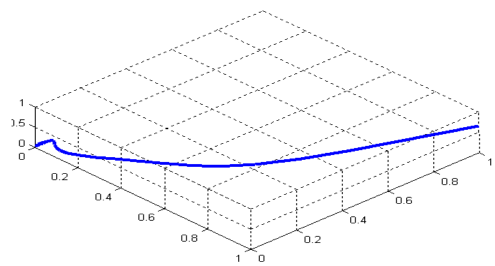

Fig.7. Side-grasp trajectory (after smoothing and rescale).

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*).

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**

Following the protocol (section III), two subjects have performed reach-to-grasp movements to test our approach. Table 3 shows the results of the classification of 10 trials of side grasp using curvatures features, using hand orientation features and combining them through the mixture model using entropy as confidence level. The false negative values in the classification using curvatures features happened due to the fact that the side-grasp trajectories are similar to the top-grasp. The classification using curvatures features when positive obtained higher confidence than the classification using hand orientation features, however, using hand orientation features, we did not obtain false negative values. Using the entropy *H* as an uncertainty measure to assign weights for each classification model, we obtained the following weights: $w_{curv} = 0.62$ and $w_{h\_or} = 0.38$. Fig. 8 shows a plot comparing these 3 models along 10 trials. The result obtained by the mixture model using entropy belief counterbalances the results of both previous models. We implemented our approach using the programming language C++. We run the tests in a laptop HP Pavilion dv5000, AMD Turion 64, 2.0Ghz, 1Gb of RAM. The processing time for the segmentation process and classification are on-the-fly.

Tab.3. Result of 10 trials of side-grasp trajectories. Two false negative (belief less than 50%) on trials 3 and 5 using curvatures. The trials 4, 6 and 10 using hand orientations were considered as side-grasp but with low probability, belief less than 70%. Just one false negative (trial 3) was obtained using entropy to combining both classification models. The trials 5 and 10 were considered side-grasp with low probability.

Trial	1 – Classification using Curvatures	2 - Classification using Hand Orientation	3 – Mixture model using 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%**

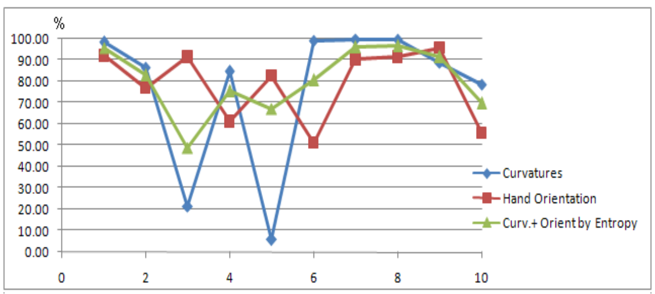

Fig.8. Comparison plot. Results of 10 trials for two classification models using two different types of features, and a third one using weights obtained by entropy to combine the previous model outputs.

To test the efficiency of the proposed method, we performed some gestures for recognition, i.e., we have learned more movements (bye-bye and circular movements), with 30 observations for both. Table 4 shows in the two first rows the classification of two circular movements and the last two rows show the classification of bye-bye movements. The results are similar to the top and side-grasp classification. In 10 trials we obtained 1 false negative for both movements.

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

Trial	1 – Classification using Curvatures	2 - Classification using Hand Orientation	3 – Mixture model using 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 %**

VI. Conclusion

We proposed a probabilistic approach for segmentation and classification of reach-to-grasp movements. Two different segmented features in 3D space were used, by curvatures (change in directions) and 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 were able to classify the trajectories using Bayesian techniques. Entropy was adopted as uncertainty measure to obtain a confidence level assigning weights for both classification models (one using only curvature features and another one using hand orientation features) for their fusion through a mixture model (weighted sum of posterior). The results show that using the weights obtained from entropy for a joint classification improved some classification results obtained from single classifiers when their confidence probability is too low, avoiding, thus, false negatives. The proposed approach can also be used for gesture recognition (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”, Comm. in Nonlinear Science & Numerical Simulation, v. 14, p. 52-67, 2009.