Real time graphical simulation for visual based pose estimation and self-calibrating of a humanoid robotic arm

Pedro Vicente
Instituto Sistemas e Robtica
Instituto Superior Tcnico
Lisboa, Portugal
Email: pvicente@isr.ist.utl.pt

Alexandre Bernardino
Instituto Sistemas e Robtica
Instituto Superior Tcnico
Lisboa, Portugal
Email: alex@isr.ist.utl.pt

***Abstract*—In this paper we propose a method for the online adaptation of a humanoid robot's arm kinematics, using its visual and proprioceptive sensors. The internal model of the robot (internal body schema) is represented by a graphical model/simulator based in the game engine Unity3d®.**

A typical reaching movement starts with a ballistic open-loop phase to bring the hand to the vicinity of the object. During this phase, as soon as the hand of the robot enters the field of view of one of its cameras, a vision based 3D hand pose estimation method feeds a particle filter that gradually adjusts the arm kinematics parameters. Our method makes use of a 3D CAD model of the robot hand and arm (geometry and texture) whose predicted position in the image is compared (with GPU enabled computation) at each time step with the cameras incoming information. When the hand gets close to the object, the kinematic errors have reduced significantly and a better control of grasping can eventually be achieved.

We have tested the method both in simulation and with the real robot and verify error decreases by a factor of 3 during a typical reaching time span.

***Index Terms*—Online adaptation, internal model learning, 3D model based tracking using GPU, reaching, humanoid robot.**

I. Introduction

Humans acquire body awareness through a process of sensorimotor development that starts in early infancy [1], or most likely in the womb already [2]. Such awareness is supported by a neural representation of the body that can be used to infer the limbs' position in space and guide motor behaviors: a body schema [3].

Considering more specifically the visual based control of reaching, a form of visual-proprioceptive calibration of the body might be performed by infants during the first months of life, as they spend a lot of time observing themselves while moving [4]. Until four months reaching movements seem to be just "ballistic", thus exploiting no visual feedback, as trajectory correction is absent [5], [6]. Then, from five months, vision is used to correct the hand position and orientation during the movement [7], with performance that improves during development [8]; however, after nine months this visual guidance almost disappears, as children become able to plan a proper hand trajectory at the movement onset

Fig. 1: The iCub humanoid robot performing a reaching task.

[9]. Bushnell claims that this decline of visually guided reaching is fundamental for the further cognitive development of the child, as it frees a big portion of visual attention that can be thus devoted to perceive and learn other aspects of the experienced situations [5]. In addition, these observations suggest that an internal model might have been learned through sensorimotor experience during the first months, and later exploited to improve the control. Indeed, a more general theory of human motor learning and control postulates that forward and inverse internal models of the limbs are learned and kept up-to-date in the cerebellum [10]. While inverse models are used to compute the muscle activations required to perform a desired movement, forward models can be used to simulate motor behaviors and to predict the sensori outcomes of specific movements [11]. These predictions are exploited in different ways: for example, they are combined with the actual sensory feedback through Bayesian integration to improve the estimation of the current state of the system [12].

Clearly, endowing artificial agents with similar capabilities is a major challenge for cognitive developmental robotics.

From a wider perspective, having an accurate and robust model of the controlled system is fundamental for any robotic application. In case of complex robots (e.g. humanoids) it is typically very difficult to obtain an accurate analytical model of the system, due to hard-to-model aspects (e.g. elasticity) and changes that might occur over time (e.g. unalignment of a

joint rotation axis); therefore, learning from data is becoming a more and more popular approach to equip robots with the necessary adaptation capabilities (see [13], [14] for recent surveys).

Our objective in this paper is to improve the accuracy of an analytical model of the robot using visual information and Bayesian estimation techniques. In particular, we consider a visual based reaching scenario using the iCub humanoid robot [15], depicted in Figure 1. Instead of learning an internal model from scratch, we exploit the *iKin* kinematic model of the robot [16] provided within the YARP/iCub software framework, and we adapt it online during reaching movements in order to cope with the modeling inaccuracies, allowing the robot to precisely reach for a desired position and orientation.

Our solution draws some inspiration from human development and learning, as: i) the internal model is updated online based on the visual feedback of the hand (something that infants supposedly do between four and eight months), and ii) the estimation of the hand pose results from the Bayesian integration of the sensory (visual) feedback and the prediction made by the internal model (a strategy that seems to characterize human perception as well [12]).

The rest of the paper is organized as follows. In Section II we report the related work in robotics and we highlight our contribution more specifically. Then in Section III we formalize the problem and we describe our robotic platform, while in Section IV we provide the details of our proposed solution. Finally, in Section VI we present the experimental results, and in Section VII we draw our conclusions and sketch future work.

II. Related Work

One of the key components of our approach relies on the detection and tracking of the robot hand and its comparison with predictions formed by the current internal model. Several approaches have been proposed to track human hands with visual information [17]. The problem is very complex due to the large number of degrees of freedom of a human hand. In [18] it is proposed an approach to track and estimate the 26-DOF of a human hand model. It combines skin color segmentation and edge maps to evaluate hypotheses that are optimized with Particle Swarm Optimization methods. The method is computationally expensive but with custom GPU implementations, quasi-real time performance can be achieved. To simplify the matching problem [19] proposes the user to wear colored gloves and develops a method based on efficient search of a database of examples. The previous methods attempt to estimate the pose of the hand in an arbitrary configuration. In our case, using the model of the robot hand and forward kinematics, we have a good approximation of the hand pose and appearance, so the problem is more constrained and we can rely on local search techniques. In [20] an algorithm was proposed for estimating the pose of a human hand with a specific gesture. Because the hand posture is known, the problem reduces to a 6D search, which is further reduced to 3D search on an orientation database, since translation can be computed analytically using image moments. Our observation model is similar to that one, but we use it in a particle filtering framework to update the robot's internal kinematics model. A few recent works investigated visual detection of a robotic hand using machine learning techniques [21], [22]. Both systems are marker-free and model-free, and they employ either Online Multiple Instance Learning [21] or Cartesian Genetic Programming [22] to learn from visual examples how to detect the robot hand inside an image. In [21] information coming from arm motor encoders and visual optic flow is integrated to autonomously label the training images, thus obtaining an unsupervised learning system. However, both solutions deal only with the hand position in the image, and not its 6D pose in task space. Different solutions have been proposed for the automatic calibration of the eye-head-arm-hand kinematic chain, that can allow accurate visual based reaching (some of them are reviewed in [13]). In [23], an upper humanoid torso is calibrated, including the sensor relative pose and the angle offsets and elasticity parameters of the kinematic chain. The method requires special markers in the wrist of the robot and operates offline with non-linear least squares optimization of data acquired during 5 minutes of robot specific movements. Online learning and adaptation of the kinematic model has been proposed as well [24], [25], but still using markers to visually detect the hand. Marker-free arm tracking has been investigated in [26], where RGB-D data from commercial depth sensors is used to correct the robot kinematics.

A. Our contribution

In terms of visual estimation of the hand pose, our approach is to combine information coming from different channels in a Bayesian way: vision, proprioception, a model of the hand appearance and an internal model of the robot kinematics. Concerning the robot calibration, our objective is to exploit the visual estimation of the hand pose to incrementally correct the kinematic model of the robot during the movements. For this purpose, we develop a particle filter to track the hand based on a likelihood metric that compares a prediction of the visual observation of the robot hand according to the current internal model, with the real images acquired by the RGB cameras in the eyes of our platform, the iCub robot. The internal model is based in a visual simulator and the comparisons are made in the GPU, increasing the speed of the algorithm.

III. Problem statement

A. The robotic platform

The iCub (see Figure 1) is a humanoid robot for research in embodied cognition, developed in the context of the EU project RobotCub (from 2005 to 2010) and subsequently adopted by more than 25 laboratories worldwide. It has 53 motors that move the head, arms and hands, waist, and legs; it has the average dimensions of a 3 years old child. It

is equipped with stereo vision (cameras in the eyeballs), proprioception (motor encoders), touch (artificial skin and tactile fingertips) and vestibular sensing (IMU on top of the head). The robot is equipped with a dynamic simulator [27] that has been used to generate the predictions of the observations that are compared to the real images acquired by the cameras. Although, this simulator is not able to generate more than 60 *frame per second*. In order to have a real-time application we developed, afterwords, a geometric model based on a game engine (Unity3D®) generating more than 500 *frames per second*. We combine, for the first time, the YARP platform (for communication purposes), the OpenCV library (for image processing) and OpenGL and CUDA programming (to use the capabilities of the GPU) within Unity3D®.

B. Joint Model

The online adaptation of the internal model consists in estimating joint offsets to the angles of the arm joints. The angular position of a joint is modeled by:

$$\theta = \theta_r + \beta + \eta \tag{1}$$

where θ is the value read by the encoders, θ_r the real value of the joint position, β is a systematic offset, and η a zero mean Gaussian noise with covariance $\mathbf{Q}$, $\eta \sim N(0, \mathbf{Q})$. This paper proposes to estimate β from visual feedback of the robot images, assuming negligible mechanical errors in the rotation axes alignment and link lengths. Also, we assume joint errors to be independent, meaning that the covariance matrix $\mathbf{Q}$ is diagonal. The iCub arm has 7 rotation joints each with a single degree of freedom. Our encoder readings ($\boldsymbol{\theta}$) are the seven joints of each arm. Three rotation joints in the shoulder, two in the elbow and two in the wrist defining: $\boldsymbol{\theta} = [\theta_0 ... \theta_6]$.

C. State Model

The offsets in Equation (1) define the state vector of an unobserved Markov process as $\mathbf{x} = [\beta_0\ \beta_1\ \beta_2\ \beta_3\ \beta_4\ \beta_5\ \beta_6]^T$ where β_i is the offset in joint i of one arm. We assume an initial distribution $p(\mathbf{x}_0)$ and a known state transition distribution $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$. To allow for small changes in $\mathbf{x}$ we introduce a state transition noise $\mathbf{w}$ and postulate the system state transition model as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{w} \tag{2}$$

Here $\mathbf{w} \sim N(0, \mathbf{K})$ is a zero mean Gaussian noise with a given covariance $\mathbf{K} = \sigma_s^2 \mathbf{I}_7$, where σ_s is the standard deviation.

D. Observation Model

At each time we have two sources of information, the encoder readings ($\boldsymbol{\theta}$) and the left and right camera images, $\mathbf{I}_L$ and $\mathbf{I}_R$, respectively. The observation vector will be a concatenation of the values of the two images defined as $\mathbf{y} = [\mathbf{I}_L\ \mathbf{I}_R]$ with a distribution of $p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_t)$. We defined belief as $p(\mathbf{x}_t|\mathbf{y}_{1:t}, \boldsymbol{\theta}_{1:t})$, which is the distribution of the state $\mathbf{x}$, at time t, conditioned by all past observation and encoder readings. Given a certain state vector $\mathbf{x}_t$ and some given encoder readings $\boldsymbol{\theta}_t$, we can form a prediction on the observed images ($\hat{\mathbf{I}}_L$ and $\hat{\mathbf{I}}_R$). Let $[\hat{\mathbf{I}}_L \hat{\mathbf{I}}_R] = f(\boldsymbol{\theta}, \mathbf{x})$ be a function that receives an angular position of the joints, here defined as the composition of $\boldsymbol{\theta}$ and $\mathbf{x}$, and creates two images ($\hat{\mathbf{I}}_L$ and $\hat{\mathbf{I}}_R$) of the corresponding visible pose on the left and right eye, respectively. We implement this function using an existing simulator that includes the forward kinematics of iCub and an image generation model. Note that f is highly nonlinear and two different sets of angles can generate the same image, or images with imperceptible differences. Redundancy in the joint angles may lead to different states ($\mathbf{x}$) with the same final pose. Therefore, the estimated $\mathbf{x}$ will be just one set of offsets that can explain our pose in the image.

To compute the image measurement probability $p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_t)$, we use the Hammoude metric [28] as a distance metric between the predicted and the real images. It is defined as:

$$d_{HMD}(y_1, y_2) = \frac{\#(\mathbf{R}_{y1} \cup \mathbf{R}_{y2}) - \#(\mathbf{R}_{y1} \cap \mathbf{R}_{y2})}{\#(\mathbf{R}_{y1} \cup \mathbf{R}_{y2})} \tag{3}$$

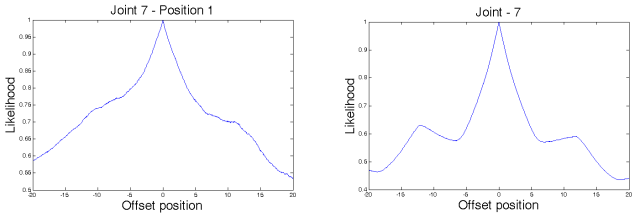
where $\mathbf{R}_{y1}$ represents the region of predicted silhouette of the hand and $\mathbf{R}_{y2}$ is the region of the real silhouette. In the simulated experiments this real silhouette is easy to obtain and in the real case a single colored background is used to simplify this step. Since, the Hammoude distance has a range between [0 1], the likelihood will be proportional to:

$$p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_t) \propto 1 - d_{HMD}(\mathbf{y}_t, f(\boldsymbol{\theta}, \mathbf{x})) \tag{4}$$

IV. Our approach

In our approach we will use a Particle Filter as defined in [29]. The choice of a particle filter is justified due to the non-linear observation model that leads to a multi-modal likelihood function (see Figure 2). To illustrate the multi-modal nature of the likelihood function we chose two reference positions and changed 0.1 degrees in joint 7 (wrist yaw). We can see that, for the two different initial positions, we get a different likelihood function with respect to the offsets in the joint. In the second case the likelihood has a multi-modal form. Under the Markov assumption we can compute recursively the *a posteriori* distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t}, \boldsymbol{\theta}_{1:t})$ using the previous estimation $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{1:t-1})$ and the observation model $p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_{1:t})$.

The particle filter has four stages: Prediction, Observation, Update and Re-sampling. In our case, the prediction step is quite simple due to the state transition equation (See Equation 2). In the observation stage we generate an image for the state $\mathbf{x}$ and compute the likelihood of each particle using the Hammoude distance. In the update stage, we use the likelihood in the previous step to re-weight each of the particles.



(a) Position 1 (b) Position 2

Fig. 2: Value of the observation likelihood function as a function of the 7th joint angle, for two different arm configuration. Note the different forms of the likelihood function for the two configuration, actually multi-modal in the second case.

The re-sampling step is probably the most important stage in the filter for its convergence. We use the systematic re-sampling method [30] that ensures the particles with a weight greater than 1/n to be always re-sampled, where n is the number of particles used.

After resampling we spread the particles using a normal distribution with zero mean and standard deviation σ_s, defined in section III-C.

A. Computing the state estimate

Although the state is represented at each time step as a distribution approximated by the particles, for evaluation purposes we must compute our best guess of the value of the state. For this purpose, we use a kernel density estimation (KDE) to smooth the weight of the particles according to the information of neighbor particles and choose the particle with the highest weight (W_i) as our state estimate:

$$W_i = L_i + \alpha * KDE_i; \tag{5}$$

where L_i is the particle likelihood, α is a smoothing parameter and KDE_i is the influence of the neighbors:

$$KDE(\mathbf{x}) = \frac{1}{n}\sum_{i=0}^{n} L_i * K(\mathbf{x} - \mathbf{x_i}) \tag{6}$$

where n is the filter particles number, the $\mathbf{x}-\mathbf{x_i}$ is the distance (in offsets space) between the particle ($\mathbf{x}$) we are smoothing , and a neighbor $\mathbf{x_i}$. K is a kernel specifying the influence of one particle in other based on their distance. We use a Gaussian Kernel in our experiments:

$$K(\mathbf{x_1}, \mathbf{x_2}) = \frac{1}{\sqrt{2\pi|\Sigma|}}\, e^{[-\frac{1}{2}(\mathbf{x_1}-\mathbf{x_2})^T\Sigma^{-1}(\mathbf{x_1}-\mathbf{x_2})]} \tag{7}$$

where Σ is the co-variance matrix and $|\Sigma|$ its determinant.

We assume that the joints are independent of each other, so Σ will be a diagonal matrix $\Sigma = \sigma_{KDE}^2 I_7$, where σ_{KDE} is the standard deviation in each joint, which we assume to be equal.This parameter defines if two particles are close or not. If we have a higher σ_{KDE} all particles will be "close" to each other. On the other hand if we have a small σ_{KDE} all particles will be "alone" in the world.

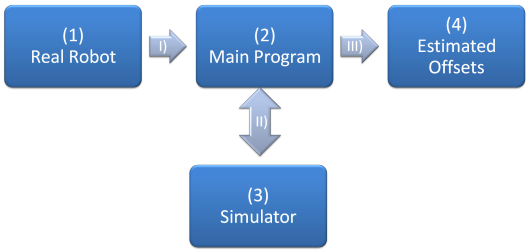


Fig. 3: General Work Flow of the our approach using a simulator to generate hypotheses

V. Implementation

A. General flowchart

A general flow chart of our approach can be seen in Fig. 3. This is the work flow of one iteration and four entities are present: Real Robot (1), Main Program (2), Simulator (3) and Estimated Offsets (4) and three communications procedures (I), II) and III)).

The encoder readings and cameras are sent from the Real Robot (1) to the main program (2) where the real image is segmented in order to be compared *a posteriori*. The main program commands (through II)) the simulator (3) to generate the poses related to each particle. The different images are compared and a likelihood score is sent back in order to estimated the hand pose (4) using a particle filter framework.

The proposed algorithm uses a new developed simulator within Unity3D® which works with CPU and GPU computation. In Fig. 4 the steps made from the robot to the final "Best estimated state" can be seen.

The comparisons between the generated images and the real one were made in GPU decreasing the computational time of the algorithm. The most considerable generation and evaluation of data is made in GPU with parallel processing. The CPU is used to communicate with the robot receiving the data from the cameras and encoders and to do minor calculations as particles' generation implementing our particle filter method.

We have integrated OpenGL, CUDA and OpenCV for the first time. This integration took place in order to compare the generated image with the received one from the real robot. We used a rendered texture in OpenGL where Unity3D renders the camera views and with CUDA programming we manage to copy this information to OpenCV GPU image class, providing methods to count non-zero pixels and to merge images. All the comparisons between the generated and real images as well as the likelihood computation were made in GPU increasing the speed of the data processing.

B. Error metrics

In order to evaluate the accuracy of our method we compute both the position and orientation errors between the

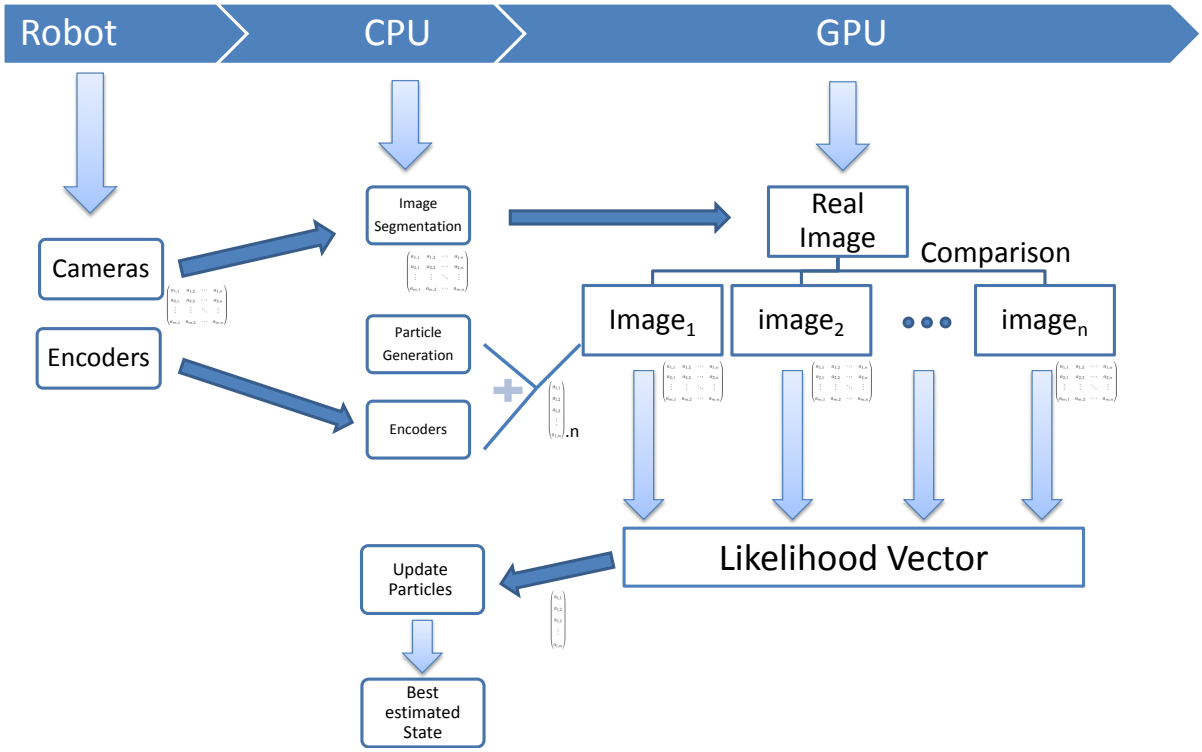


Fig. 4: Scheme showing the operations made in CPU, GPU and at the robot

real and estimated poses. The orientation error is defined as:

$$d(R_r, R_e) = \sqrt{\frac{\|logm(R_r^T R_e)\|^2}{2}} \frac{180}{\pi} \quad [^\circ] \tag{8}$$

where, R_r is the real rotation matrix from the eye frame to the end-effector and R_e is the estimated one. The principal matrix logarithm, $logm$, implements the usual distance on the group of rotations.

The position error between the two positions is computed by euclidean distance, $d(P_r, P_e)$: where P_r is the real position of the end-effector in the eye reference frame in 3D Cartesian space and P_e is the estimated one.

VI. Results

In this chapter we will show the results of various experiments. We have performed three types of tests. In the first test we use a simulated robot with artificial offsets in the arm joints and evaluate the convergence and accuracy of the filter. We consider observations taken both from one single camera and the stereo pair. In the second test we also use a simulated robot but this time we introduce also artificial offsets (calibration errors) in the head joints. Still, only the arm offsets are estimated by the filter. The head offsets are used to assess the robustness of the method to unmodelled sources of uncertainty. Finally in the third experiment we use the real robotic platform and estimate the offsets of the real arm. Despite no ground truth being available, we compare the predicted and real camera images to assess the level of precision attained.

In all experiments we used a pre-defined shape for the fingers. This posture correspond to a possible *hand preshape*, that is usually used in grasping contexts. This shape can be different, but for simplicity and comparison of the various experiments we keep it constant. Also the filter parameters are kept constant. We initiated the filter with a normal distribution $p(x_0)$ with zero mean and with standard deviation σ_0 for all the joints. The filter is composed by 200 particles, which is not a typical value in particle filtering framework, however it is an optimal value to balance real-time constraints and accuracy of the estimation in our setup. The other design parameters are $\alpha = 500$, $\sigma_{KDE} = 1.0$ and $\sigma_0 = 5.0$. To spread the particles after resampling we use σ_s decaying over time and with a lower bound. σ_s start in 3 degrees and decays 20% in every new iteration/frame with a lower limit of 0.10°, with this we achieve great precision with a reduced number of particles.

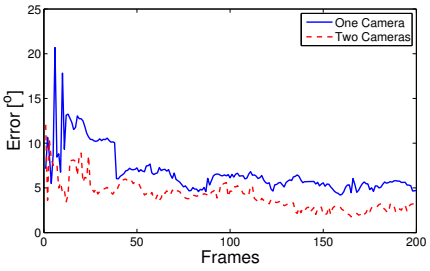
A. First experiment - Offsets in Arm

In this experiment we use the simulated robot with artificial offsets on every joint of the right arm chain. These offsets are the ground truth to evaluate the filter performance and have the values: $\beta_i = [5, 4, 3 - 2, +3, -7, +3]$.

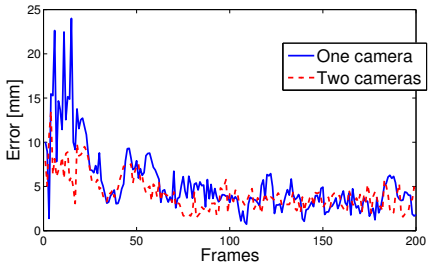
While the robot hand is in the field of view of the cameras, we iterate the particle filter to improve the state estimate. The experiments use a reaching like trajectory of the arm as illustrated in Fig. 5. The hand goes from the right bottom to the left top of the image. We perform 10 such experiments, with different initial and final poses. To quantitatively evaluate

(a) frame 0 (b) frame 019 (c) frame 086 (d) frame 139 (e) frame 200

Fig. 5: Example of a reaching task used in the first experiment.



(a) Orientation



(b) Position

Fig. 6: Orientation and position error over frames/time performing a reaching movement (Fig. 5) using one and two cameras. We can see the improvement in the orientation error with the use of two cameras

the accuracy of the filter we use the Cartesian position and orientation errors (See Eq. 8). We do not compare the state estimates directly with the ground truth joint offsets because there are redundancies in the kinematics that may lead to the estimation of offsets different from ground truth but that correspond to the same Cartesian poses. We show in Fig. 6 the evolution of the position and orientation errors of the hand base frame. The following observations can be made: (i) both errors reduce their magnitude about 3× during the trial; (ii) convergence is achieved in less than 100 frames; and (iii) the orientation error is lower in the stereo case but the position error does not change significantly with the addition of the stereo information. The mean and standard deviation of the errors for the 10 experiments are shown in Table I for the stereo case.

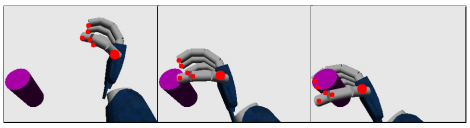
B. Second Experiment - Offsets in Arm and Head

For the second experiment we maintain the parameters of the filter and the errors in the right arm, but we added some errors in the head chain. The artificial offsets introduced in

	Mean	Standard Deviation
Angular Error[°]	4.5495	2.032
Position Error[mm]	3.3357	1.5163

TABLE I: Mean and Variance of the final orientation and position errors over 10 different experiments, for two cameras.

(a) frame 1 (b) frame 025 (c) frame 050



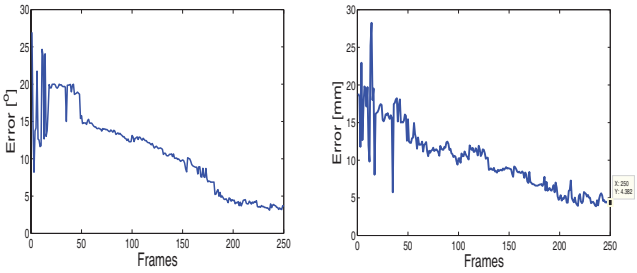
(d) frame 100 (e) frame 180 (f) frame 250

Fig. 7: Frames of 2nd Experiment - We have errors in right Arm and Head chains. The estimated positions of the fingers tips, in red, improved during the movement

the head were: -4°in the neck pitch, 6°in the roll, -2°in the yaw and -1°in the eyes tilt. The executed trajectory is illustrated in Fig. 7, where we can also observe the predicted position of the fingertips (dots in red) during the convergence of the filter. Notice that the position of the fingertips converge to the real values during the reaching time span. In Fig. 8 we plot the numerical values of the orientation and position errors along time, measured on the right eye reference frame. Note that the achieved accuracy is almost the same as when errors were only introduced in the arm chain (first experiment).

C. Third Experiment - Real Robot

In this section we will show an estimation of the offsets of the right arm with real data. The robot performed a set of arbitrary movements before approaching the target pose. The



(a) Angular error (b) Euclidean error

Fig. 8: Angular and Euclidean error over frames/time during reaching movement (Fig. 7). Despite the errors in the head we can estimate the pose of the hand and have almost the same final errors.

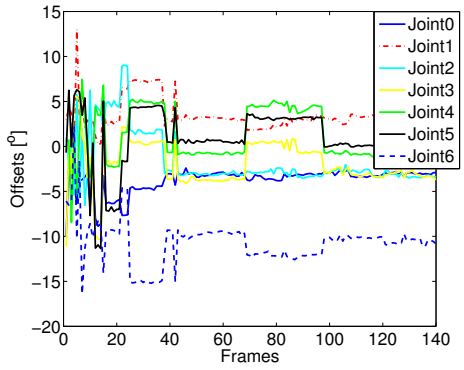
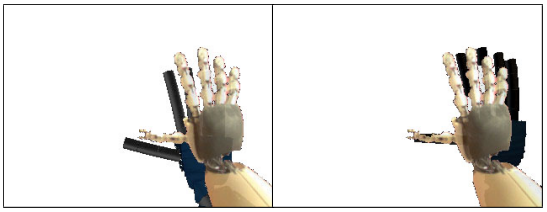


Fig. 9: Offsets estimation of the Real Robot Right Arm. The offsets converge to a value



(a) Without filter (b) With filter

Fig. 10: Estimation of the pose of the hand with the real robot. a) the simulated hand wihtout correction; b) the output estimation of our approach. Moreover, we see the similarities between the real and the simulated hand

temporal evolution of the offsets values are shown in Fig. 9. We can observe that after about 100 frames, the estimate reaches steady state. As mentioned previously, the accuracy is better evaluated using the Cartesian error, but since in this case ground truth is not available we measure the error in the image plane. This can be observed in Fig. 10, where the hand of the real robot and the predicted ones, with and without filter, are shown side by side. It's clear that the filtered pose is closer to the real one. Quantitatively, we have measured an average pixel difference between the fingertips of around 13 pixel with filtering and 25 pixel without, which at the average distance of the hand ($\sim$0.5m) corresponds more or less to 2cm and 4cm, respectively.

VII. Conclusions and Future Work

We presented a method to estimate the pose of a robotic hand, based on a particle filter framework. We have shown its convergence and the errors obtained during the motion. We used the encoder readings and the cameras of the iCub to compute a visual-proprioceptive calibration. In our simulated experiments, the position error of the hand was always below 5mm, which is a good error for grasping purposes. The orientation error was below 5°which is a reasonable value due to the errors present in each joint. In the real robot experiment, the offsets converged and despite the geometrical differences between the real robot and simulation, the error decreases in image pixels by a factor of 2.

As future work we plan to close the feedback loop, trying to do visual based hand control. The use of the Hammoude distance can be improved as well, the goal will be to smooth it using a distance transform method. We plan to implement an improved hand segmentation method capable of coping with more realistic and complex backgrounds.

References

[1] C. von Hofsten, "An action perspective on motor development," *Trends in Cognitive Sciences*, vol. 8, pp. 266–272, 2004.

[2] R. Joseph, "Fetal brain behavior and cognitive development," *Developmental Review*, vol. 20, pp. 81–98, 2000.

[3] G. Berlucchi and S. Aglioti, "The body in the brain: neural bases of corporeal awareness," *Trends in Neurosciences*, vol. 20, pp. 560–564, 1997.

[4] P. Rochat, "Self-perception and action in infancy," *Exp. Brain Res.*, vol. 123, pp. 102–109, 1998.

[5] E. Bushnell, "The decline of visually guided reaching during infancy," *Infant Behavior and Development*, vol. 8, pp. 139–155, 1985.

[6] C. V. Hofsten, "Structuring of early reaching movements: a longitudinal study," *Journal of Motor Behavior*, vol. 23, pp. 280–292, 1991.

[7] A. Mathew and M. Cook, "The control of reaching movements by young infants," *Child Development*, vol. 61, pp. 1238–1257, 1990.

[8] D. Ashmead, M. McCarty, L. Lucas, and M. Belvedere, "Visual guidance in infants' reaching toward suddenly displaced targets," *Child Development*, vol. 64, pp. 1111–1127, 1993.

[9] J. J. Lockman, D. H. Ashmead, and E. W. Bushnell, "The development of anticipatory hand orientation during infancy," *Journal of Experimental Child Psychology*, vol. 37, pp. 176–186, 1984.

[10] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends in Cognitive Sciences*, vol. 2, pp. 338–347, 1998.

[11] R. C. Miall and D. M. Wolpert, "Forward models for physiological motor control," *Neural Networks*, vol. 9, pp. 1265–1279, 1996.

[12] K. P. Körding and D. M. Wolpert, "Bayesian integration in sensorimotor learning," *Nature*, vol. 427, pp. 244–247, 2004.

[13] M. Hoffmann, H. Marques, A. Hernandez Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: A review," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 4, pp. 304–324, 2010.

[14] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, 2011.

[15] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The icub humanoid robot: an open-systems platform for research in cognitive development," *Neural Networks*, vol. 23, 2010.

[16] U. Pattacini, "Modular cartesian controllers for humanoid robots: Design and implementation on the icub," Ph.D. dissertation, Italian Institute of Technology, 2011.

[17] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, pp. 52–73, 2007.

[18] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Markerless and efficient 26-dof hand pose recovery," in *10th Asian Conference on Computer Vision*, Queenstown, New Zealand, November 2010.

[19] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM Transactions on Graphics*, vol. 28, no. 3, 2009.

[20] D. Periquito, J. Nascimento, A. Bernardino, and J. Sequeira, "Vision-based hand pose estimation: A mixed bottom-up and top-down approach," in *8th International Conference on Computer Vision Theory and Applications (VISAPP)*, Barcelona, Spain, February 2013.

[21] C. Ciliberto, F. Smeraldi, L. Natale, and G. Metta, "Online multiple instance learning applied to hand detection in a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1526–1532.

[22] J. Leitner, S. Harding, M. Frank, A. Forster, and J. Schmidhuber, "Humanoid learns to detect its own hands," in *IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 1411–1418.

[23] O. Birbach, B. Bäuml, and U. Frese, "Automatic and self-contained calibration of a multi-sensorial humanoid's upper body," in *Intl. Conf. on Robotics and Automation*, Saint Paul, Minnesota, USA, May 2012.

[24] S. Ulbrich, V. de Angulo, T. Asfour, C. Torras, and R. Dillmann, "Rapid learning of humanoid body schemas with kinematic bézier maps," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2009, pp. 431–438.

[25] L. Jamone, L. Natale, F. Nori, G. Metta, and G. Sandini, "Autonomous online learning of reaching behavior in a humanoid robot," *International Journal of Humanoid Robotics*, vol. 09, no. 03, p. 1250017, 2012.

[26] M. Klingensmith, T. Galluzzo, C. Dellin, M. Kazemi, J. A. Bagnell, and N. Pollard, "Closed-loop servoing using real-time markerless arm tracking," in *IEEE-RAS International Conference on Robotics and Automation (ICRA) - Humanoids Workshop*, 2013.

[27] V. Tikhanoff, P. Fitzpatrick, F. Nori, L. Natale, G. Metta, and A. Cangelosi, "The icub humanoid robot simulator," in *IROS Workshop on Robot Simulators*, 2008.

[28] A. Hammoude, "Computer-assisted endocardial border identification from a sequence of two-dimensional echocardiographic images," Ph.D. dissertation, 1988.

[29] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[30] J. D. Hol, T. B. Schon, and F. Gustafsson, "On Resampling Algorithms for Particle Filters," 2006.