3D Estimation and Fully Automated Learning of Eye-Hand Coordination in Humanoid Robots

S.R. Fanello[1,2,3], U. Pattacini[1], I. Gori[1], V. Tikhanoff[1],
M. Randazzo[1], A. Roncone[1], F. Odone[3] and G. Metta[1]

Abstract— **This work deals with the problem of 3D estimation and eye-hand calibration in humanoid robots. We first show how to implement a complete 3D stereo vision pipeline for humanoid robots, enabling online and real-time eyes calibration. We then introduce a new formulation for the problem of eye-hand coordination. Using the iCub humanoid robot, we developed a fully automatic procedure based on optimization techniques that does not require any human supervision. The end-effector of the humanoid robot is automatically detected in the stereo images, providing (theoretically) infinite training examples for learning the vision-kinematics mapping. We report exhaustive experiments using different machine learning techniques; we show that a mixture of linear transformations can achieve the highest accuracy in the smallest amount time, while ensuring real-time performances. We demonstrate the usefulness and the effectiveness of the proposed system in two typical robotic scenarios: (1) object grasping and tool use; (2) 3D scene reconstruction.**

I. Introduction

Recently new autonomous and humanoid robots have been designed and assembled; novel algorithms that allow them to perceive, feel, move, walk, have been proposed and successfully implemented. Still, integrating all or some of these capabilities has proved to be as demanding as developing them individually. In this regard, one of the most challenging problems is connecting visual perception to motion control, which is fundamental to make the robot correctly respond to what it sees. One of the most classical research topics in the field is estimating accurate 3D points of the real world from images acquired by a robot's visual system, and use this estimation to control the movement of the robot. This work aims at providing a robust 3D estimation and eye-hand calibration algorithm for the iCub [12] robot.

There has recently been a wide spread of cheap 3D sensors such as Kinect, which allow retrieving 3D information easily. In the context of humanoid robotics though, the goal is to achieve generality with eyes capable of fast movements, that can relocate quickly to points of interest in the world and can work indoors as well as outdoors. The Kinect for example is severely limited outdoors when the projected infrared pattern becomes virtually invisible. The humanoid robot considered in this work is iCub, which is equipped with a binocular camera system mimicking biological vision. It exploits the same principle humans rely on to retrieve depth information: the binocular disparity. Binocular disparity refers to the difference in image location of an object seen by the left and right eyes, resulting from the eyes' horizontal separation. The human brain uses binocular disparity to extract depth information from the two-dimensional retinal images in stereopsis. In computer vision, binocular disparity refers to the difference in coordinates of similar features within two stereo images [6]. Given the disparity between left and right image, a 3D point in the space can be accurately computed with respect to the camera reference system.

If the two cameras were perfectly aligned, the projection of a point in the two images would be separated only by a shift along the horizontal direction. This is extremely rare though (e.g. the cameras may be slightly rotated off-level), therefore a stereo vision calibration step, which consists in the estimation of the cameras' relative position, is required to compute disparity with a simple search along the horizontal direction. In humanoid robots this problem becomes harder, as robot cameras usually have many degrees of freedom. In our case, iCub's eyes can change their vergence, version and tilt, which impact on the relative rotation and translation between the two cameras. In this paper we first describe the full pipeline adopted to accurately estimate 3D points using the cameras of the iCub. We also integrate the iCub kinematics to achieve a better refinement of the camera positions. This is a very special case in the literature: it is not a standard stereocamera due to the degrees of freedom of the eyes, thus an on-line and real-time calibration is required.

The 3D point estimation in humanoid robotics is already a challenging problem. Furthermore, the obtained 3D points are computed with respect to the camera reference frame, therefore they are not useful for reaching or tracking as they are; we have to map the 3D points to a common reference system of the iCub, (name ROOT from now, see Fig. 1). Typically, the kinematic model of the robot is not precise enough to provide an accurate mapping between the position of a 3D point with respect to the cameras and the position of the same point with respect to the root frame. Indeed, humanoid robots are sophisticated machines and mechanical inaccuracies are common. For instance, it is not possible to ensure that the camera CCDs are mounted exactly in the modeled position, and even a small error produces large pixel shifts that increase non-linearly with the distance

This work was supported by the European FP7 ICT project No. 288382 (POETICON++), project No. 270273 (Xperience) and project No. 611909 (KoroiBot).

[1] S. R. Fanello, U. Pattacini, I. Gori, V. Tikhanoff, M. Randazzo, A. Roncone and G. Metta are with iCub Facility, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy {sean.fanello,ugo.pattacini, ilaria.gori, vadim.tikhanoff, marco.randazzo, alessandro.roncone, giorgio.metta} at iit.it

[2] S.R. Fanello is with Microsoft Research, 21 Station Road, CB1 2FB, Cambridge, UK.

[3] S.R. Fanello and F. Odone are with Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi, Università degli Studi di Genova, Via Opera Pia 13, 16145, Genova, Italy francesca.odone at unige.it

from the cameras. Another relevant aspect is represented by the unmodeled components such as tendons elasticity and friction that might have a not negligible influence in the way the robot computes its current configuration from the encoders values. These inaccuracies cause the vision-estimated 3D point with respect to the root frame to be slightly shifted. These kind of errors affect also the kinematic prediction of the 3D position of the end-effector with respect to the root frame, therefore the point perceived by vision is usually not the same acquired from the kinematics (see Fig. 1). As a consequence, the final offset between the 3D point with respect to the root frame computed through stereo vision and the same 3D point predicted by the kinematics could vary, on the iCub, from a few cm up to 6 cm. This is the typical eye-hand calibration problem. We developed a new approach to solve this problem, which is fully automatic and does not require any calibration pattern. Notably, the calibration procedure might need to be launched quite frequently, as the wear of mechanical parts and the re-tuning of relative encoders scale taking place at each robot startup are likely to produce slight modifications affecting the internal kinematic model of the platform. Therefore, our proposed method is also incremental (online) and very fast. We give a quantitative evaluation of such algorithm, comparing different methodologies. Finally, we present a number of applications that benefit from accurate calibration, allowing the robot to perform tasks that require accurate positioning.

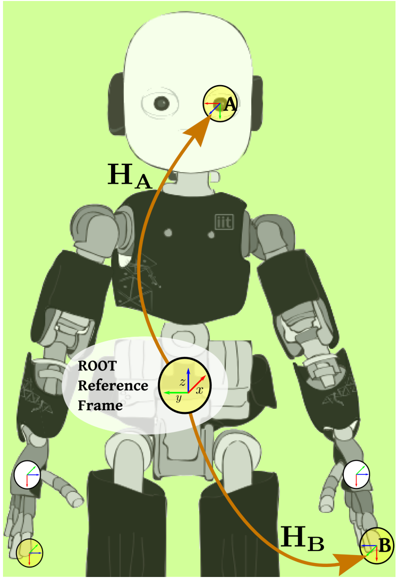

Fig. 1. A sketch of the iCub, the humanoid platform used to evaluate the proposed method. Notably, iCub's cameras have 6 degrees of freedom. The known kinematics is used to map the end effector position and 3D vision points to a common ROOT reference frame.

II. Related Work

Pioneer works in the field of eye-hand coordination are the work of Tsai et al. [23], [22] and Shiu et al. [17], which tackle the problem by considering first fitting rotations and then recovery the translation. Later approaches attempt to simplify the notation and compute closed form solutions [27], [2]. In [27] quaternions are used to estimate the rotational part, whereas [2] uses singular value decomposition. Works such as [28], [4] try to estimate both rotation and translation simultaneously. Recently (e.g. [18]), the objective became closing the loop among motions of multiple cameras and end-effector; in these calibration techniques, the assumption that the camera and the end-effector move using the same rigid transformation is dropped in favor of a wider generality. In [14] the authors reach very accurate results by means of a calibration pattern and a set of laser devices used in conjunction. The main limitation of all the mentioned methods turns out to be the use of calibration patterns in order to obtain a robust camera motion estimation. Sometimes the camera motion is even assumed to be known. The closest approaches to ours are [1], [16], which compute the camera position using a structure from motion pipeline, without resorting to calibration patterns. These techniques perform reasonably well; differently from our method though, they assume a complete knowledge of the end-effector motion (i.e. perfect kinematics model). In most of the proposed algorithms the camera is fixed on the hand, whereas our robot's cameras are mounted in the head. Furthermore, all the works mentioned so far do not deal with multi-degree-of-freedom cameras. Alternative methods are those based on machine learning, such as the Artificial Neural Network [8] and the Gaussian Processes [9]. These techniques have been lately exploited in order to estimate the spatial ego-sphere of a humanoid robot. However, they require two robotic platforms to generate ground-truth data, and the accuracy is remarkable only along one axis per time (e.g. the error on the X axis is around 0.8 cm); when considering all the axes simultaneously, the overall error is around 3.2 cm [9].

Compared to industrial robots, humanoids tend to be designed with resort to more sophisticated mechanical and software architectures, with the focus on adaption to the environment [12] (e.g. developmental robotics); in this sense the strong assumption of a perfect kinematic model cannot be satisfied. As a consequence, all the aforementioned methods are prone to failure if applied in the typical scenarios of humanoid robotics, since they build on the hypothesis that the end-effector motion is somehow provided. On the contrary, we take inspiration from neurobiological evidences [25], which show that the brain intrinsically incorporates a mapping between the eye and the hand, regardless the motion performed; interestingly, it has been demonstrated that such a spatial relation is updated incrementally, over time, while humans grow and/or use tools to modify their effectors. To replicate this mechanism, we first carry out a preliminary learning stage, where the eye-hand calibration is retrieved from the data using a supervised approach. Subsequently, at run time, this mapping is employed directly, without the need for demanding nonlinear solutions. Finally, similarly to the human brain, our map can be easily relearned online whenever new conditions arise to change the robot configuration. In summary, our main contributions are:

- An eye-hand calibration algorithm in the context of humanoid robotics, where several impairments (e.g. unmodeled elasticity and wear of the parts) prevent from accurately relying on the kinematic model. Therefore we do not have constraints regarding the camera/end-effector motion.
- Robustness against occlusion thus enabling generic camera configurations different from the in-hand cameras.
- Use of visual features to recover 3D structure without employing depth sensors. The entire system is based on standard RGB cameras and a six degree of freedom head (eye and neck).
- Use of visual features to match the end-effector frame to the camera reference system: the pipeline is fully automatic and does not require any human supervision.
- The learning strategy we propose models different end-effector and eyes configurations, and during the testing phase does not need to compute expensive non-linear solutions.

III. The System

The general formulation for eye-hand calibration starts from the following equation:

$$\mathbf{AX} = \mathbf{XB}, \qquad (1)$$

where $\mathbf{A} \in \mathrm{R}^{4\times 4}$ is a rototranslation matrix that represents the camera motion; it can be decomposed in a rotational part $\mathbf{R}_A \in \mathrm{R}^{3\times 3}$ and a translation vector $\mathbf{t}_A \in \mathrm{R}^3$. The matrix $\mathbf{B} \in \mathrm{R}^{4\times 4}$ is the end-effector motion, and $\mathbf{X} \in \mathrm{R}^{4\times 4}$ is the unknown transformation that relates the two reference systems. In general terms, the problem can be formulated as

$$\mathbf{AX} = \mathbf{ZB}, \qquad (2)$$

where the end-effector and the camera are related by two different transformation matrices.

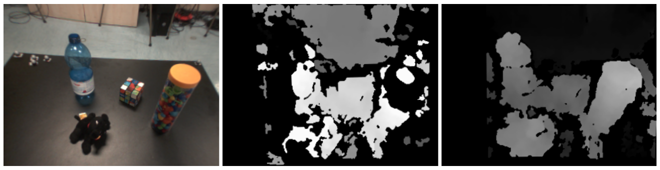

Fig. 2. Left: A typical scene seen by the robot. Middle: The depth map retrieved by the vision system based on pure kinematics calibration, producing poor results. Right: The improved depth map obtained after the calibration described in Section III-A.

State-of-the art methods usually make assumptions regarding the knowledge of $\mathbf{A}$ and $\mathbf{B}$, or they use calibration patterns. Our context is different though, since we try to solve the problem without imposing any assumption. In particular we exploit optimization techniques to model the function $f(\mathbf{A}) = \mathbf{B}$, avoiding the explicit computation of the matrix $\mathbf{X}$. Furthermore, since the matrices $\mathbf{A}$ and $\mathbf{B}$ are unknown, we estimate them, accounting for the noise.

Solving this problem enables any robotic platform to perform reaching and manipulation tasks with very high precision. In general, in order to obtain a very precise reaching given visual 3D points, the following conditions must hold:

1) 3D points with respect to the camera are very accurate.
2) The transformation from the camera to the end-effector is very accurate.

The first point can be addressed by using very precise depth sensors like Kinect. However our platform, iCub (see Fig. 1), is meant to be a cognitive platform, where the goal is to replicate human-like behaviors using the same capabilities. The iCub visual system is unusual in the literature related to vision: it mounts two cameras, but it has 6 degrees of freedom (3 for the eyes, 3 for the neck), thus it is not a stereocamera. If the kinematics between the two cameras were perfect, the extrinsic parameters of the cameras could be assumed known and 3D points could be computed accurately. However, in practice, using only the kinematics leads to imprecise image rectification and therefore poor disparity maps (see Fig. 2). On the other hand, standard off-line stereo calibrations are effective only when a particular and fixed eye configuration is imposed. Since our robot's cameras move continuously, a real-time estimation of the camera relative positions is required.

The transformation between the point perceived by the eyes and the same point perceived by the end-effector could be obtained again exploiting the known kinematics; however in general model imprecisions lead to very poor results. An example of this failure is shown in Fig. 4, where the expected end-effector position computed by the kinematics (green dot) is shifted with respect to the one computed by stereo vision (red dot).

In the following, we propose an algorithm to solve the two following sub problems:

- **3D Structure Estimation**. We show how to calibrate on-line and in real-time the iCub camera relative positions. This procedure generates 3D points with respect to the camera reference system (and therefore the matrix $\mathbf{A}$).
- **Eye-Hand Calibration**. We describe a method to collect 3D points from the camera reference system ($\mathbf{A}$) and the end-effector position ($\mathbf{B}$). Then we employ optimization techniques to learn the mapping between cameras and kinematics. We further show that a fast linear mapping is enough to obtain the highest results.

Solving both the sub problems in cascade allows retrieving final 3D points that, perceived by the stereo vision system, can be employed by the kinematics to execute reaching or more complex tasks.

A. 3D Structure Estimation

We consider couples of images acquired by the iCub stereo vision system. The main difficulty consists in the estimation of the two view geometry that allows for the rectification process of the images. After the rectification, any state-of-the-art disparity map algorithm can be used. In general a 3D point $\mathbf{X} = [\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{1}]$ is projected (up to a scale factor

s) into the image plane $\mathbf{x} = [\mathbf{u}, \mathbf{v}, \mathbf{1}]$ using a perspective transformation:

$$s\mathbf{x} = \mathbf{P}\mathbf{X}^{\mathrm{T}}, \tag{3}$$

where $\mathbf{P} \in \mathrm{R}^{3\times4}$ is known as the camera matrix and is described by intrinsic and extrinsic parameters:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}], \tag{4}$$

where $\mathbf{K} \in \mathrm{R}^{3\times3}$ is the matrix of the intrinsic parameters, the $[\mathbf{R}|\mathbf{t}]$ is the matrix of the extrinsic parameters represented by a rotation $\mathbf{R} \in \mathrm{R}^{\mathbf{3\times3}}$ and a translation vector $\mathbf{t} \in \mathrm{R}^{\mathbf{3\times1}}$. The intrinsic parameters need to be estimated only once; this can be done using standard calibration methods described in [6]. We now describe how to estimate extrinsic parameters effectively in real-time.

1) Undistorted Images: we first remove the image distortion, which entails lines to be deformed. Knowing the intrinsic parameters and the distortion coefficients, the images can be remapped to two new frames that do not contain distortion.

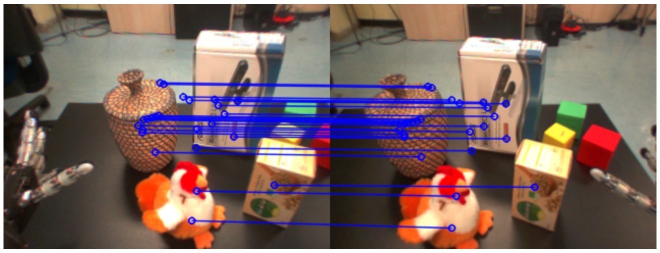

Fig. 3. An example of feature matching between left and right images. SIFT detectors and descriptors are used. We show only strong matches after the kinematic filtering and the RANSAC outlier rejection scheme.

2) Feature Matching & Fundamental Matrix: we now need to estimate the Fundamental Matrix $\mathbf{F} \in \mathrm{R}^{3\times3}$ that relates corresponding points in two images (i.e. matches). Given a match $\mathbf{x}, \mathbf{x}'$, it holds:

$$\mathbf{x}'^{T}\mathbf{F}\mathbf{x} = 0. \tag{5}$$

In this work we used SIFT detectors and descriptors [11] to compute keypoints from the undistorted images and perform the matching step (see Fig. 3). Being rank 2 and up to scale, the matrix $\mathbf{F}$ can be estimated using at least 7 points. Alternatively, it can be calculated employing the camera matrices $\mathbf{P}_L$ and $\mathbf{P}_R$ [6]. Among the set of the computed matches there will be some false positive, therefore we cannot use them at this stage. The known kinematics and the intrinsic parameters though, allow us to retrieve an initial estimation of the two camera matrices $\mathbf{P}_L$ and $\mathbf{P}_R$; however due to mechanic imprecisions, they need to be refined. Therefore we combine the two methods, first computing an estimated Fundamental Matrix $\mathbf{F}_K$ from the camera matrices; this will be used only to validate correspondences. Then we employ the good matches to calculate the real Fundamental Matrix $\mathbf{F}$. Given $i = 1, 2, 3$, $j = (i+1) \bmod 3$ and $k = (i+2) \bmod 3$, we define:

$$\mathbf{X}_i = \begin{pmatrix} P_L(j,1) & P_L(j,2) & P_L(j,3) & P_L(j,4) \\ P_L(k,1) & P_L(k,2) & P_L(k,3) & P_L(k,4) \end{pmatrix} \tag{6}$$

$$\mathbf{Y}_i = \begin{pmatrix} P_R(j,1) & P_R(j,2) & P_R(j,3) & P_R(j,4) \\ P_R(k,1) & P_R(k,2) & P_R(k,3) & P_R(k,4) \end{pmatrix} \tag{7}$$

We now compute the matrix $\mathbf{F}_K$, as follow:

$$\mathbf{F}_K = \begin{pmatrix} det([\mathbf{X}_1; \mathbf{Y}_1]) & det([\mathbf{X}_2; \mathbf{Y}_1]) & det([\mathbf{X}_3; \mathbf{Y}_1]) \\ det([\mathbf{X}_1; \mathbf{Y}_2]) & det([\mathbf{X}_2; \mathbf{Y}_2]) & det([\mathbf{X}_3; \mathbf{Y}_2]) \\ det([\mathbf{X}_1; \mathbf{Y}_3]) & det([\mathbf{X}_2; \mathbf{Y}_3]) & det([\mathbf{X}_3; \mathbf{Y}_3]) \end{pmatrix} \tag{8}$$

At this point we validate the correspondences, discarding the matches where $\mathbf{x}'^{T}\mathbf{F}_K\mathbf{x} > 0.01$. From the remaining correspondences we run the normalized 8-points algorithm [6] to compute the final Fundamental Matrix $\mathbf{F}$, using a RANSAC scheme for outliers rejection. At the end of this process we obtained a Fundamental Matrix that describes the epipolar geometry of the current eyes configuration.

3) Essential Matrix, Camera Geometry & Rectification: at this point we aim at estimating the Essential Matrix $\mathbf{E}$ [10], which relates right and left views considering calibrated contexts, i.e. the camera intrinsic parameters are known. Starting from the Fundamental Matrix, we compute:

$$\mathbf{E} = \mathbf{K}_R^T\mathbf{F}\mathbf{K}_L, \tag{9}$$

where $\mathbf{K}_R, \mathbf{K}_L$ are the 3×3 matrices of the intrinsic parameters. At the same time, the Essential Matrix can be expressed in terms of a rototranslation matrix between the two camera views [6]:

$$\mathbf{E} = \mathbf{R}[\mathbf{t}]_x, \tag{10}$$

where

$$[\mathbf{t}]_x = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}; \tag{11}$$

therefore, starting from the Essential Matrix we can finally estimate the extrinsic parameters of the cameras. We follow the classic procedure described in [6], retrieving four rototranslation matrices and disambiguating them using the *chierality* test. A further check is done considering the current kinematics of the robot: if the model does not differ too much from the retrieved matrices then the solution is accepted, otherwise it is discarded. We finally obtain a couple of camera matrices $\mathbf{P}_L = \mathbf{K}_L[\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}_R = \mathbf{K}_R[\mathbf{R}|\mathbf{t}]$, and we can perform the *rectification* process. We used the Bouguet's algorithm, which rotates the cameras so that they share the same X axis.

4) Structure Estimation: after the rectification procedure, we have reduced our current setting to a standard stereo camera. We remind that this entire procedure is repeated every time the robot moves its eyes. We are now ready to estimate the 3D structure of the scene using the disparity computation. In this work we used the procedure described in [7]. Thus, assuming known the disparity d between the left and right image for a given pixel (u, v), we can easily reproject it in the 3D space:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} (u - c_x)b/d \\ (v - c_y)b/d \\ bf/d \end{pmatrix}, \tag{12}$$

where b is the baseline of the two cameras (i.e. the norm of the translation vector $\mathbf{t}$), f the focal length and $(c_x, c_y)^T$ is the principal point of the stereo camera system. In Fig. 2,right we show an example of the obtained disparity maps.

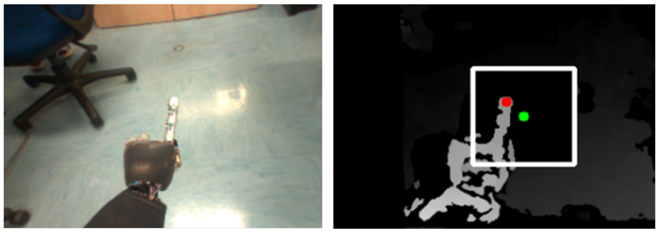

Fig. 4. RGB image and disparity map with the expected (green dot) and the real (red dot) end effector. The former position is retrieved directly using the known kinematics and projecting the 3D point into the image plane while the latter position is computed automatically by means of the depth map allowing for fast and easy segmentation of the fingertip.

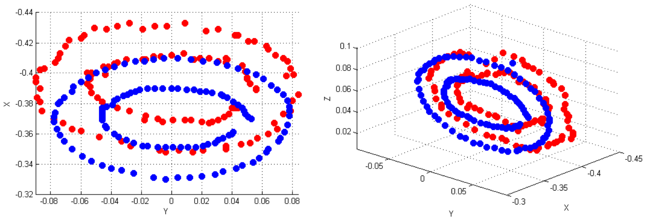

Fig. 5. Data generated for the hand-eye mapping. Red dots are 3D points retrieved from the stereo vision system after the fingertip detection. Blue dots are the end-effector positions detected via kinematics.

B. Eye-Hand Calibration

Provided with a perfect model describing how the vision system gets coupled with the kinematics, any 3D location of the end-effector can be flawlessly mapped in the camera image planes with no mismatch. In practice, as shown in Fig. 4, the expected position of the end-effector and the real one differ by some unknown offsets. In our case this shift could vary from 1 cm up to 6 cm, depending on the eyes configuration and the 3D point position. We tackle the calibration problem from a different perspective: instead of looking at the frames $\mathbf{A}$ and $\mathbf{B}$, we consider a set of N points belonging to the two different reference systems: $(\mathbf{X}_A^i, \mathbf{X}_B^i) \quad \forall i = 1, \dots N$, with $\mathbf{X}_A^i, \mathbf{X}_B^i \in \mathrm{R}^3$. Given this set of points, our goal is to learn the function $f(\mathbf{X}_A) = \mathbf{X}_B$. The main advantages with respect to other approaches are twofold: (1) when a large set of examples is provided, the function can be learned accurately and it can generalize to new positions in the 3D space; (2) we do not employ any calibration pattern for retrieving those training data but we rather rely on a fully automatic procedure. We consider the 3D end-effector position in homogeneous coordinates $\overline{\mathbf{X}}_B^i \in \mathrm{R}^4$ and we use the forward kinematics $\mathbf{H}_B \in \mathrm{R}^{4\times4}$ to map it to $\overline{\mathbf{X}}_{R,B}^i = \mathbf{H}_B \overline{\mathbf{X}}_B^i$, which gives the coordinates of the end-effector in the ROOT reference system (see Fig. 1). The detection of $\mathbf{X}_A^i$ is based on the stereo vision system of the iCub. We use the depth map to segment out all the background within a bounding box around the expected end-effector position $\mathbf{X}_B^i$. Then, considering the configuration of the hand showed in Fig. 4, we can detect the fingertip by retrieving the top-left point in the region of interest. This 2D point is reprojected in 3D space using Eq. 12, whose 3D projection is $\mathbf{X}_A^i$. The homogeneous point $\overline{\mathbf{X}}_A^i \in \mathrm{R}^4$ is then mapped to the ROOT frame $\overline{\mathbf{X}}_{R,A}^i = \mathbf{H}_A \overline{\mathbf{X}}_A^i$. Fig. 4 depicts an example of the described procedure: the end-effector point is reprojected in the image plane for visualization purposes (green dot); in red we show the detected fingertip. To automatically collect ground truth data, iCub moves its end-effector along multiple ellipsoidal paths with different centres, sizes and orientations in the 3D space. The data acquisition procedure is very simple, reliable and requires 2 minutes for the full calibration. During the acquisition, the robot actively tracks the end-effector expected point, in order to explore different eyes positions. Examples of the acquired data is depicted in Fig. 5: in blue we show the set of expected end-effector positions $\mathbf{X}_B$, while in red the 3D vision points $\mathbf{X}_A$ transformed with respect to the root frame are drawn. Given these two point clouds, our goal is to learn the offset between them. To this end, we propose to use a fast linear mapping $\mathbf{H} \in \mathrm{R}^{4\times4}$, such that $\overline{\mathbf{X}}_{R,B} = \mathbf{H}\overline{\mathbf{X}}_{R,A}$. We can solve the following minimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{H}} \frac{1}{N} \sum_i \|\overline{\mathbf{X}}_{R,B}^i - \mathbf{H}\overline{\mathbf{X}}_{R,A}^i\|^2 \\ &s.t. \mathbf{H} \in SE(3), \end{aligned} \tag{13}$$

where $SE(3)$ is the space of the admissible rototranslation matrices. We use the Ipopt solver [26], a public domain software package designed for large-scale nonlinear optimization. At run time, given a new homogeneous point from the vision system $\overline{\mathbf{X}}_A$, we compute the point with respect to the ROOT frame $\overline{\mathbf{X}}_R = \mathbf{H}\overline{\mathbf{X}}_{R,A}$. In practice, we have to explore the whole workspace of the robot and a single linear transformation may not generalize as expected. Therefore, we extend the model by introducing a mixture of transformations that we term *experts*, whose spatial competences can be easily retrieved from the corresponding training sets. As a result, any point $\overline{\mathbf{X}}_A$ is remapped to the ROOT reference system using the linear combination $\overline{\mathbf{X}}_R = \sum_i^K w_i \mathbf{H}_i \overline{\mathbf{X}}_{R,A}$, where each $\mathbf{H}_i$ is obtained by locally minimizing the Eq. 13. The weights w_i depend on the distance of the point $\overline{\mathbf{X}}_{R,A}$ from the training space centroid $\mathbf{c}_i \in \mathrm{R}^3$, taking also into account the covariance matrix $\mathbf{S}_i \in \mathrm{R}^{3\times3}$. The parameters $\mathbf{c}_i, \mathbf{S}_i$ describe the spatial occupation of the 3D points used to train the i-th expert in terms of the resulting minimum ellipsoid computed as in [21]. The weights are assigned through RBF functions computed with the Mahalanobis distance and then normalized as follows:

$$w_i = \frac{\exp(-(\mathbf{c}_i - \overline{\mathbf{X}}_{R,A})^{\mathrm{T}} \mathbf{S}_i^{-1} (\mathbf{c}_i - \overline{\mathbf{X}}_{R,A}))}{\sum_j^K \exp(-(\mathbf{c}_j - \overline{\mathbf{X}}_{R,A})^{\mathrm{T}} \mathbf{S}_j^{-1} (\mathbf{c}_j - \overline{\mathbf{X}}_{R,A}))}. \tag{14}$$

Remarkably, we tested in our experiments how the mixture of experts model runs fast, providing real-time performances, and accurately, achieving with only 4 experts very low reaching errors in the iCub work space.

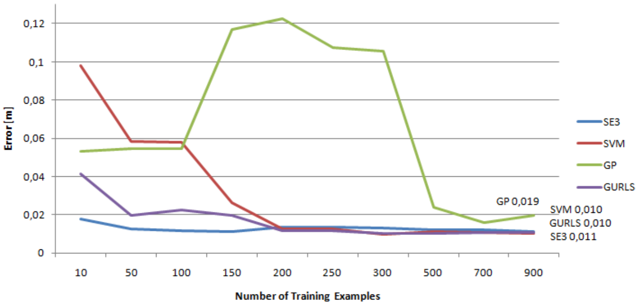

Fig. 6. Testing error [m] with respect to the training size. Learning methods require at least 150 points in order to reach the same accuracy of SE3. The lower bound error is 1 cm for all the methods.

IV. Evaluation

In this Section we evaluate the system with exhaustive quantitative experiments and comparisons.

A. Linear Experts vs. Machine Learning Techniques

Given the availability of (theoretically) infinite ground-truth data, also machine learning techniques seem to be good candidates to learn such mapping. In this Section we evaluate different methods to learn the offset between the vision and the kinematics. We compare 4 techniques: the first one is the mixture of linear transformations (SE3). We then analyze the following machine learning algorithms: Gaussian Processes (GP) [15], Regularized Least Squares (RLS) using the GURLS implementation of [19], and Support Vector Machine (SVM) [24].

During data-set acquisition, we let the iCub follow with its end effector (i.e. the index fingertip) some predefined ellipsoidal paths in the Cartesian space, which are sampled with 100 points; overall, we collect more than 2000 points. During the testing phase we generate new ellipsoidal paths. Empirically, we found that 4 experts suffice to cope with the iCub work space relevant for the envisaged tasks. In Fig. 5 we show examples of the end-effector positions acquired through kinematics (blue dots) and through the vision system (red dots). The hyper parameters of the learning methods have been estimated using standard cross-validation. In the first experiment we aim to evaluate the robustness of different methods with respect to the number of training data. In Fig. 6 we show the error with respect to number of training examples. Notably, SE3 reaches precisions up to 1 cm after a few examples (around 50, i.e. half ellipse). All the other methods require more examples, around 150, before reaching the same accuracy.

In practice, the eye-hand calibration may be often performed, since joints startup calibration entails mechanics changes. As a consequence, a fast procedure (i.e. a method that requires fewer training points) is preferred. In order to assess how much a method can generalize over unknown data given a relatively small number of training points, we train on a single ellipsoid (121 points) and test on other 19. Results for both training and test set are showed in Fig. 7. The initial error between vision and kinematics is 4 cm on the average. In this experiment, SE3 outperforms all the method achieving 1.1 cm of error. The second well-performing method is GURLS with 1.3 cm. Notably, on the training data, all the learning methods obtain higher results with respect to SE3 (around 0.2 cm of errors against 0.6 cm using SE3). This clearly shows that machine learning methods overfit the data, therefore they will perform better when the whole space is explored (i.e. infinite data).

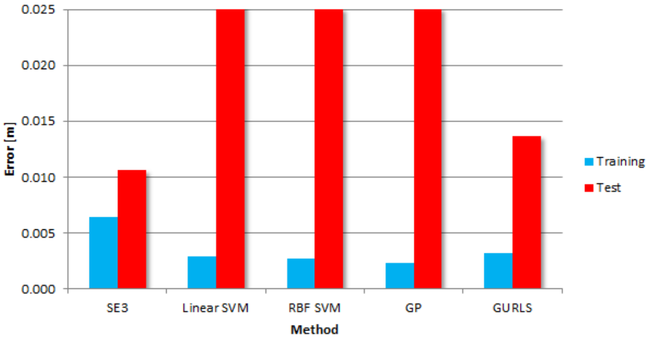

Fig. 7. Training and Testing error [m] for all the methods trained using only one ellipse and tested on other 19. SE3 turns to be the best candidate as it requires fewer examples to achieve same accuracy.

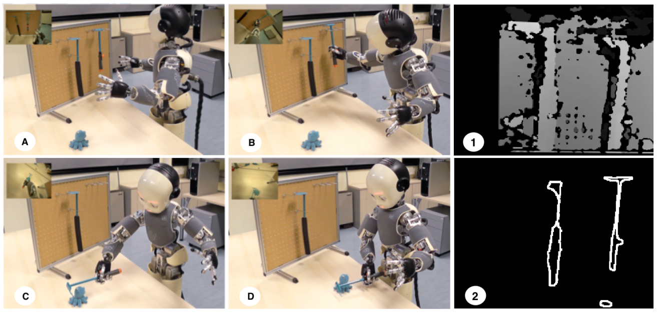

Fig. 8. Tool use experiment: (A): The instruction to grasp the object is given to the iCub, which reasons on which tool to use (B): reaches for the appropriate tool. (C): reaches for the object with the tool. (D): pulls the object towards itself for grasping. (1-2): disparity map and segmentation.

B. Reaching Performances

To evaluate the capability of the iCub to reach for 3D points in real scenarios we consider the setup of Fig. 9, using the Vicon Motion Capture System[1]. The Vicon is a state-of-the-art infrared marker-tracking system that offers millimeter resolution of 3D spatial displacements. A schematics of the setup used is depicted in Fig. 9 (right): the iCub stands in front of a table with a target object lying on top; we accommodated one Vicon marker on the robot index fingertip and a second marker on top of the target which is placed over the table in the 4 different positions P_0, P_1, P_2, P_3 on the xy plane. We also considered two different table heights

[1] Website: www.vicon.com

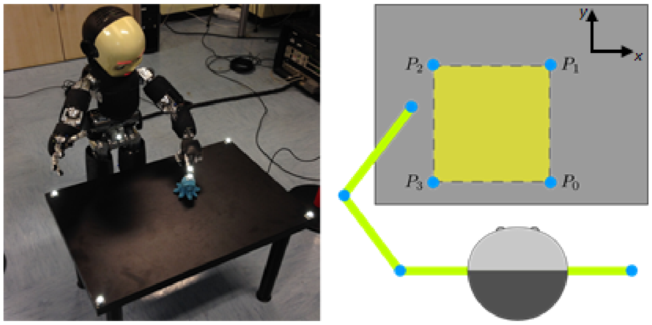

Fig. 9. Left: The Vicon system setup used to evaluate the reaching performances. Right: A sketch depicting the relative positions of target markers P_i with respect to the robot.

z_0 and z_1 ($|z_1 - z_0|$=10 cm) in order to better explore the operational space, resulting in 8 points in total per session. For each point, the iCub performs 5 reaching actions. The goal of the experiment is to verify the precision as well as the repeatability of such reaching. Table I reports the results without the calibration (i.e. using 3D vision and inverse kinematics [13] only), and with the proposed eye-hand calibration procedure. For each modality we collect the standard deviation σ_{eff} of the final 3D points reached by the end-effector (to evaluate the movement repeatability); we also collect the mean and the standard deviation of the norm of the error e between the former attained 3D locations and the corresponding target markers P_i (to give an estimation of the movement precision). Importantly, results illustrate that the pipeline composed of the cascade of stereo vision and inverse kinematics do generate very reliable and repeatable movements (as testified by very low values of σ_{eff}) and that the proposed eye-hand calibration is capable of improving the overall reaching accuracy by significantly reducing the error e of 4.2 cm on average. Finally, it is worth noting how the mean errors recorded while using the calibration are in accordance with the predictions of Section IV-A.

TABLE I

Results of the reaching evaluation using the Vicon system.

		NO CALIBRATION		CALIBRATION	
		σ_{eff} [cm]	$\|e\|$ [cm]	σ_{eff} [cm]	$\|e\|$ [cm]
	P_0	0.27	4.29 ± 0.56	0.32	0.74 ± 0.39
	P_1	0.01	7.83 ± 0.05	0.16	1.69 ± 0.08
	P_2	0.33	6.84 ± 0.52	0.35	1.74 ± 0.15
90 HEIGHT 1	P_3	0.49	5.89 ± 0.82	0.65	0.96 ± 0.39
	P_0	0.13	3.69 ± 0.68	0.11	0.70 ± 0.10
	P_1	0.51	5.79 ± 0.50	0.05	1.22 ± 0.07
	P_2	0.20	5.67 ± 0.20	0.16	0.93 ± 0.31
90 HEIGHT 2	P_3	0.16	2.97 ± 0.02	0.40	1.38 ± 0.06

V. Experiments

In order to qualitatively demonstrate that the proposed procedure actually improves how humanoids can tackle simple and more complex tasks, we present two real applications implemented on the iCub, which entail the coordination of perception and motion capabilities.

A. Power Grasp & Tool Use

The first task on which we assessed our system is a power grasp application. The procedure described in [5] is applied on several objects. In this previous work, we needed to fix an eye configuration, and then estimate empirically the offset between vision and kinematics. In particular, we performed a few grasps in fixed conditions, and then we manually set the offset between the 3D point perceived by the stereo vision and the one predicted by the kinematics, in order to obtain reliable grasps. Clearly this procedure was not automatic, and it had to be performed every time it was necessary to change the robot's eyes configuration. Here, we first let the robot execute 20 grasps on 4 objects (the ones showed in Fig. 2 , same used in [5]), and we evaluate the goodness of our system counting the number of successful grasps over the number of total grasps. First, we use the 3D points computed by the stereo vision system after the calibration procedure (see Section III-A), without applying the eye-hand coordination step, obtaining 23% of accuracy. We then report our previous results [5] obtained by manually setting the offset between vision and kinematics: 91.25% of success rate. We finally let the robot perform the same grasp actions using the procedure proposed in Section III-B achieving the remarkable precision of 97.5%. Not only the eye-hand calibration improves the overall results, but in addition it does not require any manual offset among different objects and positions in the space.

We then put to test our system in a second scenario that extends our previous work on the use of tools for exploring affordances [20]. The iCub was able to explore hand held tools, learn how to use them and finally employ the learned skill in order to accomplish his task. We placed the tools on a rack within the reach of the iCub (see the setup in Fig. 8) and used the proposed calibration procedure to successfully reach and grasp the required tool. To demonstrate that the system is robust and precise enough we performed 20 reach and grasp actions on tools placed on the rack arranged in four different orientations with respect to the gravity direction: 0, -45 and 45 degrees obtaining $95\%, 90\%, 90\%$ of successful grasps respectively.

B. 3D Scene Reconstruction

The estimated depth map can be also exploited to reconstruct the 3D space surrounding the robot, integrating data that belong to different views of the environment into a single 3D scene. This is a typical application also for mobile robotics, in which 3D cameras or laser systems are commonly employed to perform simultaneous localization and mapping tasks. In our specific case we attempt to evaluate the quality of our depth data by reconstructing scenes, such as the workspace in front of the robot. For our experiment we used the CCNY Visual Odometry package that is publicly available. The algorithm [3] works by tracking relevant RGBD features between camera frames and aligning them against a unique 3D model of the world, obtaining an estimated camera pose. This pose is then used to expand the model of the world, by inserting new landmarks for each feature which is not associated to the model set. An example of a 3D scene reconstructed using our robot's cameras is shown in Fig. 10. It is worth noting that the reconstruction exhibits many non valid regions, which correspond to the areas where the depth map algorithm [7] fails, usually because of uniform color and lack of features. Nevertheless, the overall performance is good and data acquired from different views are successfully merged into a single coherent spatial model.

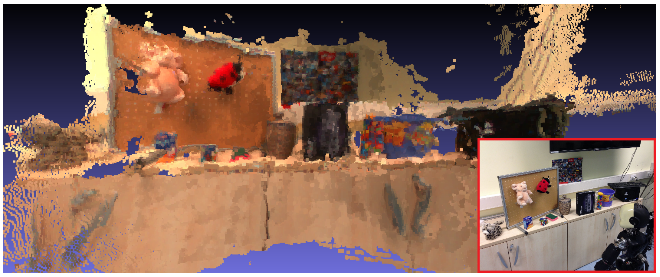

Fig. 10. 3D scene reconstruction obtained by moving the iCub head.

VI. Discussion

In this paper we tackled the problem of learning the vision-kinematics mapping in humanoid robots. We showed the importance of having a reliable coordination between the vision system and end effector for high level applications. We proposed a fully automated procedure for the eye-hand calibration problem. The method is based on the stereo vision system of the iCub robot, it takes into account the mechanics inaccuracies of the robot, it works for non in-hand camera setups and it does not require any supervision. Furthermore, the procedure is very fast and it can be performed on the fly. We also showed different applications which benefit from the proposed method. Future work includes the evaluation of different depth map algorithms; indeed we noticed that the implementation of [7] available in OpenCV suffers from illumination changes and it represents a bottleneck in terms of speed and accuracy. Our aim is also to extend the method to other applications such as navigation and localization in unknown environments, 3D feature extraction and body pose estimation.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Dr. Ambra Bisio for her invaluable help and expertise provided during the Vicon experiments.

References

[1] N. Andreff, R. Horaud, and B. Espiau. Robot hand-eye calibration using structure-from-motion. *IJRR*, 2000.

[2] J.C. K. Chou and M. Kamel. Finding the position and orientation of a sensor on a robot manipulator using quaternions. *IJRR*, 1991.

[3] I. Dryanovski, R. Valenti, and J. Xiao. Fast visual odometry and mapping from rgb-d data. In *ICRA*, 2013.

[4] I. Fassi and G. Legnani. Hand to sensor calibration: A geometrical interpretation of the matrix equation ax=xb. *JRS*, 2005.

[5] I. Gori, U. Pattacini, V. Tikhanoff, and G. Metta. Ranking the good points: a comprehensive method for humanoid robots to grasp unknown objects. *IEEE International Conference on Advanced Robotics (ICAR)*, 2013.

[6] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[7] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[8] J. Leitner, S. Harding, M. Frank, A. Förster, and J. Schmidhuber. Learning spatial object localization from vision on a humanoid robot. *International Journal of Advanced Robotic Systems*, 2012.

[9] J. Leitner, S. Harding, M. Frank, A. Förster, and J. Schmidhuber. Artificial neural networks for spatial perception: Towards visual object localisation in humanoid robots. *IJCNN*, 2013.

[10] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[12] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori. The icub humanoid robot: an open platform for research in embodied cognition. In *Workshop on Performance Metrics for Intelligent Systems*, 2008.

[13] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini. An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots. In *IROS*, 2010.

[14] V. Pradeep, K. Konolige, and E. Berger. *Calibrating a Multi-arm Multi-sensor Robot: A Bundle Adjustment Approach*. Springer, 2014.

[15] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT-Press, 2006.

[16] Jochen Schmidt, Florian Vogt, and Heinrich Niemann. Calibration-free hand-eye calibration: A structure-from-motion approach. In *Conference on Pattern Recognition*, 2005.

[17] Y.C. Shiu and S. Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax=xb. *Robotics and Automation*, 1989.

[18] S. H Strobl and G. Hirzinger. Optimal hand-eye calibration. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4647–4653. IEEE, 2006.

[19] A. Tacchetti, P. Mallapragada, M. Santoro, and L. Rosasco. Gurls: a toolbox for large scale multiclass learning. In *NIPS workshop on parallel and large-scale machine learning*, 2011.

[20] V. Tikhanoff, U. Pattacini, L. Natale, and G Metta. Exploring affordances and tool use on the icub. In *HUMANOIDS*, 2013.

[21] M.J. Todd and E. A. Yildirim. On khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathemathics*, 2007.

[22] Roger Y. Tsai and Reimer K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand-eye calibration. In *International Symposium on Robotics Research*, 1988.

[23] R.Y. Tsai and R.K. Lenz. Real time versatile robotics hand/eye calibration using 3d machine vision. In *Robotics and Automation*, 1988.

[24] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., 1998.

[25] R. Volcic, C. Fantoni, C. Caudek, J.A. Assad, and F. Domini. Visuomotor adaptation changes stereoscopic depth prediction and tactile discrimination. *Journal of Neuroscience*, 2013.

[26] A. Wächter and L.T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 2006.

[27] H. Zhuang, Z.S. Roth, Y.C. Shiu, and S. Ahmad. Comments on "calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax=xb". *Robotics and Automation*, 1991.

[28] H. Zuang and Y.C. Shiu. A noise-tolerant algorithm for robotic hand-eye calibration with or without sensor orientation measurement. *Systems, Man and Cybernetics*, 1993.