2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Weakly Supervised Strategies for Natural Object Recognition in Robotics

Sean Ryan Fanello[1], Carlo Ciliberto[1], Lorenzo Natale[1] and Giorgio Metta[1]

Abstract— **The paper aims at building a computer vision system for automatic image labeling in robotics scenarios. We show that the weak supervision provided by a human demonstrator, through the exploitation of the independent motion, enables a realistic Human-Robot Interaction (HRI) and achieves an automatic image labeling. We start by reviewing the underlying principles of our previous method for egomotion compensation [1], then we extend our approach removing the dependency on a known kinematics in order to provide a general method for a wide range of devices. From sparse salient features we predict the egomotion of the camera through a heteroscedastic learning method. Subsequently we use an object recognition framework for testing the automatic image labeling process: we rely on the State of the Art method from Yang et al. [2], employing local features augmented through a sparse coding stage and classified with linear SVMs. The application has been implemented and validated on the iCub humanoid robot and experiments are presented to show the effectiveness of the proposed approach. The contribution of the paper is twofold: first we overcome the dependency on the kinematics in the independent motion detection method, secondly we present a practical application for automatic image labeling through a natural HRI.**

I. Introduction

In recent years there has been a spread of robotics technologies that start to enable the real human-robot interaction. It is natural to believe that in the next future this interaction will keep increasing both in complexity and variety, until it will become necessary for the artificial system to be endowed with a physical embodiment in order to carry out tasks requested by the human.

In view of this futuristic scenario, the goal of Robotics is to improve the capabilities of artificial agents so that they can actually interact with other entities in a safe and meaningful way. Indeed, nowadays robots play already an important role in our life (take for example the industrial context), but cannot currently share physical space with humans. In this paper we identify such incompatibility to originate from the perceptual domain. Robots are unable to correctly perceive complex scenes and therefore robotics applications are validated in structured environments in which such task is greatly simplified.

This work is an attempt to fill the gap between the traditional Computer Vision field and the Vision for Robotics. Indeed acquiring a proper dataset requires a lot of effort and usually a strong supervision is needed when multiple views of the same object/scene are desired. This means that researchers always focus on well-known datasets such as Caltech-101 [3] or PASCAL Visual Object Classes [4], but often they are not well-suited for robotics scenarios. We show that with a weak supervision provided by humans, we are able to overcome one of the major limitations in the Computer Vision field: the automatic labeling of images.

This work is validated on the iCub humanoid robot [5]. The particular visual system at disposal involves a pair of stereo cameras mounted on a 6 Degrees of Freedom (DoFs) controllable head mounted on a 3 DoFs torso. The experimental discussion is focused on the specific robotic platform, however, our application it is presented in a generic form to manifest clearly its applicability to a wider range of architectures.

The rest of the paper is organized as follows: in Section II we introduce related works and explain our contribution. In Section III, IV, V we describe the main ingredients of the application: we first present our independent motion algorithm with the new improvements, then we briefly review the active tracking and the object recognition pipeline. Experiments, the application and results are presented in Section VI and finally Section VII concludes the paper with remarks and future work.

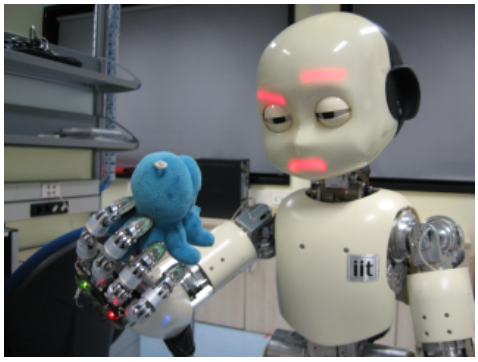

Fig. 1. The iCub humanoid robot used as experimental platform for the proposed application.

II. Contribution and Related Work

The main contribution of this paper is a computer vision application in robotics setting aiming at showing that a weak supervision can really improve the HRI, providing an efficient and natural way for image labeling in robotics scenarios. A human demonstrator asks the robot to actively track

Research supported by the European FP7 ICT project No. 270490 (EFAA) and project No. 270273 (Xperience).

S.R. Fanello, C. Ciliberto, L. Natale and G. Metta are with the iCub Facility, Istituto Italiano di Tecnologia, Genova, Italy {sean.fanello, carlo.ciliberto, lorenzo.natale, giorgio.metta} at iit.it

a moving unknown object; the robot builds a model of such object while the operator shows the object from different views. In the later recognition stage, the robot is asked not only to track the moving object, but also to recognize it. Notably, the recognition in this setting is really challenging due to the presence of a varying cluttered background and due to the demonstrator's arm that contributes to every image representation, increasing the probability of misclassification. We show that, using state of the art techniques, it is possible to implement such application obtaining high recognition performances also in this realistic task. The elements of such application are the detection of independent motion in the scene and an object recognition framework.

Solving the first problem is crucial in order to make a robot able to track an unknown object and extract proper Regions of Interest (ROIs) for the learning stage. Although a wide number of trackers have been proposed in the literature, most of them demand to specify the model of the object to track, which is required to initialize the tracker or to recover from unexpected situations. On the contrary our main aim is to exploit the tracker information to build models of unknown objects, therefore common trackers such as particle filter based ones are not suitable choices, whilst the motion constitutes a more general cue.

The other advantage of using the independent motion is that, although one object can be learned simply using a still camera taking images from different views, when a robotic platform is available a more realistic interaction is desirable. Looking at an object naturally induces an egomotion stream in the human brain; this occurs if the object is still, but even more if we are observing moving objects. As well as human-human interaction, human-robot interaction should preserve this behavior, otherwise traditionally built offline object datasets already respond the demand.

Many approaches have been proposed regarding the problem of independent motion detection, which is complementary to the estimation of egomotion. Despite the excellent results achieved in the state-of-the-art, all current methods share the same strong assumption requiring egomotion to be the dominant element of the observed scene, see for example [6], [7], [8], [9]. Independent motion on the other hand, can occur only in small portions of the acquired images. It is clear that this restriction is often violated in dynamic scenes that are typical of human environments where many moving subjects are present at the same time, therefore in order to solve this problem, others make use of the known kinematics and stereovision [1], [10]. Despite we have already achieved remarkable performances with our previous approach [1], it relies on the strong assumption that the kinematics of the camera was known. In order to provide a more general method for the automatic image labeling we first extend the proposed framework removing the dependency on the kinematics: we predict the robot egomotion using a heteroscedastic learning algorithm. Modeling the egomotion allows us to detect *anomalies* that correspond to the presence of independent moving objects. The first contribution in this paper is a simplification of the detection scheme proposed in [1]; the motion ROI is then used to automatically label objects. This method reduces the effort in the data acquisition process and allows the robot to build more accurate models of objects, boosting the recognition performances even in challenging robotics scenarios.

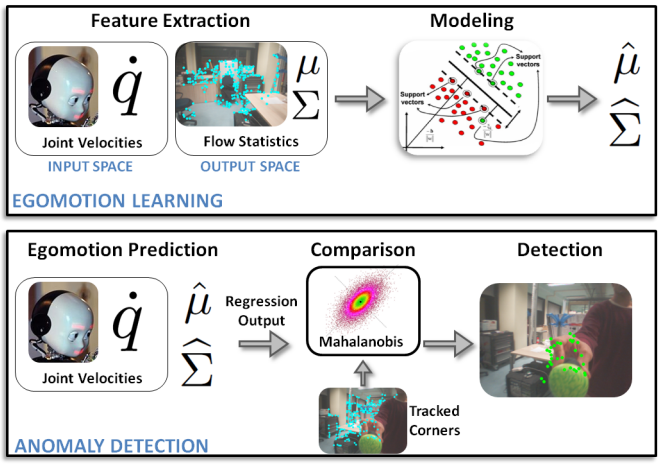

Fig. 2. The independent motion detection method. First the system learns the egomotion using joint velocities and optical flow statistics. In the test phase the expected optical flow is predicted and it is used as probability distribution of the expected egomotion. Flow vectors that do not belong to this distribution are classified as independent moving objects.

In order to show that the system is effectively able to label automatically object classes, we employ an object recognition framework. The problem of recognizing objects is probably one of the most studied in computer vision and robotics communities. Starting from the pioneer Lowe's work [11], a lot of new methods have been proposed in the last years [12], [2] reaching high level of accuracy using object part-based models [13], [14]. In our scenario, due to real-time requirements, we rely on the Yang et al. work [2] using a sparse coding approach on local features that greatly improve performances of linear classifiers.

III. Detecting ROIs through Motion

In our previous work [1] we used the stereo vision system of the iCub in combination with the known kinematics in order to predict the egomotion. In this paper we show that the dependency on the kinematics and the stereo vision can be overcome, making our framework suitable for a larger number of devices. The proposed system is depicted in Fig. 2. The new method can be divided in two stages, namely the egomotion learning stage and the detection of anomalies.

A. Egomotion Learning

The learning of camera movements can be reformulated as a machine learning problem. The basic features come from the proprioperception of the robot joints and from its vision system. During this stage we assume the robot to randomly move its head and eyes in a scene where no independent moving is occurring. The main idea is to learn

the *normal behavior* of the optical flow in presence of pure egomotion, thus we can model independent moving objects as *anomalies* of the optical flow. Following the notation of [1] the followed steps are carried on:

Acquisition
A pair of subsequent images I_{t-1} and I_t is acquired from an RGB camera.

Extraction
A sparse set of N points of interest $\{x_i(t) = (u_i(t), v_i(t))\}_{i=1}^{N}$ is extracted from image I_t. In this work the well-known Harris corner detector [15] is employed, but any typical keypoint detector (e.g. SIFT, SURF) is a viable alternative.

Tracking
Each point $x_i(t-1)$ is tracked to a correspondent $x_i(t)$ in image I_t; this step can be performed using classic sparse optical flow techniques. We chose the well-known KLT algorithm [16] because of its favorable tradeoff between accuracy and computational efficiency.

Computing Flow Statistics
For each pair of frames I_{t-1} and I_t we calculate the set of optical flow vectors $\nu_1(t), \ldots, \nu_N(t)$ where $\nu_i(t) = x_i(t) - x_i(t-1)$. From this set of 2D vectors we compute the expectation optical flow vector $\mu(t) \in \mathbb{R}^2$ and the covariance matrix Σ (a 2×2 matrix) as:

$$\mathbb{E}[\nu] = \mu(t) = \frac{\sum_{i=1}^{N} \nu_i}{N} = \begin{pmatrix} \mu_u \\ \mu_v \end{pmatrix} \tag{1}$$

$$\Sigma(t) = \mathbb{E}[(\nu - \mathbb{E}[\nu])(\nu - \mathbb{E}[\nu])^{\mathrm{T}}] = \begin{pmatrix} \sigma_{uu}^2 & \sigma_{uv}^2 \\ \sigma_{uv}^2 & \sigma_{vv}^2 \end{pmatrix} \tag{2}$$

where μ_u and μ_v are the means of optical flow along the u and v component respectively, σ_{uu}^2, σ_{vv}^2 are the variances along u and v, whereas σ_{uv}^2 is the covariance between u and v directions.

Modeling
For each pair of frames I_{t-1} and I_t we collect the head/eyes joint velocities $\dot{q}(t) = q(t) - q(t-1)$ that represent the *input space*, and the optical flow statistics $[\mu_u(t),\ \mu_v(t),\ \sigma_{uu}^2(t),\ \sigma_{vv}^2(t),\ \sigma_{uv}^2(t)]$ that is our *output space*. Given the training data the problem is reduced to the learning of one function for each output. Namely we train 5 regressors $\hat{\mu}_u(\dot{q}),\ \hat{\mu}_v(\dot{q}),\ \hat{\sigma}_{uu}^2(\dot{q}),\ \hat{\sigma}_{vv}^2(\dot{q}),\ \hat{\sigma}_{uv}^2(\dot{q})$ using ν-SVM [17].

B. Anomaly Detection

Once the normal optical flow behavior has been learned, the occurrence of independent moving objects can be easily detected through a local evaluation of each tracked corner. Intuitively, knowing the high level statistics of the flow implies that each tracked corner should be modeled with these statistics, whereas independent moving pixels will belong to another probability distribution.

Prediction and Comparison
We first predict the flow statistics $\hat{\mu}_u(\dot{q}),\ \hat{\mu}_v(\dot{q}),\ \hat{\sigma}_{uu}^2(\dot{q}),\ \hat{\sigma}_{vv}^2(\dot{q}),\ \hat{\sigma}_{uv}^2(\dot{q})$ using the learned machines, therefore we compare each optical flow vector $\nu_i(t) = x_i(t) - x_i(t-1)$ with the flow statistics. For simplicity of the notation we rewrite the flow statistics in a vectorized form $\hat{\mu}, \widehat{\Sigma}$ such as in Eq. 1 and Eq. 2, and we also suppress the dependency on $\dot{q}$. The comparison between the predicted statistics and the flow vector is obtained by computing the likelihood $P(\nu_i(t) \mid \hat{\mu}, \widehat{\Sigma})$. Intuitively this probability can be interpreted as the system's confidence in observing a particular optical flow given the predicted egomotion.
One may argue that flow statistics could be computed directly in the current images without the prediction step, however this is true only when the egomotion dominates the scene. One of the major contributions of [1] was the detection of the independent motion covering all the scene; with the proposed improvements we want to simplify the detection scheme, while preserving the same benefits and performances.

Detection
As the likelihood $P(\nu_i(t) \mid \hat{\mu}, \hat{\Sigma})$ gets smaller, it becomes likely that a particular optical flow $\nu_i(t)$ is observed when motion $\hat{\mu}$ with covariance $\widehat{\Sigma}$ has been predicted. In other words such distribution represents the probability that a point $x_i(t)$ is being exclusively subject to egomotion. Therefore, we can define a natural criterion for independent motion detection by setting a threshold θ and considering only points $x_i(t)$ such that $P(\nu_i(t) \mid \hat{\mu}, \hat{\Sigma}) > \theta$. In order to detect all points belonging to the probability distribution that generates the egomotion, we compute the Mahalanobis distance $\mathbb{M}(\nu_i(t), \hat{\mu}, \widehat{\Sigma})$ between each flow vector $\nu_i(t)$ and the flow statistics $(\hat{\mu}, \widehat{\Sigma})$:

$$\mathbb{M}(\nu_i(t), \hat{\mu}, \widehat{\Sigma}) = \sqrt{(\nu_i(t) - \hat{\mu})^{\mathrm{T}}\ \widehat{\Sigma}^{-1}\ (\nu_i(t) - \hat{\mu})} \tag{3}$$

A simple criterion to detect independent moving objects is to consider only the flow vectors $\nu_i(t)$ such that $\mathbb{M}(\nu_i(t), \hat{\mu}, \widehat{\Sigma}) > \theta$. In [1] we showed that the problem of egomotion prediction is heteroscedastic, i.e. the output standard deviation is input-dependent, therefore the output noise is not constant. Intuitively this means that for high egomotion movements (i.e. high joint velocities $\dot{q}$) it is likely that the distance between the predicted optical flow mean $\hat{\mu}$ and the actual one μ could increase, leading to detection errors. In order to overcome this issue we learned also the covariance matrix $\widehat{\Sigma}$ that represents the output noise depending on the input $\dot{q}$.

IV. Active Tracking

The active tracker of unknown objects is particularly challenging because it requires the independent motion detector

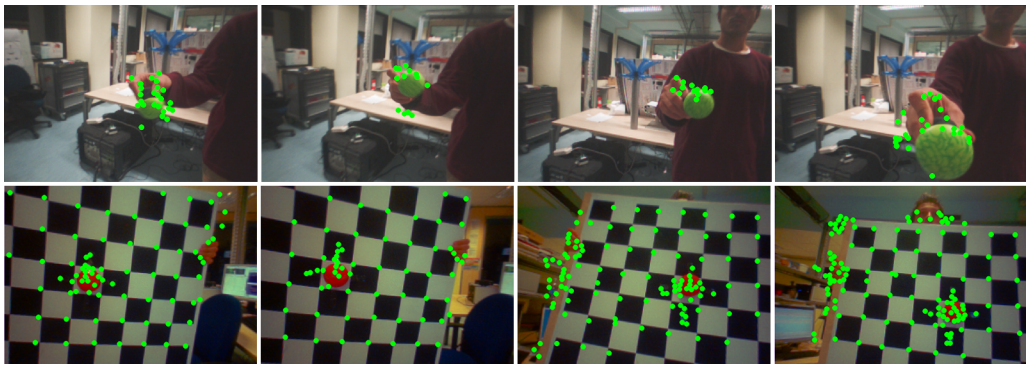

Fig. 3. Top row: detection results for small moving objects. Bottom row: a sequence of motion detection when the moving objects are the biggest regions of the image. In this experiment the robot is tracking the red ball attached to a chessboard.

to perform in real-time. In addition, when the camera follows the target, the induced egomotion and the target flow of motion often become parallel. This makes the problem even harder since it increases the chances of misclassification. We performed active tracking providing as target to the neck/head controller the center of mass of those image points detected as moving independently. Assuming that only one object is active in the scene, this strategy resulted sufficient to achieve accurate tracking. Details on the gaze controller lies outside the scope of this paper and are reported in [18]. The active tracking provides a Region of Interest modeled as a bounding box around the tracked object. From this region we learn proper descriptors for the recognition stage.

V. Learning and Recognition of Objects

The object recognition framework chosen in our application is the work of Yang et al. [2], which fulfills both accuracy and real-time requirements. The main idea of their approach is based on biological evidence that coding schema or vector quantization of local features are the same mechanism coming from neurons of the visual cortex V1, which produces sparse and overcomplete activations [19]. Working with sparse data ensures high quality results in terms of accuracy even if simple linear models are used in classification stage. We briefly introduce the general recognition pipeline that can be mainly divided in three main stages.

- **Features Extraction**. From the image I, we compute local descriptors on a dense grid, such as [20], obtaining a set of features $\mathbf{x}_1, \ldots, \mathbf{x}_m$ with $\mathbf{x}_i \in \mathbb{R}^n$. Popular local descriptors are SIFT [21], SURF [22] or simply the greyscale values. Another viable choice is to run a corner detector such as in Sec. III, however it has been shown that dense grids lead to higher recognition performances [20].
- **Vector Quantization/Coding Stage**. Local descriptors are usually redundant representations and they are not enough informative since they do not catch high level statistics of an image. A well known approach to overcome this issue is the vector quantization (VQ) method that applies the K-means clustering algorithm to learn a *codebook* $V = [\mathbf{v_1}, \ldots, \mathbf{v_K}]$, with $\mathbf{v}_i \in R^n$. With this centroids, local features are remapped into a compact histogram $B \in \mathrm{R}^K$ counting their cluster memberships (hard quantization) or using their distances among clusters as weights (soft quantization). Hopefully, high density areas of this histogram are the same for features belonging to the same class and different among the other classes. This approach is also referred to as Bag of Words (BoW) paradigm. The main limitation of this method is that, in order to obtain good performances, BoW must be applied together with particular types of nonlinear kernel [2]. This means that the classification stage costs $\mathcal{O}(M^3)$ where M is the size of the dataset. To overcome this issue Yang et al. provided the linear spatial pyramid matching with Sparse Coding [2]: the coding stage maps the input features $\mathbf{x}_1, \ldots, \mathbf{x}_m$ into a new overcomplete space $\mathbf{u}_1, \ldots, \mathbf{u}_m$. This step can be achieved minimizing:

$$\min_{\mathbf{U}} \|\mathbf{X} - \mathbf{D}\mathbf{U}\|_F^2 + \lambda \|\mathbf{U}\|_1 \tag{4}$$

where $\mathbf{D}$ ($n \times d$ matrix) is a previously learned dictionary, $\|\cdot\|_1$ is the $l1$-norm, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$ are the local descriptors in the images and $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$ that is a $d \times m$ matrix are the new coded features. An image is still represented as a set of local descriptors that cannot be directly fed to classifiers. Therefore a *pooling* operator must be employed in order to obtain a compact representation of the given frame. Experimental results [23] showed that the **max** pooling operator obtains the highest performances in terms of accuracy for classification tasks. After this stage the final descriptor becomes a single feature vector.
- **Classification** At this stage each image is represented as a vector $\bar{\mathbf{u}} \in \mathrm{R}^K$ where K is the size of the dictionary $\mathbf{D}$. Classical machine learning algorithms can be applied in order to build a model for each class. In order to preserve real-time performances we fed the final description to a linear classifier such as linear SVM [24]. We use a one-versus-all strategy to train a binary linear SVM for each class O_s, so that at the end of the training phase we obtain a set of N linear SVM

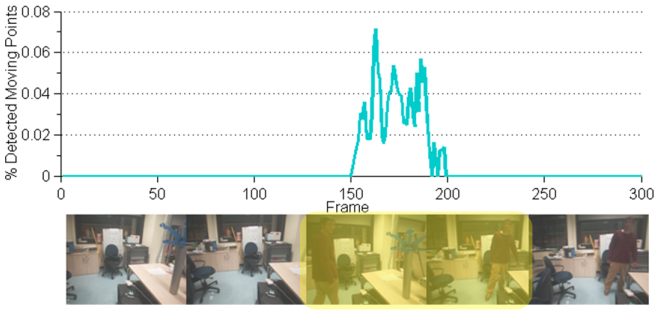

Fig. 4. Example of anomaly detection: in the first frames no moving pixels are detected and only egomotion is occurring, whereas the percentage increases when an anomaly comes into the scene.

classifiers $f_1(\bar{\mathbf{u}}), \ldots, f_N(\bar{\mathbf{u}})$, where N is the number of objects. Given the training data $\{\mathbf{U}, \mathbf{y}\}$ where $\mathbf{U}$ is the set of positive and negative examples for the class O_s, $y_i = 1$ if the example is positive, $y_i = -1$ otherwise, the goal of SVM is to learn a linear function $(\mathbf{w}^T, b)$ such that a new test vector $\bar{\mathbf{u}}$ is predicted as:

$$f(\bar{\mathbf{u}}) = y = \mathbf{w}^{\mathrm{T}} \bar{\mathbf{u}} + b \tag{5}$$

In the one-vs-all paradigm each image $\bar{\mathbf{u}}$ will be labeled with i-th class such that:

$$\max_i f_i(\bar{\mathbf{u}}) \tag{6}$$

With both the independent motion detector and an object recognition framework it is possible to obtain a real human-robot interaction for object recognition tasks.

VI. Experiments

A. Robotic Platform

The proposed application is designed and implemented for the iCub [5], a humanoid robot that resembles a 3 years old child. It has 53 degrees of freedom (DoFs) among head, arms, hands and legs and it has been intended for manipulation tasks, indeed hands and arms contain the biggest number of DoFs. In particular, the hands have 9 DoFs, each with three independent fingers; the fourth and fifth are linked and they are meant to be used for additional stability (only one DoF). From the sensory point of view, the iCub is equipped with digital cameras, gyroscopes and accelerometers, microphones, and force/torque sensors. Additionally the torso, arms, hands and fingertips are equipped with a sensorized skin that can be used for haptic perception. Two standard dragonfly cameras play the role of the eyes, streaming 320x240 resolution images at 33 Hz. Notably the image resolution is forced to be low as the necessity to process data in real time imposes strict bandwidth limits on the network. The software running on the iCub is released under the GPL license and it is publicly available for download.

B. Independent Motion Detection

As described in Sec. III we used the robot joint velocities $\dot{q}$ as input features. For these experiments we used the 6 DOFs of the iCub's neck/eyes. In order to learn the egomotion we relied on the *libsvm* implementation [25], employing a RBF kernel. The SVMs parameters and the threshold θ have been estimated through a standard grid-search with K-fold cross-validation procedure. The test phase is performed in real-time ($>$ 30fps, which is the camera bandwidth upper bound) thanks to the low dimensional space of the input features and the quality of the implementation. During the training phase a video of approximately 3000 frames has been acquired with the robot randomly moving eyes and neck while no other motion was occurring. In Fig. 4 an example of anomaly detection is depicted: the camera is randomly moving and the anomaly is detected when it comes into the scene. In Fig. 3 top row an example of independent moving object detection is shown. In the bottom row we replicated the same experiment performed in [1] in order to show that the system is still able to perform motion detection even though the moving objects cover the majority of the scene. We set up a scenario where we imposed a high percentage of independent motion in the scene by moving a large chessboard pattern; while the robot was tracking a red ball attached to the chessboard, we were moving the whole structure. The off-line learning procedure of the optical flow statistics allows dropping, at run time, the egomotion's dominance assumption.

C. Automatic Image Labeling

We consider a common setting in the HRI scenario. A demonstrator in front of the robot asks it to actively track an unknown moving object through a simple speech recognition system; the active tracking is carried out as described in Sec. IV, assuming that only one moving entity is present. The demonstrator rotates the object in order to show the robot all the possible views, making the learning phase more reliable and enabling a natural interaction with iCub. These images are then used by the robot that extracts SIFT descriptors and builds one model for each object following the pipeline described in Sec. V. A few frames of the learning stage are shown in Fig. 5, top row. In the recognition phase the robot recognizes which object is in the operator's hand as depicted in Fig. 5, bottom row, and Fig. 7. Notably, there are some frames (see Fig. 5, bottom row, frame 3) where the recognition fails due to the bounding box retrieved by the independent motion detector: indeed the bounding box is not centered in the object and this leads to classification errors.

D. Recognition Results

In order to evaluate rigorously the recognition capabilities of the system in presence of moving objects, we followed standard computer vision benchmarks. We considered a dataset of 8 classes; one training video showing different views of an object is used for each class, whilst two videos per class have been recorded for the test stage. Each video contains at least 500 frames. We computed performances in three different conditions: **per frame classification**, where each frame must be correctly classified and it contributes for the final accuracy; **without the bounding box information**,

Fig. 5. Qualitative results for the learning and recognition phase. In the top row the acquired dataset and an example of learning stage is depicted: the demonstrator shows an unknown object and the iCub tracks it. In the bottom row the recognition phase is showed: the classifier misclassifies the third frame due to the bounding box retrieved by the independent motion detector.

and lastly we show how to improve the accuracy via **temporal smoothing** of the classifier scores. The performances have been computed in terms of precision-recall curves and Average Precisions (APs) following the PASCAL Visual Object Classes (VOC) Challenge protocol [4]. Recall and Precision values are defined as:

$$Recall = \frac{tp}{tp + fn} \tag{7}$$

$$Precision = \frac{tp}{tp + fp} \tag{8}$$

where tp and fp are true positive and false positive respectively and fn are the false negative examples. In the computation of the precision-recall curves recall is defined as the portion of all the positive examples scored above a fixed score, whereas precision is the portion of all examples above the score that are from the positive class. The AP is a compact value that summarize this curve and is defined as the mean precision at a set of eleven recall levels $[0, 0.1, \dots 1]$ (see [4] for details). In Fig. 6 the curves of the per frame classification (violet) and the temporal smoothing (cyan) are depicted for the 8 classes of the dataset; notably we get a mean AP equal to 83.19% in the case of per frame classification; this means that the system is effective also in presence of cluttered scenes and different views of the object thanks to the employment of the egomotion compensator. Indeed, removing the bounding box retrieved by the motion, the mean AP drops to 44.61%. This behavior was expected due to the complexity of the varying background and the choice of the particular dataset: all the objects were chosen according to the manipulation capabilities of the iCub, therefore we only considered small objects covering a low percentage of pixels in the scene.

E. Boosting the Recognition Rate

Despite the good results obtained with the per frame classification, it is clear that in robotics scenarios higher recognition rates are required. In particular the system fails

	Turtle	Octo	Car	Lettuce	Bottle	Box	Phone	Pouch
Turtle	97%			3%				
Octo		96.9%	3.1%					
Car		2.1%	97.9%					
Lettuce	5.8%	4.2%	2.9%	87.1%				
Bottle					93%	7%		
Box						95.6%	4.4%	
Phone						5.4%	94.6%	
Pouch					2.4%	1.9%	1.1%	94.6%

TABLE I

CONFUSION MATRIX OF THE 8 OBJECTS. WE USED THE TEMPORAL SMOOTHING PROCEDURE DESCRIBED IN SEC. VI-E

the recognition in particular conditions (i.e. wrong bounding box) that indicates peaks in the classifier scores, leading to misclassifications. In order to prevent this behavior we select a buffer of $T = 5$ frames and we compute the classification scores as the average score in the selected temporal window for each classifier. The highest average score within the buffer will represent the recognized object. This allows improving the recognition rate, obtaining a mean AP equal to 97.6% (see also the cyan curves in Fig. 6). As deeper per-class analysis, we also computed the confusion matrix: each column of the matrix represents the instances in a predicted class, whilst each row represents the instances in an actual class. As result, we get an overall mean accuracy of 94.56%. This improvement makes the system really robust in assigning labels and in learning objects even in this challenging setting.

Examples of the independent motion detection, learning and recognition of objects can be found in the attached video.

VII. DISCUSSION

This paper was our first attempt to fill the gap between computer vision field and robotics, showing how to increase the learning capabilities of the iCub robot exploiting a weak supervision given by a human demonstrator. We first

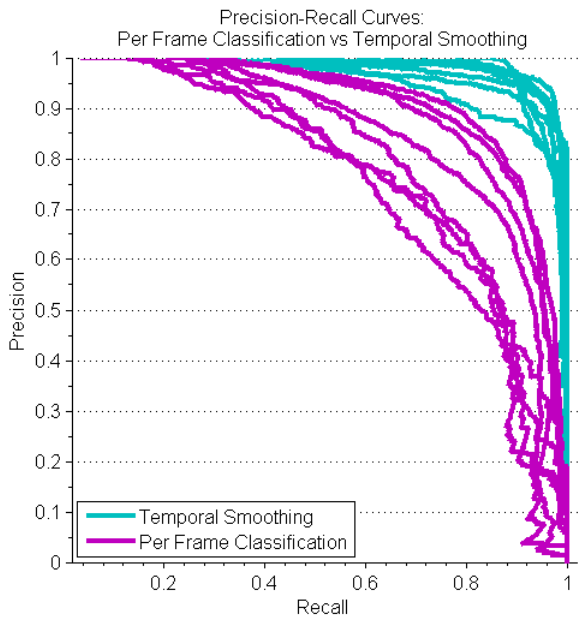

Fig. 6. Precision-Recall curves for the considered dataset. Each curve correspond to a particular class of the dataset; in cyan the performances obtained via temporal smoothing (see Sec. VI-E), in violet the per frame classification results. We followed the standard Visual Object Challenge (VOC) protocol [4] for the curve computation.

extended the independent motion detector [1] simplifying the overall pipeline and removing the dependency on known kinematics and stereo vision. These improvements made the method suitable for a larger number of devices, while still maintaining high performances even when the independent motion dominates the scene. Furthermore we presented an application aiming at enabling a realistic human-robot interaction when robot and human share the same goal of image labeling. The application was implemented and validated experimentally on a robotics platform, iCub, but is meant to be generalized to other robots. Motion is a powerful cue for a large set of vision applications (people or car detection, object segmentation, human-computer interaction), we showed in this paper a possible application in the HRI field. We are currently working on a high-level processing of the predicted independent motion; our goal is to handle multiple active entities in the scene and use the motion information for region segmentation, tracking purposes and multiple object recognition/labeling tasks.

REFERENCES

[1] C. Ciliberto, S. R. Fanello, L. Natale, and G. Metta, "A heteroscedastic approach to indipendent motion detection for actuated visual sensors," in *IROS*, 2012.

[2] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009.

[3] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *CVPRW*, 2004.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, 2010.

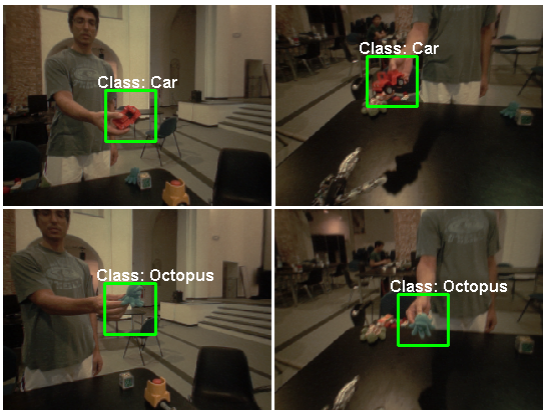

Fig. 7. Other examples of the recognition phase.

[5] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *8th Workshop on Performance Metrics for Intelligent Systems*, 2008.

[6] M. Irani and P. Anandan, "A unified approach to moving object detection in 2d and 3d scenes," *PAMI*, 1998.

[7] R. Nelson, "Qualitative detection of motion by a moving observer," in *CVPR*, 1991.

[8] B. Jung and G. S. Sukhatme, "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," in *International Conference on Intelligent Autonomous Systems*, 2004.

[9] A. A. Argyros and S. C. Orphanoudakis, "Independent 3d motion detection based on depth elimination in normal flow fields," 1997.

[10] N. Moutinho, N. Cauli, E. Falotico, R. Ferreira, J. Gaspar, A. Bernardino, J. Santos-Victor, P. Dario, and C. Laschi, "An expected perception architecture using visual 3d reconstruction for a humanoid robot," in *IROS*, 2011.

[11] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999.

[12] L. Fei-Fei, R. Fergus, and A. Torralba, "Recognizing and learning object categories," *CVPR*, 2007.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *PAMI*, 2010.

[14] R. Girshick, P. Felzenszwalb, and D. McAllester, "Object detection with grammar models," in *ICCV Workshops*, 2011.

[15] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[16] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *DARPA Imaging Understanding Workshop*, 1981.

[17] B. Schölkopf, A. Smola, R. C. Williamson, and P. Bartlett, "New support vector algorithms," *Neural Compututation.*, 2000.

[18] U. Pattacini, "Modular cartesian controllers for humanoid robots: Design and implementation on the icub," Ph.D. dissertation, RBCS, Italian Institute of Technology, 2010.

[19] B. A. Olshausen and D. J. Fieldt, "Sparse coding with an overcomplete basis set: a strategy employed by v1," *Vision Research*, 1997.

[20] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, 2005.

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

[22] H. Bay, A. Ess, T. Tuytelaars, and L. Vangool, "Speeded-up robust features (SURF)," *CVIU*, 2008.

[23] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010.

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

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.