Online Multiple Instance Learning Applied to Hand Detection in a Humanoid Robot

Carlo Ciliberto, Fabrizio Smeraldi, Lorenzo Natale and Giorgio Metta

***Abstract*— We propose an algorithm for the visual detection and localisation of the hand of a humanoid robot. This algorithm imposes low requirements on the type of supervision required to achieve good performance. In particular the system performs feature selection and adaptation using images that are only labelled as containing the hand or not, without any explicit segmentation. Our algorithm is an online variant of Multiple Instance Learning based on boosting. Experiments in real-world conditions on the iCub humanoid robot confirm that the algorithm can learn the visual appearance of the hand, reaching an accuracy comparable with its off-line version. This remains true when supervision is generated by the robot itself in a completely autonomous fashion. Algorithms with weak supervision requirements like the one we describe are useful for autonomous robots that learn and adapt online to a changing environment. The algorithm is not hand–specific and could be easily applied to wide range of problems involving visual recognition of generic objects.**

I. Introduction

On-line adaptation is an essential capability for cognitive robots operating in the real world. As the robotics community devotes growing attention to the development of such systems, there is an increasing demand for learning techniques that allow data acquisition and training to be performed online and largely autonomously.

Ideally, training data should be collected automatically without human supervision. Learning in real world applications is therefore hampered by an unfavorable tradeoff between the accuracy of the training examples and their availability. One way to relax the requirements on supervision is to adopt a learning paradigm such as Multiple Instance Learning (MIL). In this framework, training examples come in "bags" that contain positive and negative instances sharing a common label; a positive bag contains at least one positive instance, while a negative bag is guaranteed to contain no positive instances at all.

In this work, we develop an online MIL algorithm that uses an online variant of Adaboost [25], [15] to combine a family of weak hypotheses specifically designed for MIL [4]. As the online algorithm does not make provisions for feature selection, we implement this by wrapping weak learners inside *selectors*, similarly to what is done in [17]. We apply our algorithm to a visual learning problem, namely hand detection in the iCub humanoid robot (Figure 1). Our algorithm only requires knowledge of whether the hand is present in the visual field for training. This enables the robot to learn the appearance of the hand without any supervision other than a self-generated coarse labelling based on the co-occurrence of motion in the visual stream and in the motors. Experiments are performed in a challenging scenario using images coming from the embarked cameras while the robot is operating in realistic, cluttered environment. The resulting classifier performs reliably, with error rates comparable to its offline version. Remarkably we demonstrate that the salient features identified in the process of learning the MIL classifier can be used to localise the hand in the visual field with good accuracy: this is an important result because it shows that the algorithm can be useful not only for object detection but also for localisation.

We are aware of only a few other online MIL algorithms in the literature [5], [28], also based on a variant of boosting. While these are specifically applied to tracking, our main focus in this work is on self-supervised learning of the appearance of the hand, rather than on tracking. Our approach further differs in that the Multiple Instance nature of the problem is dealt with at an early stage by the weak learners rather than at the level of the ensemble method; this allows the classification of targets described by more than one positive instance (see Section III).

To summarize this paper makes two contributions: i) it proposes an algorithm for autonomous learning of the visual appearence of the hand and ii) it describes the implementation of a novel online Multiple Instante Learning algorithm and its application to robotics. To the best of our knowledge the implementation of a MIL algorithm on a robotic platform has never been reported before.

II. Previous work

A. Hand detection

Visual localisation of the end-point is crucial for closed loop control of robotic manipulators [19]. Normally this problem is solved by employing markers that greatly simplify the detection. This approach, however, has clear limitations and can be applied only in controlled settings. In humanoid robotics, there have been recently some attempts to solve the problem of hand detection in a generic way [24], [20], [16]. These works share a common idea in that they integrate vision with the arm joints state to perform the autonomous visual segmentation of the hand. In these cases, however, the main concern is to investigate to what extent the integration

This work is funded by the European Commission as part of the project ICT-FP7-215843 Poeticons.

C.Ciliberto, L.Natale and G.Metta are with the Robotics Brain and Cognitive Science Department, Istituto Italiano di Tecnologia, Genova, Italy. carlo.ciliberto, lorenzo.natale, giorgio.metta @ iit.it

F. Smeraldi is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, UK. fabri@eecs.qmul.ac.uk

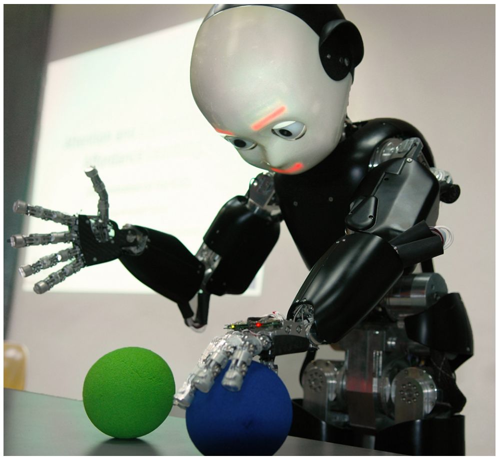


Fig. 1. The hardware platform: the iCub robot

between vision and motorial information can help solve the hand detection task. These works generally employ simple visual descriptors (for example [20] uses a colour histogram); arguably better result could be expected by using more sophisticated and robust visual features (e.g. [21], [7]).

B. Multiple Instance Learning

The MIL approach originated in a bio-informatics setting in the late Nineties [12], [3], when the Diverse Density algorithm was developed [22], and was quickly applied to object recognition. In our terms, MIL corresponds to a scenario in which training images (that play the role of the bags of instances) are identified as either containing or not containing the object of interest, without its location and size being specified. Interest in this technique has been renewed more recently with the development of SVM-based algorithms such as DD-SVM [10] and MILES [9].

Of special interest to us are the boosting-based approaches [2], [27], [4], that lend themselves more naturally to modifications for on-line learning because of the iterative nature of boosting algorithms. These approaches differ in the particular flavour of boosting used as well as in the way that the MIL paradigm is implemented. A variant of the Linear Programming Boosting framework (LPBoost) is used in [2]. At each iteration of LPBoost, a linear programming problem is solved to maximise the margin of the training examples. The MIL generalisation relaxes this requirement by optimising to achieve a large margin for at least one of the patterns in each bag. In [27] the AnyBoost framework is used. Weak learners classify single instances, and the probability that a bag is positive is obtained by the Noisy-OR of the probability of each instance being positive. The target function optimised by AnyBoost is the likelihood of each bag being positive. The online MIL algorithm presented in [5] builds on this approach by introducing a variant of AnyBoost. Weak learners based on Haar-like features are trained in an online fashion; for each new training example however, a new strong classifier is estimated from scratch by the boosting algorithm. In the semi-supervised MIL approach presented in [28] the Noisy-OR function is replaced by a geometric mean and gradient descent on the loss function is used for boosting.

In the above boosting approaches, with the exception of [4], the weak learners act on single instances, while the multiple instance problem is handled by the ensemble algorithm. By contrast, in [4] the weak learners themselves, defined as balls of optimal radius in feature space, directly classify bags of instances as opposed to single instances. By dealing with bags of instances at the level of the weak learners, this algorithm allows using a standard implementation of two-class Adaboost (or indeed, of any equivalent variant). As detailed below, we choose this approach because of its flexibility and of the compatibility of the particular type of weak learners with our choice of visual descriptors.

III. Contributions

We propose an online implementation of Multiple Instance Learning for the purpose of semi-supervised recognition of a robot hand. The MIL nature of the problem is derived from application: the robot does not initially know the appearance or the location of the manipulator in the visual field, but it can control the motors to bring the manipulator in view or out of view.

We extract from the image a set of interest points on which we compute SURF descriptors [7]. SURF are robust to scale and orientation variations and have been empirically proved [6] to be remarkably faster than SIFT [21], which makes them more suitable for online processing. Descriptors are extracted from the whole image; thus the object of interest (the hand) is represented by a group of relevant descriptors (instances) embedded in a larger positive bag, that also includes interest points from the background. This differs from [27], [5] where the object is represented by a single instance. It also differs from semi-supervised approaches such as [18] in which some of the instances are labelled and pseudo-labels are estimated for the other instances: in our case, the position of the manipulator being unknown, all images (bags) are labelled while no individual instances are.

For the reasons detailed above, it is crucial that the weak learner themselves are able to deal with multiple instances; we therefore follow the boosting-based approach of Auer et al. [4], as their framework involves weak learners that directly classify bags rather than single instances. Also, the specific nature of the weak learners they propose (detailed below in Section IV-B) makes them more suitable for classifying high-dimensional descriptors such as SURF.

In order to adapt the algorithm for online use, the standard AdaBoost algorithm used in [4] must be replaced with an online boosting procedure such as described in [13], [25], [8]. We chose the algorithm introduced by Oza and Russell in [25], that has already been used with success in vision applications [17]. While Oza's algorithm can be trained more efficiently than the variant used in [5], it has no provisions for feature selection. This limitation has been dealt with in [17] by introducing *selectors*, that are essentially wrappers for the weak learners that allow selecting the hypotheses that perform best. We follow this scheme, apply it to the MIL problem and extend it to the case that not all the weak learners are known from the start (see Section IV-C).

Finally, we prove that the salient features identified by the boosting algorithm and the selectors during the process of training the strong MIL classifier can be used to effectively locate the position of the hand in the image.

We present an implementation of our algorithm on an iCub humanoid robot and use it to produce the experimental results reported in Section V. The code has been submitted to the iCub repository [1] and is available to all researchers using the open-source robotic platform.

IV. Multiple Instance Learning and Online Boosting

A. AdaBoost and online boosting

AdaBoost is a well established off-line boosting algorithm that adopts a greedy strategy to combine a series of inaccurate weak classifiers into a highly precise *strong classifier* [15]. It does so by maintaining a distribution of weights $\mathbf{\Lambda}$ over the training set. At each iteration, the weak learner with the lowest misclassification rate with respect to $\mathbf{\Lambda}$ is added to the strong classifier with a coefficient dependent on its accuracy. The weights are subsequently updated so that misclassified training examples become more important at the next iteration.

The main obstacle to an online formulation of the algorithm is the need to keep track of a weight distribution over a training set that is constantly growing. In the online variant introduced by Oza [25] the iterative structure of the algorithm is retained, but the examples are now propagated down a pre-ordered, fixed list of weak learners that make up the strong classifier (Table I). Each weak learner increases (decreases) the weight of the samples it misclassifies (classifies correctly) before passing them on to the next weak learner.

Finally, each weak learner keeps track of its error rate based on the weight of the samples it classifies. Because the weak learners are fed the training examples one at a time, an online *Learning Principle* (Figure 2) needs to be specified for them.

B. MIL and Boosting

Auer and Ortner [4] proposed to combine MIL and boosting from the perspective of high dimensionality features. In their framework, a weak learner is a ball B in the feature space $\mathbb{R}^N$. If we denote a bag of instances by $I = \{x_i\}$, a ball B classifies as positive the bags I such that $I \cap B \neq \emptyset$. Under these assumptions, given a training set $\mathcal{I}$ over which a weight distribution $\mathbf{\Lambda}$ has been provided, the quality of any classifier B can be assessed by evaluating its *distribution accuracy* $D(B, \mathbf{\Lambda})$, i.e. the sum of all the weights $\mathbf{\Lambda}$ associated to the training bags correctly classified by B.

In the original work, classical Adaboost is applied to the set of weak classifiers represented by the balls $\{B_r(x)\}$ centred on every positive instance x in the training set. Weak learners are trained by optimising their radius according to $r = \operatorname{argmax}_{r'>0} \; D(B_{r'}(x), \mathbf{\Lambda})$; however, this is not directly feasible in an online context as training data are provided to the algorithm in a sequential fashion.

```
Online Boosting

Initialization
  • Let $\mathcal{H} = \{h_1 ,..., h_N\}$ be a set of weak classifiers with Learning Principle $L$.
  • Set $\lambda_n^w = \lambda_n^c = 0 \;\; \forall n \in \{1,...,N\}$.
Training
At each iteration step $t$ a novel sample $I_t$ is presented to the system:
  • Set the importance weight of the sample to $\lambda = 1$.
  • For $n \in \{1,...,N\}$ do:
      1) update $h_n \leftarrow L(h_n, I_t, \lambda)$
      2) if $h_n$ correctly classifies $I_t$:
           – $\lambda_n^c \leftarrow \lambda_n^c + \lambda$ and $\lambda \leftarrow \lambda \frac{1}{2(1-\epsilon_n)}$
         else:
           – $\lambda_n^w \leftarrow \lambda_n^w + \lambda$ and $\lambda \leftarrow \lambda \frac{1}{2\epsilon_n}$
         where the updated error is $\epsilon_n = \frac{\lambda_n^w}{\lambda_n^c + \lambda_n^w}$ .
      3) Define the relevance weight of the n-th weak learner as $\alpha_n = \log\left(\frac{1-\epsilon_n}{\epsilon_n}\right)$
Strong Classifier
After every learning iteration, the score assigned by the strong classifier to a bag $I \in \mathcal{I}$ is:
  $$S(I) = \sum_{n=1}^{N} \alpha_n \cdot h_n(I) \; .$$
```

TABLE I

Oza and Russel's online AdaBoost algorithm [25]

In Table II we propose an adaptation of Auer and Ortner's ball learners to the online framework presented in Section IV-A. The main difference is in the Learning Principle: whenever a novel training bag arrives, the radius is updated to keep the distribution accuracy maximized. However, as new data comes in, training samples with the lowest weights assume less and less importance and can be discarded to avoid memory overstress.

C. Weak learner selectors

MIL over a continuous data stream can in principle be achieved by applying the online boosting algorithm described in Section IV-A to the weak learners introduced in Section IV-B above. In an online context, however, it is likely that useful and descriptive features (and hence potential centres for new weak classifiers) will not be available from the start, but may become available, for instance, as the object to be learned rotates and some of its previously hidden parts become visible. Unfortunately, the algorithm in Table I has no way to access such information, as it requires the set of weak learners to be defined a priori.

A possible solution is to employ a class of more general weak learners: the *selectors*. These were originally introduced in [17] to approach feature selection problems via online boosting. A selector acts as a wrapper for a pool of weak learners (Table III). Whenever a training sample

Online MIL weak learners

Definition
An *online MIL* weak learner is a pair $h = (B_r(x), \mathbf{\Lambda})$ associated to a weighted distribution $\mathbf{\Lambda}$ and to a ball $B_r(x) \subset X$ centered on a positive instance x .

Classification

$$\forall I \in \mathcal{I} \qquad h(I) = \begin{cases} 1 & \text{if } I \cap B_r(x) \neq \emptyset \\ -1 & otherwise \end{cases}$$

Learning Principle (Figure 2 (Left))
For any pair $(I, \lambda) \in \mathcal{I} \times \mathbb{R}_+$ and weak classifier $h = (B_r(x), \lambda)$, the Learning Principle $L(h, I, \lambda)$ is defined as follows (see Figure III Left):

- update $\mathbf{\Lambda} \leftarrow \mathbf{\Lambda} \cup \{(I, \lambda)\}$.
- if $|\mathbf{\Lambda}| > n_{max}$, eliminate from $\mathbf{\Lambda}$ the pair (I, λ_{min}) with minimum weight λ_{min}.
- compute $\tilde{r} = argmax_{r>0} \quad D(B_r(x), \mathbf{\Lambda})$.
- return the updated weak classifier $h = (B_{\tilde{r}}(x), \mathbf{\Lambda})$.

TABLE II
DEFINITION OF ONLINE MIL WEAK LEARNER

arrives, the selector updates all the weak learners in its pool and marks the one with lowest error rate as its current decision function. At the same time, the element that exhibits the worst classification performance is substituted with a new one, selected at random from a weak learner cache that stores all weak classifiers corresponding to the instances in the positive bags seen so far. This allows the boosting algorithm to access novel weak learners as they are needed.

V. Hand Detection and Localisation

A. Robotic Platform

The experiments described in this paper were carried out on an iCub humanoid robot (see Figure 1). The iCub is a complete humanoid robot, with 53 degrees of freedom [23]. This platform was designed to study manipulation and for this reason particular attention has been devoted to the design of the hands (overall each arm has 16 degrees of freedom). Six motors actuate the head: three control the head at the level of the neck, whereas other three control two cameras around a common tilt axis and two independent pan axes. The sensory system includes two cameras for vision, an inertial sensor, force and position feedback from all the motors (optical encoders).

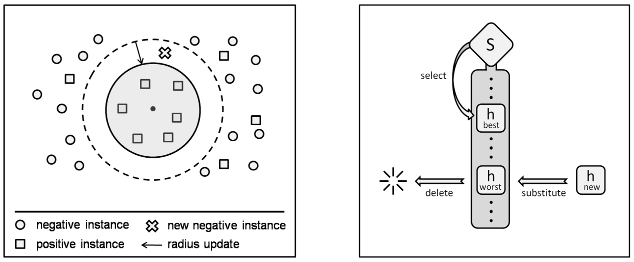


Fig. 2. **Learning Principles**. (Left) Example of a learning step for an online MIL weak learner: a new negative instance has occurred (the cross) and the radius is consequently reduced to keep the distribution accuracy optimised. (Right) a typical learning iteration for a selector: the best weak learner is chosen as current decision function while the worst one is substituted with a completely new one randomly extracted from the weak learner cache.

Selectors

Definition
A selector is a couple $s = (P, \bar{m})$ where $P = \{(h_1, \epsilon_1)$,..., $(h_M, \epsilon_M)\}$ is a set (or pool) of weak learners h_i with associated error rate ϵ_i and $\bar{m}$ is the index of the weak learner currently chosen by the selector

Classification

$$\forall I \in \mathcal{I} \qquad s(I) = h_{\bar{m}}(I) \ .$$

Learning Principle (Figure 2 (Right))
For any couple $(I, \lambda) \in \mathcal{I} \times \mathbb{R}_+$ and selector $s = (P, \bar{m})$, the learning rule $L(s, I, \lambda)$ is defined as follows (see Figure III Right):

- For $i \in \{1$,..., $M\}$ do:
  - update $h_i \leftarrow L^h(h_i, I, \lambda)$ where L^h is h's Learning Principle.
  - update the error rate ϵ_i (as in step 2. of Table I) according to $h_i(I)$.
- set $i_{min} = \text{argmin}_i \ \epsilon_i$ and $i_{max} = \text{argmax}_i \ \epsilon_i$.
- substitute $h_{i_{max}}$ with a new weak learner chosen at random and set $\epsilon_{i_{max}} = 0$.
- return the updated selector $s = (P, i_{min})$

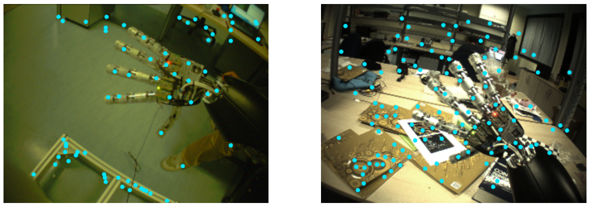
TABLE III
DEFINITION OF A WEAK LEARNER SELECTOR

The iCub is an open system, the design and documentation of hardware and software are licensed under the Free Software Foundation licenses. All parts of the system can be freely replicated and customised; at the moment of writing several copies of the robot exist in research laboratories around the world (though mostly in Europe). Researchers working on these robots have access to a large repository of software [1] which contains the results of the work of other laboratories, including the work presented in this paper.

B. Hand Detection

We validate the online MIL boosting framework by testing it on the hand detection problem. The system is trained using the image stream from the right eye camera, acquired while the robot performed random gaze shifts and right arm movements. Gaze shifts involved motion of the head and torso and increased background and illumination variability. The frequency of hand occurrences in the data was controlled and kept around 50%. In this paper we train the detector on a single view of the hand. For this reason the wrist was controlled so that the back of the hand faced the cameras and was approximately parallel to the image plane. Both gaze and arm random trajectories were generated using the iKin cartesian controller [26] available on the iCub repository [1].

We present results over three datasets, each consisting of a sequence of images. The first two datasets were recorded in settings characterised by a different background in order



(a) Uniform background. (b) Cluttered background.

Fig. 3. SURF features extracted from images depicting two different kind of backgrounds.

Dataset	Labelling	Order	Equal Error Rate	
			Online	Offline
Uniform	Manual	Acquisition	(8.8 ±0.3)%	7.3%
(1500 images)		Shuffled	(8.0±0.7)%	
Cluttered	Manual	Acquisition	(13.0±1.9)%	7.5%
(1500 images)		Shuffled	(11±1)%	
Autonomous	Manual	Acquisition	(2.0±0.5)%	0.5%
(9000 images)		Shuffled	(1.1± 0.4)%	
	Auto	Acquisition	(2.3±0.7)%	1.2%
		Shuffled	(1.75±0.5)%	

TABLE IV

EER of the proposed algorithm (Online) and of off-line MIL Adaboost (Off-line) over the three datasets, averaged across 20 runs. Labelling strategies are identified with *Manual* (human supervision) and *Auto* (automatic labelling by the robot). Images are learned either in the order of acquisition (*Acquisition*) or in random order (*Shuffled*).

to assess the system robustness to environmental clutter. The third dataset, on the other hand, was recorded with the purpose of evaluating system performance on a longer sequence and with automatically generated supervision.

Sequence 1 (**Uniform**): In this case the robot operated in a relatively simple environment. The hand generally appears on a uniform, uncluttered background (Figure 3(a)). For each image, the presence or absence of the hand was manually annotated by a human observer. This sequence consist of 1500 frames, of which the first 500 are used for training and the remaining 1000 for testing.

Sequence 2 (**Cluttered**): Feature-rich distractors are added to the environment, so that the features extracted from each image have a much larger probability of belonging to the background (Figure 3(b) shows a typical image). As for the Uniform sequence, image labels are assigned by the human observer. This sequence also consist of 1500 frames, of which the first 500 are used for training and the remaining 1000 for testing.

Sequence 3 (**Autonomous**): As for the cluttered sequence, images are recorded in an environment cluttered by distractors. Images are labelled autonomously by the robot during acquisition by detecting the co-occurrence of movement in the visual field and motion in the motor encoders. This labelling strategy is intrinsically imprecise, leading to a 10% of erroneous training labels. As a baseline, a second set of labels were assigned by the human observer. This sequence consists of 9000 images, of which 3500 are used for training and the remaining 5500 for testing.

We measure the performance of our online MIL classifier on the three datasets by determining the Equal Error Rate (EER). For each sequence, results are averaged across 20 runs in order to account for the random substitution of new weak learners in the selectors. We also test the robustness of the algorithm to changes in the order of appearance of the samples. For this purpose we trained the classifier in two ways: 1) on the natural sequence of images (i.e. as they were acquired by the robot), averaged 20 times and 2) averaging over 20 random permutations of the sequence. Finally we compare the performance of the online classifier with the batch MIL algorithm of Auer and Ortner [4].

Experimental results are summarised in Table IV, that lists for each experiment the average EER across 20 runs and its standard deviation. In all experiments the number of selectors was set to 100, each containing 50 weak learners; these numbers were chosen empirically.

From the comparison between the **Uniform** and **Cluttered** datasets, it can be noticed that the noise caused by spurious feature matches on the background influences the online MIL boost algorithm much more than the batch version. However, tests on the **Autonomous** sequence, that also features a cluttered background, prove that this difference is largely dependent on the limited number of training images.

Figure 4 shows a detailed comparison of the automatic and the manual labelling strategies performed on the **Autonomous** dataset on two distinct runs. The figure shows the EER as a function of the number of training examples when the classifier is trained online using automatic and manually labelled data respectively. In this case the order of the images was untouched. The two error curves show an oscillatory behaviour that is progressively damped as more and more training samples are presented to the system. This is due to the order of presentation of the images, as consecutive samples are strongly correlated as concerns the presence

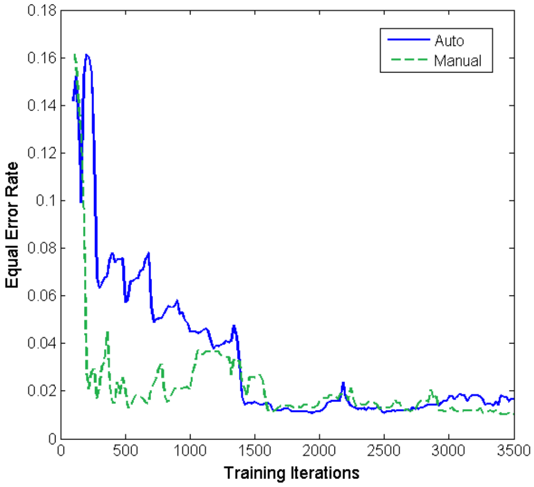


Fig. 4. EER vs number of training images for the **Autonomous** dataset, Manual and Automatic labelling.

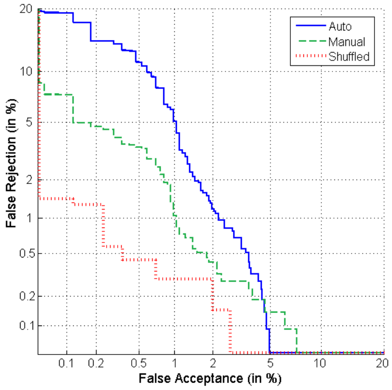


Fig. 5. ROC curves, **Autonomous** sequence. The curves correspond to manual labelling of the original (Manual) and reordered image sequence (Shuffled), and to automatic labelling by the robot (Auto).

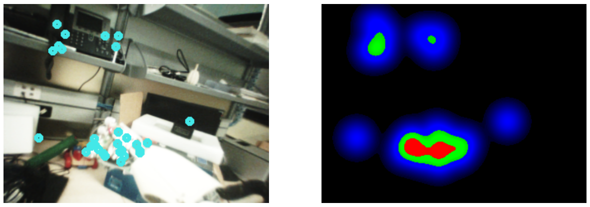
or the absence of the hand. The relative influence of the resulting unbalance between positive and negative training examples decreases as more samples are observed, leading to a more stable EER. The ROC curves obtained after learning all the training examples of this sequence are shown in Figure 5, that also shows a ROC corresponding to a random reordering of the training samples (as in Table IV).

Our algorithm consistently achieves a good detection performance. Even in the case of a cluttered environment, the EER approaches that of the batch version of the algorithm when enough training examples are available. Of particular significance for our application is that the classifier maintains good performance when labels are assigned autonomously by the robot (which introduces mistakes in the training set). Also, performance is largely insensitive to the order of presentation of the samples; this can be thought of as a "sanity check" that confirms that the algorithm can be used online as it can deal with correlation between consecutive images and does not require a randomisation of the training set.

C. *Hand Localisation*

The architecture of the proposed detection algorithm leads to a natural object localisation strategy. During training, the selectors choose from their pools the weak classifiers best suited to detect the object in the training images, i.e. the feature space "balls" centred on vectors that are most likely to belong to the hand. Hence, whenever a selected weak learner responds positively to a given image, we can consider the 2D key-points associated to the feature(s) lying within the ball radius as likely to be placed on the hand.

Let us consider an image positively classified by the strong MIL classifier. The set of feature points that are positively classified by the weak learners (depicted as blue dots in Figure 6(a)) can be loosely interpreted as sampled from the probability distribution $p(x|I)$ of having the hand at position x in the image, given the set of extracted SURF features I. Similarly to what is done in Parzen window estimation [14], we considered the mixture of Gaussians obtained by centring a Gaussian kernel at each of these feature points (Figure 6(b)). The mode of the resulting distribution is our estimate of the location of the hand (this can also be interpreted as the first step in a subtractive clustering procedure, see [11]). We further stabilise the detection by weighted averaging of the mixture of Gaussians across the few frames immediately preceding the image on which detection is performed.



(a) Set of positive respondent features.

(b) The corresponding mixture of Gaussians.

Fig. 6. Hand localisation: on the left the blue dots represent the position of the SURF features positively classified by the weak learners. On the right, the mixture of Gaussians centred on such points ($\sigma = 10pix$).

Figure 7 shows some localisation results. As can be seen detection tends to be less precise when the hand is only partially visible in the image - in some of these cases, the MIL algorithm can still detect the presence of the hand in the image, but localisation subsequently fails. In our experiments using the **Autonomous** sequence, the detected point lies on the hand in 98.4% of all positive frames. In these cases the mean error distance between the detected point and the centre of the hand (as visually estimated by a human observer) is 14 pixels with a standard deviation of 7 pixels; the apparent linear size of the hand varies between 60 and 100 pixels according to the image. A short video of the localisation is included in the supplementary material.

VI. Conclusions

We presented an on-line MIL algorithm based on a variant of Adaboost. Our algorithm tackles the MIL problem at the level of the weak learners and includes a mechanism for

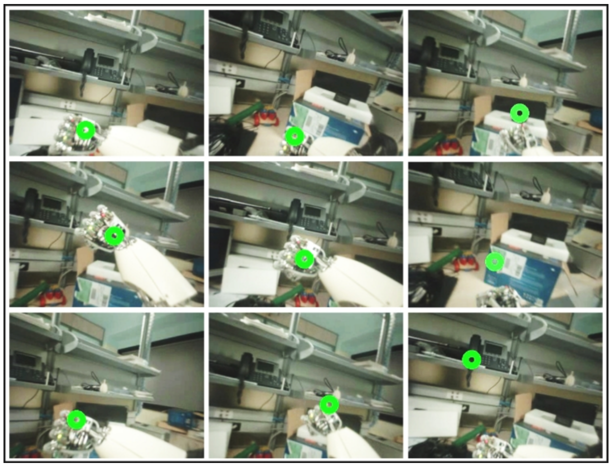


Fig. 7. Examples of hand localisation (**Autonomous** sequence): the green dot represents the detected hand position. The rightmost column shows misdetections.

online feature selection. We validated our algorithm with an application to the problem of hand detection and localisation with a humanoid robot, using SURF descriptors to encode the salient points of the visual scene. We implemented the algorithm on an iCub platform, that we used to run experiments in realistic conditions. These showed that online MIL boost performs consistently well even in a cluttered visual environment, eventually reaching a detection accuracy comparable with the equivalent off-line algorithm.

The reduced demands of MIL algorithms in terms of the type of supervision provided to the system allowed the robot to generate the supervision signal autonomously by labelling images based on the co-occurrence of visual motion and motor activity. Even in this case the system behaved reliably, in spite of the inevitable inclusion of erroneous labels in the training set.

As we showed, the salient feature vectors selected in the process of learning the strong MIL classifier can be used in a natural way to achieve reliable localisation of the hand. In the end, the robot was able to learn autonomously not only to detect, but also to locate its own hand within the images. We remark that nowhere in the training process is the location of the hand in the images specified or otherwise derived from other cues.

It is important to point out that knowledge of the robot kinematics can improve the hand localisation by providing important cues on its location and orientation. In this paper we have purposely avoided using this information. Our goal was to solve the problem without relying on calibration procedures or precise knowledge on the robot kinematics. As a result the technique proposed in the paper is quite general and can be easily adapted to any robot. Of course the localisation could be greatly improved if the learning algorithm incorporated in the detection the current configuration of the arm obtained from the motor encoders. This could help the system to better discriminate different views of the hand (something we did not investigate in this paper).

To the best of our knowledge, this is the first application of online MIL in robotics. In this paper we tested our algorithm on the specific problems of hand detection and localisation; however, the MIL nature of the approach opens up a wide range of applications in cognitive robotics. In our experiments labelling was autonomously generated by detecting co-occurrence of events in different sensory systems (vision and motor system); the same algorithm could in principle be applied in contexts where supervision is provided by integrating sensory modalities like vision, touch and sound (or speech). Also, the flexibility of the boosting algorithm makes it easy to integrate different features in the classification process.

References

[1] The icub web site. `http://www.icub.org`.

[2] S. Andrews and T. Hofmann. Multiple instance learning via disjunctive programming boosting. In *Advances in Neural Information Processing Systems*, volume 16, 2004.

[3] P. Auer. On learning from multiple-instance examples: empirical evaluation of a theoretical approach. In *NeuroCOLT Technical Report Series, NC-TR-97-025*, March 1997.

[4] P. Auer and R. Ortner. A boosting approach to multiple-instance learning. In *Proceedings of the 15th European Conference on Machine Learning, Pisa (Italy)*, pages 63–74, 2004.

[5] B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009.

[6] J. Bauer, N. Sunderhauf, and P. Protzel. Comparing several implementations of two recently published feature detectors. In *International Conference on Intelligent and Autonomous Systems (IAV), Toulouse (France)*, 2007.

[7] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *9th European Conference on Computer Vision*, Graz Austria, May 2006.

[8] J. K. Bradley and R. E. Schapire. Filterboost: regression and classification on lager datasets. In *Advances in Neural Information Processing Systems*, 2008.

[9] Y. Chen, J. Bi, and J. Z. Wang. MILES: Multiple instance learning via embedded instance selection. *IEEE Transactions on PAMI*, 28(12):1931–1947, 2006.

[10] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004.

[11] S. Chiu. Fuzzy model identification based on cluster estimation. *Journal of intelligent and fuzzy systems*, 2:267–278, 1994.

[12] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[13] C. Domingo and O. Watanabe. Madaboost: a modification of adaboost. In *Proc. of the 13th Annual Conference on Computational Learning Theory*, pages 180–189, 2000.

[14] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis*. John Wiley and Sons, 1973.

[15] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Science*, 55(1), 1997.

[16] K. Gold and B. Scassellati. Using probabilistic reasoning over time to self-recognize. *Robotics and Autonomous Systems*, 57(4):384–392, 2009.

[17] H. Grabner and H. Bischof. On-line boosting and vision. In *Proceedings of CVPR*, volume 1, pages 260–267, 2006.

[18] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, pages 234–247, 2008.

[19] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12:651–670, 1996.

[20] C. Kemp and A. Edsinger. What can I control?: The development of visual categories for a robot's body and the world that it influences. In *International Conference on Epigenetic Robotics*, Paris, France, September 2006.

[21] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.

[22] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems*, volume 10, pages 570–576, 1998.

[23] 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, NIST*, Washington DC, USA, August 19-21 2008.

[24] L. Natale, F. Orabona, G. Metta, and G. Sandini. Sensorimotor coordination in a baby robot: learning about objects through grasping. In C. V. Hofsten and K. Rosander, editors, *Progress in Brain Research, From Action to Cognition*, volume 164, pages 403–424. Elsevier, 2007.

[25] N. C. Oza and S. Russell. Online bagging and boosting. In *Proceedings of the 8th International Workshop on Artificial Intelligence and Statistics*, pages 105–112, January 2001.

[26] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1668–1674, Taipei, Taiwan, 2010.

[27] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *Advances in Neural Information Processing Systems*, volume 18, 2005.

[28] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. On-line semi-supervised multiple-instance boosting. In *Proceedings of CVPR*, pages 1879–1886, 2010.