Object Tracking with Adaptive HOG Detector and Adaptive Rao-Blackwellised Particle Filter

Stefano Rosa, Marco Paleari, Paolo Ariano [a], and Basilio Bona [b]

[a]IIT Italian Institute of Technology
Center for Space Human Robotics, Turin, Italy;
[b]Dipartimento di Automatica e Informatica, Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Turin, Italy

ABSTRACT

Scenarios for a manned mission to the Moon or Mars call for astronaut teams to be accompanied by semi-autonomous robots. A prerequisite for human–robot interaction is the capability of successfully tracking humans and objects in the environment. In this paper we present a system for real-time visual object tracking in 2D images for mobile robotic systems. The proposed algorithm is able to specialize to individual objects and to adapt to substantial changes in illumination and object appearance during tracking. The algorithm is composed by two main blocks: a detector based on *Histogram of Oriented Gradient* (HOG) descriptors and linear Support Vector Machines (SVM), and a tracker which is implemented by an adaptive *Rao-Blackwellised particle filter* (RBPF). The SVM is re–trained online on new samples taken from previous predicted positions. We use the *effective sample size* to decide when the classifier needs to be re-trained. Position hypotheses for the tracked object are the result of a clustering procedure applied on the set of particles. The algorithm has been tested on challenging video sequences presenting strong changes in object appearance, illumination, and occlusion. Experimental tests show that the presented method is able to achieve near real–time performances with a precision of about 7 pixels on standard video sequences of dimensions 320 x 240.

Keywords: People Tracking, HOG, Particle Filter, Space, Robotics

1. INTRODUCTION

NASA recently announced programmed efforts on sending cosmonauts to the Moon and Mars (,[1,2]). Scenarios for a manned mission to another planet call for astronaut extravehicular teams to be accompanied by semi-autonomous rovers. These robots must be able to safely follow the astronauts with minimal assistance and to support the EVA crew during his sortie in order to accomplish a variety of tasks. For such applications it is mandatory to successfully tracking humans and objects in the environment. Many object tracking approaches have been proposed in recent years.[3] Human detection is a particular case of object detection and it requires a robust people detector.

There are in general two different approaches in people detection: probability based detection and sliding window techniques. In the first, mostly part detectors are fused together and a model of how these parts can be configured to one another leads to a probability whether a human is present or not. This is especially useful for scenes where pedestrians are highly occluded. For the latter, a fixed-size detection window is moved over an image and at each position a classifier decides, whether an object is present or not. To detect different sizes, the image is step by step resized and the same detection method is used. In this work, the focus lies on sliding window techniques. A very good overview of different methods and detection results can be found in.[4]

Many different approaches have been proposed based on different vision sensors. For stereo and time-of-flight cameras different approaches have been proposed (,[5],[6,7]). For monocular cameras example different algorithms exist, such as histograms of oriented gradients,[8] which are based on the analysis of gradients in the image, or part detectors, which try to detect different body parts of the person (,[9,10]). While gradient based detectors are easier

Further author information: Stefano Rosa, Marco Paleari, Paolo Ariano: E-mail: {name.surname}@iit.it
Basilio Bona: E-mail: basilio.bona@polito.it

to manage and can be easily optimized, multi-part detectors have the ability to detect objects even in presence of partial occlusions; on the other hand they require an accurate construction of models for the different body parts. Other approaches are based on sensor fusion, which has been shown to improve the robustness of people detection.[11] Due to recent advances in GPUs, real-time people detection have been proposed (see e.g.:,[12,13]). An adaptive approach has been proposed in.[14] The detector is based on an adaptive classifier using Haar-like features, Gentle AdaBoost for training, and Condensation algorithm for tracking.

In this paper we present a system for visual object tracking in 2D images for mobile robotic systems. The proposed algorithm is able to specialize to individual objects and to adapt to substantial changes in object appearence during tracking.

2. THE APPROACH

The algorithm is composed by two main parts: the detection part is based on Histogram of Oriented Gradient (HOG) descriptors[8] and support vector machines, while the tracking part is carried out by an adaptive Rao-Blackwellised particle filter (RBPF).[15]

The detector is coupled with the particle filter. Detections are used for the update of the filter and poses predicted by the filter are used in order to acquire new samples for the training of the classifier. Moreover the state of the particle filter is used to decide when a new training of the classifier is needed.

At each time only a certain region of interest is scanned for detections. This allows to considerably reduce detection time, which is the most expensive part in terms of processing time. The region of interest is updated at every time step based on previous detections.

Figure 1 illustrates the various parts of the algorithm in action on a single frame taken form a sequence of images. The red rectangle is the output from the adaptive detector; the yellow rectangle represents the current region of interest; The blue dots represent the particles; the brighter ones represent particles with an high weight. The yellow circle represents the position hypothesis with the highest weight, as explained in Section 2.3.

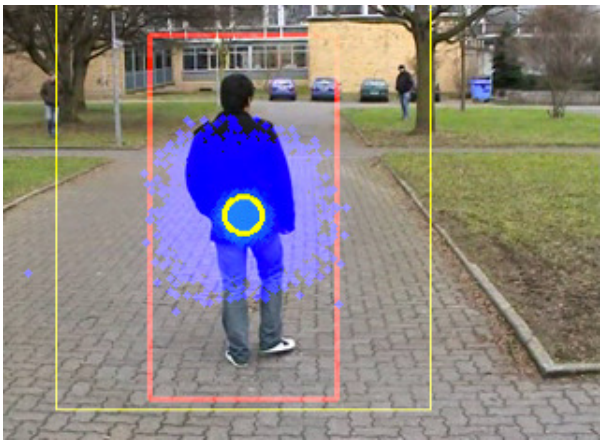


Figure 1. The algorithm in action

2.1 Detector

The detection part is based on HOG features and an svm classifier, which is trained online over a sliding window of samples taken from previous images on the basis of previous detections. In oder to reduce the probability of introducing a bias in the classificator, a subset of older samples is always preserved and introduced in the training.

The idea of *historgrams of oriented gradients* based detection can briefly be explained as following: first, calculate the gradient of an image and then divide the image into cells. For each cell, a histogram of the orientation of the gradient is calculated. Now a detection window is moved over the image and at each location

the histograms of each cell contained in the search window is combined to a feature vector. This feature vector is then fed into a support vector machine for classification. This is repeated for different scales of the image. Details can be found in.[8]

The HOG descriptor has a few advantages over other descriptor methods. It proved to be more invariant to changes in illumination and shadowing than other features. Moreover it upholds invariance to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions.

The classifier is trained only when the variance of the weights of the particles N_{eff} increases over a given threshold. This can be taken as a coarse measure of how new detections do not help to identify unlikely hypotheses any more. In this case the classifier needs to be trained again on newer samples.

In order to improve the tracking of objects that are moving away from the camera or the detection of object in small resolution videos, we implemented a simple rule: if the last detected window is close to the minimum detectable height of the classifier, than the region of interest is magnified (digital zoom). There is one threshold parameter (default is 0) in the HOG algorithm that allows to vary the accuracy of detection. For higher positive values of the parameter, less false positives, but more false negatives are produced. For negative values more false positives but also less false negatives are produced. Therefore, if no prior detection has occurred the threshold is set to an higher value(0.2 on our case) to avoid false detections and otherwise lower (0.2 on our case) to avoid false negatives.

In order to improve the performances of the detector, at every time step only a small area around the last detection, a region of interest is fed into the detector. This can be done only under the assumption that the tracked object does not move very fast, but we found that the assumption holds for most kind of objects to be tracked. This simple expedient can provide frame rates up to 10-25 FPS depending on the size of the tracked object in the image.

2.2 Training

For the training of the svm we use the approach described in.[16] The approach is composed of several steps: first, a HOG feature vector is calculated for every positive training image. At the same time 10 random windows for each negative training image are selected and HOG features are calculated. With this training set, the svm training algorithm of the SVM light library[17] is used in order to obtain a suitable detector. Since SVM light only gives the support vectors v_i and their corresponding α_i, and the HOG algorithm from the OpenCV library needs a weight vector w, we simply calculate it with

$$w = \sum \alpha_i v_i$$

For bootstrapping our detector in the case of people detection we train our classifier with the INRIA Person Dataset;[18] in the case of object detection we use the Pascal dataset.[16]

The training is done over a sliding window of samples taken from previous images on the basis of previous detections. Every k_1 frames a rectangular area is taken from the current frame and put into a subset S_1 of samples. The position of the area corresponds to the current object position hypothesis with the highest weight from the particle filter (see Section 2.3 for details). The dimensions of the area are the same as the latest detection window, as we can assume that the tracked object will not have big changes is dimensions between subsequent frames. Every $k_2 > k_1$ frames the rectangular area is put into another subset S_2 of samples. This subset is used in order to include also old samples in the training, thus preventing the problem of overfitting in most cases.

When the dimensions of the subset S_1 is above a given threshold the classifier is re–trained on the current dataset, and the resulting weight vector w is fed to the classifier.

The state of the particle filter is also used to decide when a new training of the classifier is needed. When the effective sample size N_{eff} falls below a given threshold the classifier is re–trained on the current dataset, and the resulting weight vector is used instead of the older one.

2.3 Tracker

For the tracking of the detected object we use a particle filter implementation. Particle filters are sequential Monte Carlo methods based on point mass representations of probability densities. The first advantage of this method over classic Kalman filters is the ability to cope with non-linearity and non-gaussianity, which are critical in the case of a moving object. Another important aspect is the possibility to model multi-modal distributions.

In our approach the prediction phase is based on a simple linear motion motion model. The estimate of the new state of each particle is a linear extrapolation of the previous state plus Gaussian noise. We chose a simple motion model because for people and object tracking the advantage of using a stochastic model is not prominent compared to simpler motion models.[19]

The update phase is based on object detections. Particles are weighted based on the distance from detection hypotheses provided by the adaptive HOG detector.

The resampling phase is based on *Kullback-Leibler divergence* (KLD). The number of particles representing the belief on the object pose at each step is adaptive, so that only the number of particles sufficient to represent the belief distribution is used.

Particle filters address the problem of estimating the state x of a dynamical system from measurements z. The goal of particle filters is to estimate a posterior probability density over the state space conditioned on the data collected so far. Let the belief $Bel(x_t)$ be the posterior at time t. Particle filters represent this belief by a set S_t of n weighted samples distributed according to $Bel(x_t)$.

$$S_t = \{\langle x_t^{(i)}, w_t^{(i)}\rangle | i = 1, .., n\}$$

where each $x_t^{(i)}$ is a sample and the $w_t^{(i)}$ are non-negative numerical factors called importance weights, which sum up to one.

We implement the adaptive particle filter approach proposed in[15] for robot localization. Since the approach is generic it can also be adopted for the particular case of object tracking. In this approach the key-point is to bound the error introduced by the sample-based representation of the particle filter. The underlying assumption is that the true posterior is given by a discrete, piecewise constant distribution such as a discrete density tree or a multi-dimensional histogram. Under this assumption we can determine the number of samples so that the distance between the Maximum Likelihood Estimate based on the samples and the true posterior does not exceed a pre-specified threshold ϵ. The distance between the Maximum Likelihood Estimate and the true distribution is measured by the Kullback-Leibler distance. The number of particles at each step i can therefore be set to

$$n_i = \frac{1}{2\epsilon}\chi^2_{k-1,1-\delta}$$

where $\chi^2_{k-1,1-\delta}$ is a chi-square distribution with $1-k$ degrees of freedom. This value is the required number of particles to guarantee that with probability $1-\delta$ the Kullback-Leibler distance between the Maximum Likelihood Estimate of the position hypothesis and the true distribution is less than ϵ.

This is a clear advantage in terms of both memory occupation and computational resources.

Moreover we use the *effective sample size* N_{eff} as a measure of how well the current set of particles represents the true posterior of the object pose, by measuring the variance on the particles weight. If N_{eff} stays constant the new information does not help to identify unlikely hypotheses represented by the individual particles. In that case, the variance in the importance weights of the particles does not change over time. If, in contrast, the value of N_{eff} decreases over time, the new information can be used to identify that some particles are less likely than others.

The position hypotheses for the tracked object at each step are the result of a *Density-Tree* clustering procedure applied on the set of particles.

3. EXPERIMENTAL TESTS

In this section we present an evaluation of our algorithm on several video sequences and we compare the performances of our adaptive approach with other non-adaptive approaches.

In our implementation we used *OpenCV* libraries for the implementation of the detector. The algorithm was implemented in C++ and all the tests were conducted on a standard PC equipped with a 2.4 Ghz CPU and 2 Gb of RAM.

The algorithm has been tested on several video sequences from the *BoBoT* dataset (Bonn Benchmark on Tracking),[20] as well as on video sequences presenting strong changes in object appearence, illumination and occlusion. We show that the method is able to achieve a frame rate up to 20 fps on 320x240 video sequences on a standard PC.

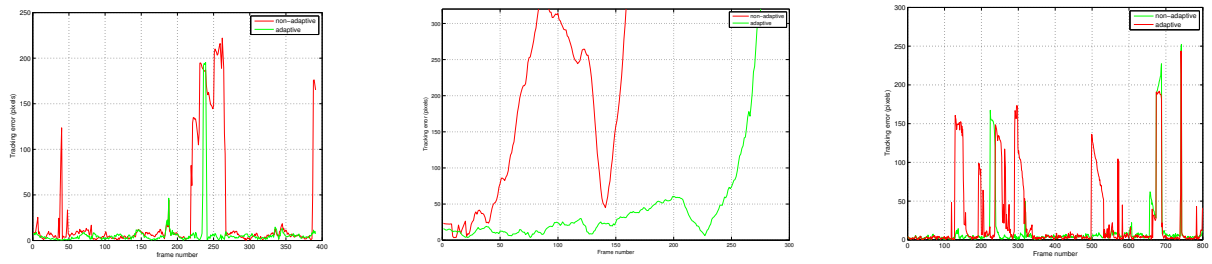
In Experiment 1 we show a comparison of two different algorithms on some image sequences taken from the *BoBoT* dataset. This dataset contains both people and objects, and each video presents some challenging condition for detection and tracking. We also compare our algorithm with the results presented in[14] for three different sequences.

In Experiment 2 we show how the detection time and the number of particles changes over time.

3.1 Experiment 1

In Figure 3.1 we show a comparison of two different algorithms on some image sequences taken from the *BoBoT* dataset (*SeqA*, *SeqB*, *SeqI*). The first algorithm is a non-adaptive approach based on HOG features and particle filters. The second approach is the proposed approach with adaptive HOG detector and adaptive particle filters.

The ground-truth data which is available in this dataset is in the form of rectangular shapes surrounding the object to be tracked. Since in our approach the detection window is of fixed size, we do not use this ground-truth directly. We calculate instead for each frame the center of the detected object and we measure the error between this center and the output of our particle filter S_t.



(a) SeqA (b) SeqB (c) SeqI

Figure 2. Tracking error for three different sequences

Sequence *SeqA* presents significative changes in appearence and fast movements of the tracked object; sequence *SeqB* presents strong changes in background appearence; sequence *SeqI* is an example of people tracking with many occlusions from crossing pedestrians.

We can see how the adaptive approach significantly outperforms the non-adaptive approach in every image sequence. In particular the adaptive method is able to recover more quickly from detection errors.

In Table 1 we present a comparison between 6 different algorithms. The first one is a simple tracking algorithm based on color histograms, the second one is based on multi-component tracking, the next three are different versions of the approach proposed in[14] and the last column contains the results form our adaptive algorithm. For every approach and for every sequence the mean score over the sequence is reported (calculated as suggested in[20]). Since our detection window is of fixed size, the results of our algorithm are generally better than the reported score. We show that the results are comparable with the results presented in.[14]

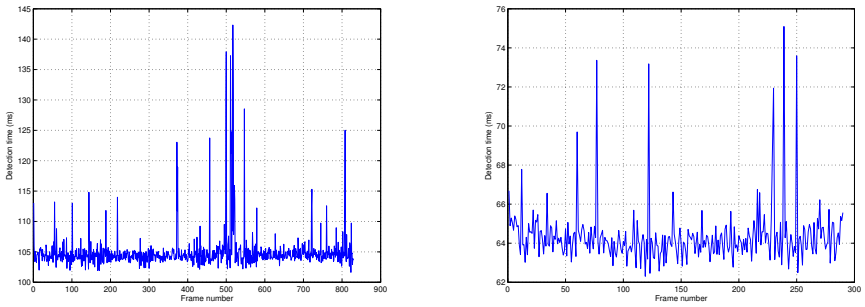
Table 1. Comparison of six methods

Seq.	Histogram	Multi-Comp.	non-adaptive H.-cs	adaptive H.-cs	adapt. p. part. H.-cs	Adaptive HOG
A	70.73	63.24	30.35	65.06	59.35	70.16
B	67.02	50.73	6.02	79.01	77.38	55.10
I	68.94	47.63	48.97	75.02	56.33	64.81

3.2 Experiment 2

In Figure 3(a) we show the detection time for each frame of sequence *SeqI* from the *BoBoT* dataset. The dimensions of the video sequence are 320 x 240 and the HOG detection window is 64 x 128. In Figure 3(b) we show the detection times for sequence *SeqB*. The dimensions of the video sequence are 320 x 240 and the HOG detection window is 64 x 64.

We show that by using a variable region of interest around the tracked object an high frame rate can be achieved. The peaks in detection time correspond to the time instants when the region of interests grows bigger or the full image is scanned.



(a) SeqI (b) SeqB

Figure 3. Detection time for two sequences

In figure 4 we show how the number of particles changes over time for a single run. The test video sequence is *SeqI* and the maximum number of particles was set to 5000. We also set a lower bound of 1000 to the number of particles, in order to avoid the convergence of the particle filter to a wrong position hypothesis in the case of bad detections. We can see how the number of particles drops and in some cases could be lower than the lower bound, without any significant downgrade in tracking performances.

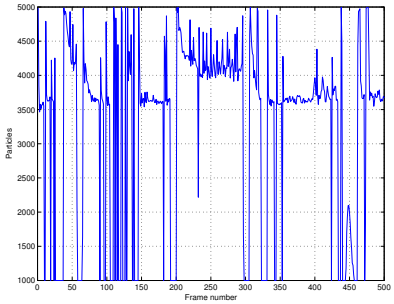


Figure 4. Number of particles over time

4. CONCLUSIONS

We presented an algorithm for real-time visual object tracking in 2D images for mobile robotic systems. The algorithm is composed by two main blocks: a detector based on *Histogram of Oriented Gradient* descriptors and linear Support Vector Machines, and a tracker which is implemented by an adaptive *Rao-Blackwellised particle filter*.

The algorithm has been tested on challenging video sequences presenting strong changes in object appearance, illumination, and occlusion. Experimental tests show that the presented method is able to achieve near real–time performances with a good precision on standard video sequences.

Future work will include the use of a GPU implementation of HOG in order to speed up the computation of the HOG features, and a per-particle svm classifier. Moreover experimental tests will be carried out on real rovers.

REFERENCES

[1] Chien, S., Doyle, R., Davies, A., Jonsson, A., and Lorenz, R., "The future of ai in space," *Intelligent Systems, IEEE* **21**(4), 64 –69 (2006).

[2] Fong, T. W. and Nourbakhsh, I., "Interaction challenges in human-robot space exploration," *Interactions* **12**, 42–45 (March 2005).

[3] Yilmaz, A., Javed, O., and Shah, M., "Object tracking: A survey," *ACM Comput. Surv.* **38** (December 2006).

[4] Wojek, C. and Schiele, B., "A performance evaluation of single and multi-feature people detection," in [*Pattern Recognition*], Rigoll, G., ed., *Lecture Notes in Computer Science* **5096**, 82–91, Springer Berlin - Heidelberg (2008).

[5] Muoz-Salinas, R., Aguirre, E., Garca-Silvente, M., and Gonzalez, A., "People detection and tracking through stereo vision for human-robot interaction," in [*MICAI 2005: Advances in Artificial Intelligence*], Gelbukh, A., de Albornoz, ., and Terashima-Marn, H., eds., *Lecture Notes in Computer Science* **3789**, 337–346, Springer Berlin / Heidelberg (2005).

[6] Mndez-Polanco, J., Muoz-Melndez, A., and Morales, E., "People detection by a mobile robot using stereo vision in dynamic indoor environments," in [*MICAI 2009: Advances in Artificial Intelligence*], Aguirre, A., Borja, R., and Garci, C., eds., *Lecture Notes in Computer Science* **5845**, 349–359, Springer Berlin / Heidelberg (2009).

[7] Abd-Almageed, W., Hussein, M., and Abdelkader, M., "Real-time human detection and tracking from mobile vehicles," in [*Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*], 149 –154 (2007).

[8] Dalal, N. and Triggs, B., "Histograms of oriented gradients for human detection," in [*Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*], (2005).

[9] Felzenszwalb, P., Girshick, R., and McAllester, D., "Cascade object detection with deformable part models," in [*Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*], 2241 –2248 (2010).

[10] Corvee, E. and Bremond, F., "Body parts detection for people tracking using trees of histogram of oriented gradient descriptors," in [*Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*], 469 –475 (2010).

[11] Schiele, B., Andriluka, M., Majer, N., Roth, S., and Wojek, C., "Visual people detection: Different models, comparison and discussion," in [*Proceedings of the IEEE ICRA 2009 Workshop on People Detection and Tracking*], 1–8 (2009).

[12] Bilgic, B., Horn, B., and Masaki, I., "Fast human detection with cascaded ensembles on the gpu," in [*Intelligent Vehicles Symposium (IV), 2010 IEEE*], (2010).

[13] Prisacariu, V. A. and Reid, I., "fasthog- a real-time gpu implementation of hog technical report no. 2310/09," (2009).

[14] Klein, D., Schulz, D., Frintrop, S., and Cremers, A., "Adaptive real-time video-tracking for arbitrary objects," in [*Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*], 772 –777 (oct. 2010).

[15] Fox, D., "Kld-sampling: Adaptive particle filters," in [*In Advances in Neural Information Processing Systems 14*], 713–720, MIT Press (2001).

[16] "The pascal object recognition database collection." Website. `http://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html`.

[17] "Svm-light support vector machine." Website. `http://svmlight.joachims.org/`.

[18] "Inria person dataset." Website. `http://pascal.inrialpes.fr/data/human/`.

[19] Pellegrini, S., Ess, A., and Gool, L., "Predicting pedestrian trajectories," in [*Visual Analysis of Humans*], Moeslund, T. B., Hilton, A., Krger, V., and Sigal, L., eds., 473–491, Springer London (2011).

[20] "Bobot bonn benchmark on tracking." Website. `http://www.iai.uni-bonn.de/~kleind/tracking/`.