Experimental Assessment of Probabilistic Integrated Object Recognition and Tracking Methods

Francesc Serratosa[1], Nicolás Amézquita[1], and René Alquézar[2]

[1] Universitat Rovira i Virgili
Av. Països Catalans 26, 43007 Tarragona, Spain
[2] Inst. Robòtica i Informàtica Industrial
CSIC-UPC, Llorens Artigas 4-6, 08028 Barcelona, Spain
francesc.serratosa@urv.cat, nicolas.amezquita@urv.cat,
ralquezar@iri.upc.edu

Abstract. This paper presents a comparison of two classifiers that are used as a first step within a probabilistic object recognition and tracking framework called PIORT. This first step is a static recognition module that provides class probabilities for each pixel of the image from a set of local features. One of the implemented classifiers is a Bayesian method based on maximum likelihood and the other one is based on a neural network. The experimental results show that, on one hand, both classifiers (although they are very different approaches) yield a similar performance when they are integrated within the tracking framework. And on the other hand, our object recognition and tracking framework obtains good results when compared to other published tracking methods in video sequences taken with a moving camera and including total and partial occlusions of the tracked object.

Keywords: Object tracking, object recognition, occlusion, performance evaluation.

1 Introduction

The first important issue while dealing with object locating and tracking is to determine the type of object model to learn, which usually depends on the application environment. In our case, we want a mobile robot equipped with a camera to locate and track general objects (people, other robots, wastepaper bins…) in both indoor and outdoor environments.

On one hand, a useful model should be relatively simple and easy to acquire from the result of image processing steps. For instance, the result of a color image segmentation process, consisting of a set of regions or spots, characterized by simple features related to color, may be a good starting point to learn the model. Hence, we have decided to represent an object just as an unstructured set of pixels.

On the other hand, we want the system to have the capacity of determining occlusions and re-emergencies of tracked objects. Various approaches that analyze occlusion situations have been proposed. The most common one is based on background subtraction [1]. Although this method is reliable, yet it only works with a fixed camera and a known background, which is not our case. Other approaches are based on

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 817–824, 2009.

examining the measurement error for each pixel [2, 3]. The pixels that their measurement error exceeds a certain value are considered to be occluded. These methods are not very appropriate in outdoor scenarios, where the variability of the pixel values between adjacent frames may be high. Finally, contextual information is exploited in [4, 5], but in these approaches, there is a need of knowing a priori the surroundings of the mobile robot.

This paper presents a comparison of two possible alternative classifiers to deal with the first step of a previously reported approach for integrated object recognition and tracking [6, 7]. These are a simple Bayesian method and a neural net based method, both providing posterior class probabilities for each pixel of the images.

The rest of the paper is organized as follows. A summary of our probabilistic framework for object recognition and tracking is given in Section 2. The methods used for the static recognition module are described in Section 3. Experimental results are presented in Section 4. Finally, conclusions are drawn in Section 5.

2 A Probabilistic Framework for Object Recognition and Tracking

Let us assume that we have a sequence of 2D color images $I^t(x,y)$ for $t=1,\ldots,L$, and that there are a maximum of N objects of interest in the sequence of different types (associated with classes $c=1,\ldots,N$), and that a special class $c=N+1$ is reserved for the background. Hence, we would like to obtain N sequences of binary images $T_c^t(x,y)$, that mark the pixels belonging to each object in each image; these images are the desired output of the whole process and can also be regarded as the output of a tracking process for each object.

In our PIORT (Probabilistic Integrated Object Recognition and Tracking) framework [6, 7], we divide the system in three modules. The first one performs object recognition in the current frame (**static recognition**) and stores the results in the form of probability images (one probability image per class) $Q_c^t(x,y)$, that represent for each pixel the probabilities of belonging to each one of the objects of interest or to the background, according only to the information in the current frame (see Section 3). In the second module (**dynamic recognition**), the results of the first module are used to update a second set of probability images, $p_c^t(x,y)$, with a meaning similar to that of $Q_c^t(x,y)$ but now taking into account as well both the recognition and tracking results in the previous frames through a dynamic iterative rule. Finally, in the third module (**tracking decision**), tracking binary images $T_c^t(x,y)$ are determined for each object from the current dynamic recognition probabilities, the previous tracking image of the same object and some other data, which contribute to provide a prediction of the object's apparent motion in terms of translation and scale changes. See [6] for a detailed description of the second and third modules and [7] for an extension of the tracking decision module.

3 Static Recognition Module

The static recognition module in our PIORT framework is based on the use of a classifier that is trained from examples and provides posterior class probabilities for each pixel from a set of local features. The local features to be used may be chosen in many different ways. A possible approach consists of first segmenting the given input image $I^{t}(x,y)$ in homogeneous regions (or spots) and computing some features for each region that are afterwards shared by all its constituent pixels. Hence, the class probabilities $Q_c^t(x,y)$ are actually computed by the classifier once for each spot in the segmented image and then replicated for all the pixels in the spot. For instance, RGB colour averages can be extracted for each spot after colour segmentation and used as feature vector $v(x,y)$ for a classifier. In the next two subsections we present two specific classifiers that have been implemented and tested within the PIORT framework using this type of information.

3.1 A Simple Bayesian Method Based on Maximum Likelihood and Background Uniform Conditional Probability

Let c be an identifier of a class (between 1 and N+1), let B denote the special class c=N+1 reserved for the background, let k be an identifier of an object (non-background) class between 1 and N, and let v represent the value of a feature vector. Bayes theorem establishes that the posterior class probabilities can be computed as

$$P(c \mid v) = \frac{P(v \mid c)P(c)}{P(v)} = \frac{P(v \mid c)P(c)}{P(v \mid B)P(B) + \sum_{k=1}^{N} P(v \mid k)P(k)} \tag{1}$$

Our simple Bayesian method for static recognition is based on imposing the two following assumptions:

a) equal priors: all classes, including B, will have the same prior probability, i.e. $P(B)=1/(N+1)$ and $P(K)=1/(N+1)$ for all k between 1 and N.
b) a uniform conditional probability for the background class, i.e. $P(v|B)=1/M$, where M is the number of values (bins) in which the feature vector v is discretized.

Note that the former assumption is that of a maximum likelihood classifier, whereas the latter assumes no knowledge about the background. After imposing these conditions, equation (1) turns into

$$P(c \mid v) = \frac{P(v \mid c)}{\frac{1}{M} + \sum_{k=1}^{N} P(v \mid k)} \tag{2}$$

and this gives the posterior class probabilities we assign to the static probability images, i.e. $Q_c^t(x,y) = P(c \mid v(x,y))$ for each pixel (x,y) and time t.

It only remains to set a suitable M constant and to estimate the class conditional probabilities $P(v \mid k)$ for all k between 1 and N (object classes). To this end, class histograms H_k are set up using the labelled training data and updated on-line afterwards using the tracking results in the test data.

For constructing the histograms, let $v(x,y)$ be the feature vector consisting of the original RGB values of a pixel (x,y) labelled as belonging to class k. We uniformly discretize each of the R, G and B channels in 16 levels, so that M =16×16×16= 4096. Let b be the bin in which $v(x,y)$ is mapped by this discretization. To reduce discretization effects, a smoothing technique is applied when accumulating counts in the histogram as follows:

$$\begin{aligned} &H_k(b) := H_k(b) + (10 - \#neighbors(b)) \\ &H_k(b') := H_k(b') + 1 \quad \text{if } b' \text{ is a neighbor of } b \end{aligned} \tag{3}$$

where the number of neighbors of b (using non-diagonal connectivity) varies from 3 to 6, depending on the position of b in the RGB space. Hence, the total count C_k of the histogram is increased by ten (instead of one) each time a pixel is counted and the conditional probability is estimated as $P(v \mid k) = H_k(b) / C_k$ where b is the bin corresponding to v. The above smoothing technique is also applied when updating the histogram from the tracking results; in that case the RGB value $v(x,y)$ in the input image $I^t(x,y)$ of a pixel (x,y) is used to update the histogram H_k (and the associated count C_k) if and only if $T_k^t(x,y)$=1.

3.2 A Neural Net Based Method

In this method, a neural net classifier (a multilayer perceptron) is trained off-line from the labelled training data. The RGB colour averages extracted for each spot after colour segmentation are used as feature vector $v(x,y)$ and supplied as input to the network in both training and test phases. To the contrary of the Bayesian method described previously, training data for the background class are also provided by selecting some representative background regions in the training image sequence, because the network needs to gather examples for all classes including the background. The network is not retrained on-line using the tracking results in the test phase (this is another difference with respect to the Bayesian method described).

It's well known that using a 1-of-c target coding scheme for the classes, the outputs of a network trained by minimizing a sum-of-squares error function approximate the posterior probabilities of class membership (here, $Q_c^t(x,y)$), conditioned on the input feature vector [8]. Anyway, to guarantee a proper sum to unity of the posterior probabilities, the network outputs (which are always positive values between 0 and 1) are divided by their sum before assigning the posterior probabilities.

4 Experimental Results

We were interested in testing both PIORT approaches in video sequences taken with a moving camera and including object occlusions. To this end, we have used three test sequences with N=1 objects of interest to track, which are available at http://www-iri.upc.es/people/ralqueza/S5.avi, S8.avi and S9.avi, respectively. The first sequence shows an office scene where a blue ball is moving on a table and is temporally occluded, while other blue objects appear in the scene. A similar but different sequence was used for training a neural network to discriminate between blue balls and typical sample regions in the background and for constructing the class histogram of the blue ball (available at http://www-iri.upc.es/people/ralqueza/bluetraining.avi). The second sequence is a long sequence taken on a street where the aim is to track a pedestrian wearing a red jacket and which includes total and partial occlusions of the followed person. In this case, a short sequence of the scene taken with a moving camera located in a different position was used as training sequence (http://www-iri.upc.es/people/ralqueza/redpedestrian_training.avi). The third sequence, S9.avi, is even longer and shows an outdoor scene in which a guy riding a Segway robot and wearing an orange T-shirt is followed; the associated training sequence is at http://www-iri.upc.es/people/ralqueza/T-shirt_training.avi.

All images in the sequences were segmented independently using the EDISON implementation of the mean-shift segmentation algorithm, code available at http://www.caip.rutgers.edu/riul/research/code.html. The local features extracted for each spot of each image were the RGB colour averages of the pixels in that spot. For object learning, spots selected through ROI (region-of-interest) windows in the training sequence were collected to train a two-layer perceptron using backpropagation and to build the target class histogram. When using the neural net in the test phase, the class probabilities for all the spots in the test sequences were estimated from the net outputs. When using the histogram, the spot class probabilities were estimated according to equation (2). In both cases, the spot class probabilities were replicated for all the pixels in the same spot. For object tracking in the test sequences, ROI windows for the target object were only marked in the first image to initialise the tracking process.

The results for the test sequences were stored in videos where each frame has a layout of 2 x 3 images with the following contents: the top left is the image segmented by EDISON; the top middle is the image of probabilities given by the static recognition module for the current frame; the top right is the *a priori* prediction of the tracking image; the bottom left is the image of dynamic probabilities; the bottom right is the *a posteriori* binary tracking image (the final result for the frame); and the bottom middle is an intermediate image labelled by the tracking module where yellow pixels correspond to pixels labelled as “certainly belonging to the object”, light blue pixels correspond to pixels initially labelled as “uncertain” but with a high dynamic probability, dark blue pixels correspond to pixels labelled as “uncertain” and with a low probability, dark grey pixels are pixels labelled as “certainly not belonging to the object” but with a high probability and the rest are black pixels with both a low probability and a “certainly not belonging to the object” label. The tracking results videos with this layout are attainable at http://www-iri.upc.es/people/ralqueza/S5_NN.mpg, S5_Bayes.mpg, S8_NN.mpg, S8_Bayes.mpg, S9_NN.mpg and S9_Bayes.mpg.

For comparison purposes, tracking of the target objects in the test sequences was also carried out by applying the six following methods, which only need the ROI window mark in the first frame of the test sequence: Template Match by Correlation, which refers to normalized correlation template matching [9]; Basic Meanshift [10]; Histogram Ratio Shift [11]; Variance Ratio Feature Shift [12]; Peak Difference Feature Shift [12]; and Graph-Cut Based Tracker [13].

From the tracking results of all the tested methods, two evaluation metrics were computed for each frame: the **spatial overlap** and the **centroid distance** [14]. The spatial overlap is defined as the overlapping level $A(GT_k,ST_k)$ between the ground truth GT_k and the system track ST_k in a specific frame k:

$$A(GT_k, ST_k) = \frac{\text{Area}\,(GT_k \cap ST_k)}{\text{Area}\,(GT_k \cup ST_k)} \tag{4}$$

and *Dist*(*GTC*$_k$, *STC*$_k$) refers to the Euclidean distance between the centroids of the ground truth (*GTC*$_k$) and the system track (*STC*$_k$) in frame *k*. Naturally, the larger the overlap and the smaller the distance, the better performance of the system track.

Since the centroid distance can only be computed if both GT_k and ST_k are non-null, a **failure ratio** was measured as the number of frames in which either GT_k or ST_k was null (but not both) divided by the total number of frames. Finally, an **accuracy** measure was computed as the number of good matches divided by the total number of frames, where a good match is either a true negative or a true positive with a spatial overlap above a threshold of 0.243 (which is the overlap obtained between two circles of the same size when one of the centers is located in the border of the other circle).

Tables 1, 2 and 3 present the results (mean ± std. deviation) of the two former evaluation measures together with the failure ratio and accuracy of each tracking method for the three tests (best values in bold). Our PIORT tracking methods worked fine in the three test sequences, obtaining the best values of the evaluation measures and outperforming clearly the rest of the methods compared.

Table 1. Tracking performance results on blue ball test sequence (103 frames)

Tracking method	**Spatial Overlap**	**Centroid Distance**	**Failure Ratio**	**Accuracy**
1Template Match by Correlation	0.275 ± 0.481	74.65 ± 91.53	0.192	0.433
2 Basic Meanshift	0.234 ± 0.523	78.40 ± 90.33	0.192	0.365
3 Histogram Ratio Shift	0.155 ± 0.450	125.88 ± 111.80	0.433	0.298
4 Variance Ratio Feature Shift	0.197 ± 0.375	96.72 ± 134.84	0.385	0.596
5 Peak Difference Feature Shift	0.281 ± 0.566	103.60 ± 136.77	0.413	0.587
6 Graph-Cut Based Tracker	0.007 ± 0.287	188.79 ± 118.13	0.750	0.212
7 Our Tracker PIORT-Neural Net	**0.603 ± 0.400**	12.53 ± 59.38	**0.048**	**0.952**
8 Our Tracker PIORT-Bayesian	0.586 ± 0.394	**12.46 ± 59.40**	**0.048**	**0.952**

Table 2. Tracking performance results on pedestrian test sequence (215 frames)

Tracking method	**Spatial Overlap**	**Centroid Distance**	**Failure Ratio**	**Accuracy**
1Template Match by Correlation	0.441 ± 0.307	25.25 ± 61.10	0.066	0.772
2 Basic Meanshift	0.241 ± 0.581	72.08 ± 64.33	0.066	0.336
3 Histogram Ratio Shift	0.354 ± 0.237	13.49 ± 38.27	**0.024**	0.644
4 Variance Ratio Feature Shift	0.453 ± 0.320	34.27 ± 81.13	0.118	0.820
5 Peak Difference Feature Shift	0.503 ± 0.203	11.42 ± 45.11	0.033	0.953
6 Graph-Cut Based Tracker	0.039 ± 0.323	194.7 ± 105.3	0.772	0.161
7 Our Tracker PIORT-Neural Net	**0.790 ± 0.238**	11.90 ± 50.87	0.043	**0.957**
8 Our Tracker PIORT-Bayesian	0.737 ± 0.244	**11.15 ± 48.14**	0.038	0.953

Table 3. Tracking performance results on guy-on-Segway test sequence (297 frames)

Tracking method	**Spatial Overlap**	**Centroid Distance**	**Failure Ratio**	**Accuracy**
1Template Match by Correlation	0.102 ± 0.526	130.3 ± 69.75	**0.003**	0.149
2 Basic Meanshift	0.221 ± 0.126	41.30 ± 58.70	0.010	0.402
3 Histogram Ratio Shift	0.527 ± 0.252	22.83 ± 58.43	0.054	0.861
4 Variance Ratio Feature Shift	0.691 ± 0.249	27.69 ± 75.15	0.101	0.895
5 Peak Difference Feature Shift	0.556 ± 0.207	29.19 ± 74.65	0.101	0.895
6 Graph-Cut Based Tracker	0.136 ± 0.218	101.6 ± 112.7	0.365	0.193
7 Our Tracker PIORT-Neural Net	0.734 ± 0.156	**3.40 ± 14.77**	**0.003**	0.973
8 Our Tracker PIORT-Bayesian	**0.743 ± 0.132**	3.70 ± 14.61	**0.003**	**0.980**

5 Conclusions and Future Work

In this paper, we have compared two static recognition methods which are embedded in a probabilistic framework for object recognition and tracking in video sequences called PIORT. Both methods are based on the use of a classifier that is trained from examples and provides posterior class probabilities for each pixel from a set of local features. The first classifier is based on a maximum likelihood Bayesian method in which the conditional probabilities for object classes are obtained from the information of the class histograms (for discretized RGB values) and a uniform conditional probability is assumed for the background. The second classifier is based on a neural net which is trained with the RGB colour averages extracted for each spot of the segmented images.

Even though the characteristics of these two classifiers are quite different, the recognition and tracking results of PIORT using both approaches were similar in the three test sequences, which means that the good ability of PIORT to track the objects is mostly due to a smart cooperation of the three inner modules and is not very dependent on the specific method used for object recognition. In the experimental comparison with other reported methods for object tracking, PIORT obtained the best results and much better in most of the cases than those of the other methods. However, as observed in some frames of the test sequences, still there are cases where the behaviour of the tracking decision module of PIORT should be improved, particularly in the step of object re-emergence after occlusion and when other objects of similar appearance are next to the target. The upgrade of this tracking module will be subject of future research.

Acknowledgements

This research was partially supported by Consolider Ingenio 2010, project CSD2007-00018, by the CICYT project DPI 2007-61452 and by the Universitat Rovira I Virgili (URV) through a predoctoral research grant.

References

1. Senior, A., et al.: Appearance models for occlusion handling. J. Image Vis. Comput. 24(11), 1233–1243 (2006)
2. Nguyen, H.T., Smeulders, A.W.M.: Fast occluded object tracking by a robust appearance filter. IEEE Trans. Pattern Anal. Mach. Intell. 26(8), 1099–1104 (2004)
3. Zhou, S.K., Chellappa, R., Moghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Trans. Image Process. 13(11), 1491–1506 (2004)
4. Ito, K., Sakane, S.: Robust view-based visual tracking with detection of occlusions. In: Int. Conf. Robotics Automation, vol. 2, pp. 1207–1213 (2001)
5. Hariharakrishnan, K., Schonfeld, D.: Fast object tracking using adaptive block matching. IEEE Trans. Multimedia 7(5), 853–859 (2005)
6. Amézquita Gómez, N., Alquézar, R., Serratosa, F.: Dealing with occlusion in a probabilistic object tracking method. In: IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, Alaska (2008)
7. Alquézar, R., Amézquita Gómez, N., Serratosa, F.: Tracking deformable objects and dealing with same class object occlusion. In: Fourth Int. Conf. on Computer Vision Theory and Applications (VISAPP 2009), Lisboa, Portugal (2009)
8. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
9. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Machine Intell. 25(4), 564–577 (2003)
10. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Machine Intell. 24(5), 603–619 (2002)
11. Collins, R., Zhou, X., The, S.K.: An open source tracking testbed and evaluation web site. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2005 (2005)
12. Collins, R., Liu, Y.: On-line selection of discriminative tracking features. IEEE Trans. Pattern Anal. Machine Intell. 27(10), 1631–1643 (2005)
13. Bugeau, A., Pérez, P.: Track and cut: simultaneous tracking and segmentation of multiple objects with graph cuts. In: Third Int. Conf. on Computer Vision Theory and Applications (VISAPP 2008), Funchal, Madeira, Portugal (2008)
14. Yin, F., Makris, D., Velastin, S.A.: Performance evaluation of object tracking algorithms. In: 10th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance, PETS 2007 (2007)