# IMPROVED PROBABILISTIC INTEGRATED OBJECT RECOGNITION AND TRACKING (PIORT) METHODOLOGY

Ashwin.M. [1]
Assistant Professor, [1]Dept of Computer Science and Engineering,
Adhiyamaan College of Engineering, Anna University,
[1]mailmeashwin@gmail.com

Dinesh Babu.G.[2]
Student, [2]Dept of Computer Science and Enggineering,
Adhiyamaan College of Engineering, Anna University,
[2]dinesh4.g@gmail.com

**Abstract**

A thinning algorithm is used to reduce unnecessary information by peeling objects layer by layer so that the result is sufficient to allow topological analysis. It has several applications, but is particularly useful for skeletonization. A new sequential thinning algorithm is introduced to preserve both the topology and geometry of the object. In sequential thinning, only a single point may be deleted at a time and it always guarantees the preservation of the topology of the original image. It is based on removing the central pixel in the 3x3 neighborhood of the candidate pixel which preserves the topology and geometry. The algorithm is based on computing the local Euler number before and after removing the candidate pixel and then checking whether there is a difference in the computed values. Furthermore, in order to preserve the geometric criteria (preserve end points of the object), we consider change of the boundary length when removing the central pixel.

***Keywords:*** static recognition, Bayesian method, neural net based method, dynamic recognition module, tracking decision module

## 1. Introduction

Recognition and tracking of multiple objects is one of the main challenges in computer vision that currently deserves a lot of attention from researchers. Almost all the reported approaches are very application-dependent and there is a lack of a general methodology for dynamic object recognition and tracking that can be instantiated in particular cases. In this project the work is oriented towards the definition and development of such a methodology which integrates object recognition and tracking from a general perspective using a probabilistic framework called PIORT (probabilistic integrated object recognition and tracking framework).

Object tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around the scene. The estimation of the trajectory of an object is pertinent in the following tasks between others: motion-based recognition (human identity based on gait, automatic object detection), automated surveillance, video indexing, human-computer interaction (gesture recognition, eye gaze tracking), traffic monitoring or vehicle navigation. The process of tracking objects can be very complex and application dependent due to: the projection of the 3D world to a 2D image, noise in images, complex object motion, non-rigid or articulated objects, partial and full object occlusions, complex object shapes, scene illumination changes or real-time processing requirements. Nevertheless, in most of the methods presented elsewhere, the tracking process is simplified by imposing constraints on the motion or appearance of objects. For example, almost all tracking algorithms assume that the object motion is smooth with no abrupt changes. Another usual simplification is to have prior knowledge of about the number and size of the objects or the object appearance and shape.

Numerous approaches for object tracking have been proposed. These primarily differ from each other based on the way they approach the three following questions:

- Which object representation is suitable for the tracking?

- Which image features should be used?
- How should the motion, appearance and shape of the object be modeled?

The answers to these questions depend on the context in which the tracking is performed and the use for which the tracking information is being sought. A large number of tracking methods have been proposed which attempt to answer these questions for a variety of scenarios.

Objects can be represented by their shapes and appearances. We first describe the object shape representations commonly employed for tracking. After that, we address the appearance representations and we finish with the joint shape and appearance representations.

**2. Static recognition**

In the static models for object detection, the only information considered is the current frame. The most common models are:

- **Interest points**: Interest points in the images are the pixels that have an expressive texture in their respective localities. They have been long used in the context of motion, stereo and tracking problems. A desirable quality of an interest point is its invariance to changes in illumination and camera viewpoint. For a comparative evaluation of interest point detectors, we refer the reader to the survey [Mikolajczyk, 2003].
- **Segmentation**: The aim of the segmentation algorithms is to partition the image into perceptually similar regions. Every segmentation algorithm addresses two problems, the criteria for a good partition and the method for achieving efficient partitioning [Shi, 2000].
- **Supervised learning:** Object detection can be performed by learning different object views automatically from a set of examples by means of supervised learning mechanism. Learning of different object views waives the requirement of storing a complete set of examples, then, the learning methods generate a function that maps inputs to desired outputs. A standard formulation of supervised learning is the classification problem where the learner approximates the behavior of a function by generating an output in the form of a class label. In the context of object detection, the learning examples are composed of pairs of object features and an associated object class where both of these quantities are manually defined. The learning methods include, but are no limited to, neural networks [Rowley, 1998] adaptive boosting [Viola, 2003], decision trees [Grewe, 1995] and support vector machines [Papageorgiu, 1998].

**3. Dynamic model**

The dynamic methods make use of the current frame and some previous frames or knowledge taken from them to detect the objects. The principal approach is:

- **Background subtraction:** In this method, object detection is achieved by building a representation of a scene called the background model and then finding deviations from the model for each incoming frame. Any significant change in image region from the background model signifies a moving object. The pixels constituting the regions undergoing change are marked for further processing. Usually, a connected component algorithm is applied to obtain connected regions corresponding to objects. This process is referred to as the background subtraction [Wren, 1997].

**4. Object tracking**

The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. The object tracker may also provide the complete region in the image that is occupied by the object at every time instant. The tasks of detecting the object and establishing the correspondence between the object Instances across frames can either be performed separately or jointly. In the first case, possible object regions in every frame are obtained by means of an object detection algorithm and then, the tracker corresponds objects across frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous frames. We now briefly introduce the main tracking models.

- **Point tracking**: Objects are represented by *Points*. The recognition algorithm is based on *Interest points*. The detected points in consecutive frames are tracked based on the previous point state which can include point position, speed and acceleration (figure 2.a). There are basically two main categories, deterministic methods [Veenman, 2001] and statistical methods [Streit, 1994].
- **Kernel tracking**: Objects are represented by *Primitive geometric shapes*. The recognition algorithm is based on *Segmentation* or *Supervised learning*. The kernel can be a rectangular or elliptical shape with an associated histogram (figure 2.b). Objects are tracked by computing the motion of the kernel in consecutive frames. This

motion is usually in the form of a parametric transformation such as translation, rotation and affine [Schweitzer, 2002].

- **Silhouette tracking**: Objects are represented by the silhouette. The recognition algorithm is based on *Segmentation* or *Supervised learning*. Tracking is performed by estimating the object region in each frame. Silhouettes are tracked by either shape matching or contour evolution (figure 2.c and 2.d). Both of these methods can essentially be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames [Huttenlocher, 1993].

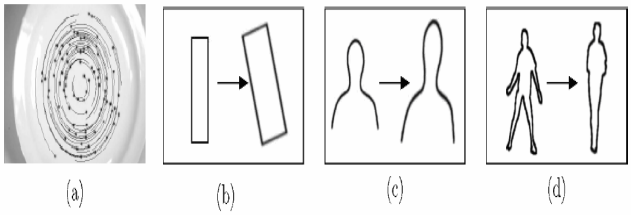



Figure 1 Different tracking approaches. (a) Point tracking (b) Parametric transformation of a rectangular patch, (c, d) Two examples of contour evolution

## 5. Experimental work and results

The experiments were realized without object occlusion. These include both the feature selection and static recognition results and the initial experiments on dynamic recognition and tracking (in which the static recognition module also is involved).

Two video sequences of 88 color images each, that correspond to the left and right sequences of a stereo vision system installed on the MARCO mobile robot at the IRI1 research centre, were employed for an initial validation of the proposed approach. They show an indoor scene with slight changes in perspective and scale caused by the robot movement. In this scene we selected $N$=3 objects of interest: a box, a chair and a pair of identical wastebaskets put together side by side; and the objective was to discriminate them from the rest of the scene (background) and locate them in the images. Figure 2 displays three frames of the right sequence; the whole right sequence can be obtained in http://www.lsi.upc.edu/~alquezar/ris.avi. Before segmentation, the images in the sequences were preprocessed by applying a median filter on the RGB planes to smooth the image and reduce some illumination reflectance effects and noise. Then, all images in both sequences were segmented independently using the Felzenszwalb-Huttenlocher algorithm [Felzenszwalb, 98], which is a pixel merging method based on sorted edge weights and minimum spanning tree. Figure 3 displays three frames of the segmented right sequence

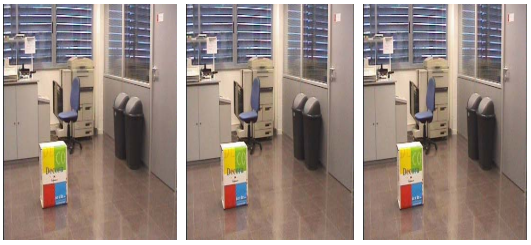


Figure: 2 Three consecutive frames of the right sequence

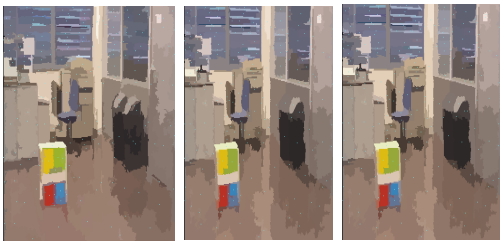


Figure: 3 Three consecutive frames of the segmented right sequence

The output of the segmentation process for each image consists of a list of regions (spots) that partition the image in homogeneous pieces, where each region is defined by the set of coordinates of the pixels it contains. For

each spot, its mass center and several features listed were computed. The experimental work carried out with these image sequences can be split in two parts: the former covers the experiments related only to feature selection and the performance of the static recognition module, whereas the latter includes the experiments related mainly to dynamic recognition and tracking. This presentation order also coincides with the temporal order in which the experiments were performed. This is relevant because the static recognition module is also involved in the dynamic recognition and tracking results, so that the results of the experiments were taken into account for the subsequent work.

5.1 **Feature selection and static recognition results**

In order to be processed as a pattern by a neural network, a spot must be described by a feature vector. Two types of information were extracted from the spots: color and geometry. With regards to color, average and variance values for each one of the three RGB bands were calculated for each spot on the basis of the corresponding intensity values of the spot pixels in the original image (not in the segmented image, for which spot color variance would be zero). This is, the result of the segmentation algorithm served to identify the pixels of every spot, but the color characteristics of these pixels were taken from the original RGB image. The geometrical information might include features related to position, orientation, size and shape. Because of the robot movement, we were mainly interested in shape descriptors that were invariant to translation and scale, and to this end, we decided to use the seven invariant geometric moments defined by Hu [Hu, 62]. In addition and since the range of variation of the objects' size was rather limited in the video sequence, we also calculated and used the size of each spot, i.e. its area measured in number of pixels.

For object learning, spots selected through ROI (region-of-interest) windows in the left image sequence were collected to train the neural networks. These windows were manually marked on the images with a graphics device to encompass the three objects of interest and a large region on the floor. Figure 4 shows one of the images and its segmentation together with the ROI windows on them. The remaining set of spots, those with its mass center inside the ROI windows, was further filtered by removing all the spots with a size lower than 100 pixels, with the purpose of eliminating small noisy regions caused by segmentation defects. Hence, from the 88 images, a total number of 3,411 spots were finally chosen.

The inputs of the neural nets are the spot features and the target is the class that we impose to the spot. In order to assign a class label to each spot, to be used as target for the spot pattern in the neural network training and test processes, a simple decision was made: each one of the four ROI windows constituted a class and all the spots in a window were assigned the same class label. Note that this is a rough decision, since several background spots may be included in the ROI windows of the objects (especially in the case of the chair) and therefore are not correctly labeled really. Incorrectly labeled patterns are a clear source of error that puts some bounds on the level of classification accuracy that the learning system, in this case a neural net, may reach. However, we preferred to carry out this simple but more practical approach instead of manually labeling each spot, which is obviously a very tedious task.

For illustrative purposes, the spots of figure 4 that were assigned to the three classes associated with the objects of interest are displayed in figure 5; for each class, the union of selected spots is shown in the left and isolated spots that belong to the class are shown in the right.

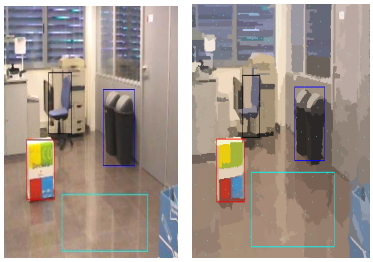
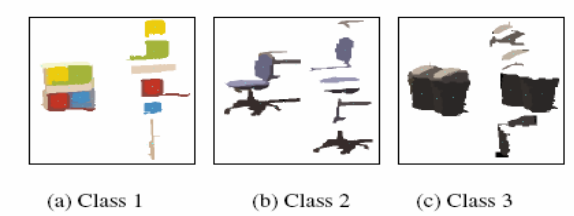


Figure 4 ROI windows. One of the original images (left) and the corresponding segmented image (right), with four boxes marked on them. Spot mass centers are also displayed in the right image



(a) Class 1 (b) Class 2 (c) Class 3

Figure 5 Selected spots. Labeling of the object spots from the segmented image in figure 4

In the experiments we used neural nets with a feed-forward two-layer preceptor architecture using standard gradient descent through back propagation as training algorithm. After some preliminary experiments, we set the number of hidden units to 180, although we observed that the results were not very sensitive to this choice. Hyperbolic tangent and sine functions were used as activation functions in the hidden layer and the output layer, respectively. For back propagation, we set a learning rate of 0.003, a momentum parameter of zero and a maximum number of 500 training epochs for each run.

In addition, starting from the architecture selected for the full set of features, a sequential backward selection method [Romero, 03] was applied trying to determine a good subset of input features by eliminating variables one by one and retraining the network each time a variable is temporarily removed. In this case, each partition of the cross-validation procedure divided the dataset in 60% of patterns for training and 40% for test (no validation set) and the training stop criterion was to obtain the best result in the training set for a maximum of 2,000 epochs. The results of the sequential backward feature selection clearly confirmed that invariant moments and RGB color variances were practically useless (since they were the first features removed without significance performance degradation) and that RGB color averages provided almost all the relevant information to classify the spots. Using spot size and RGB averages and variances as features, the network (and associated dataset partition) that gave the best result in the training set (97.25%) was selected for computing the weighted mass centers and to assess the effect of the clustering process on the spot classification performance. A 78.8% of the spots misclassified by the network were correctly reclassified by the clustering and only a 0.1% of the correctly classified spots were incorrectly reclassified. Figure 6 displays an example of the beneficial effects of performing the structural reclassification. In the left hand image, there are two spots that were misclassified by the net, one in the chair was classified as wastebasket and one in the wastebasket was classified as chair. These spots could be correctly reclassified after the structural reclassification, as shown in the right hand image.

Finally, the network selected in the previous experiment (trained from just the left image sequence) was applied to classify the spots in the right image sequence but only those within the same ROI windows, giving a 90% of correctly classified spots. However, for the test phase, it is somewhat tricky to restrict the object recognition to predefined ROI windows, since we cannot rely on having the ROI windows marked on every frame in a realistic experimental scenario. ROI windows for each object were only defined in the first frame to initialize the tracking images. To the contrary of the experiments that have been reported in this subsection, in the following experiments we were not so interested in achieving a high spot classification ratio but a sequence of tracking images of good quality for each object of interest, as a first validation of the methodology proposed.

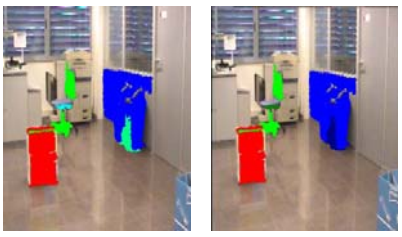


Figure 6 Spots classified as belonging to the three objects by the net (left) and the result of the reclassification after the clustering (right)

### 5.2 **Dynamic recognition and tracking results**

In the next experiment, dynamic object recognition was achieved through the use of probability images and the update function and we employed the tracking decision function based on a simple predictive model. For static object recognition, the trained neural network was applied to estimate the class probabilities for all the spots in the right image sequence. As mentioned before, the spot class probabilities were replicated for all the pixels in the same spot.

For object tracking in the right sequence, ROI windows for each one of the three objects were only marked in the first image to initialize the tracking process and the dynamic class probabilities.
Figures 7 and 8 shows the results of tracking the three objects of interest in three consecutive frames that belong to the right sequence. In Figure 23, the corresponding binary images of tracking each object are applied as a transparency mask to the original images in order to visualize only the pixels considered to belong to the object (the rest of pixels are set to white). Similar results for the whole sequence can be seen in http://www.lsi.upc.edu/~alquezar/box.avi , chair.avi and basket.avi, respectively.

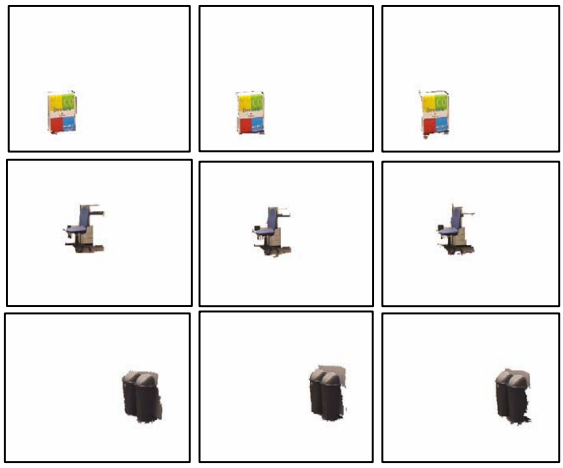


Figure 7 Tracking of the three objects of interest in the second experiment

Figure 8 displays a more informative picture of the tracking process in the same frames shown in figure 7. The one-valued pixels of the tracking images are divided in two groups: those colored in yellow correspond to pixels labeled as “certainly belonging to the object” by the tracking method, and those colored in light blue correspond to pixels initially labeled as “uncertain” but with the largest dynamic recognition probability for the object class . The zero-valued pixels of the tracking images are divided in three groups: those colored in dark blue correspond to pixels labeled as “uncertain” and with a low probability, those shown in dark grey correspond to pixels labeled as “certainly not belonging to the object” but with a high probability for the object class (which are mostly recognition mistakes that are ignored thanks to the tracking prediction) and the rest are black pixels with both a low probability and a “certainly not belonging to the object” label. Similar results for each object in the whole right sequence can be observed in http://www.lsi.upc.edu/~alquezar/box_track.avi, chair_track.avi and basket_track.avi, respectively. The preliminary tracking results can be considered quite satisfactory for the three objects, especially if we note that numerous spots are incorrectly classified by the neural network within the static recognition module. The proposed tracking method allows a reasonable recovery of these recognition errors without relying on any contour detection and tracking procedure.

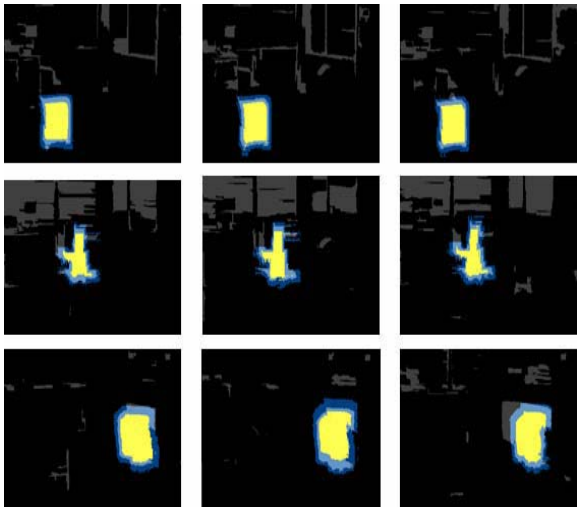


Figure 8 Analysis of tracking process

## 6. CONCLUSIONS AND FUTURE WORK

An improved method for object tracking, capable of dealing with rather long occlusions and same-class object crossing, has been proposed to be included within our probabilistic framework that integrates recognition and tracking of objects in image sequences. PIORT does not use any contour information but the results of an iterative and dynamic probabilistic approach for object recognition. These recognition results are represented at pixel level as probability images and are obtained through the use of a classifier (e.g. a neural network) from region-based features. We think that PIORT approaches for object tracking are especially suitable in noisy environments where segmented images vary so much in successive frames that it is very hard to match the corresponding regions or contours of consecutive images. The empirical results presented are quite satisfactory, despite the numerous mistakes made by the static recognition module, which can be mostly ignored thanks to the integration with the proposed tracking decision module.

As future work, we want to extend the experimental validation of PIORT by applying it to new and more difficult image sequences. In addition, we are interested in implementing and testing new classifiers in the static recognition module, which could exploit other features completely different to the basic color features used up to now. For instance, an SVM classifier could be applied to a set of features formed by Gabor filter responses, provided that class probability values were estimated from margin values.

## REFERENCE


[1] Sajjad Baloch, and Hamid Krim, Object Recognition through Topo-Geometric Shape Models Using Error-Tolerant Sub graph Isomorphism's, IEEE transactions on image processing, vol. 19, no. 5, May 2010

[2] N. Amézquita Gómez, R. Alquézar, F. Serratosa,"A Probabilistic Integrated Object Recognition and Tracking Framework" submitted to Journal on Image and Vision Computing, JIVC , ISSN: 0262-8856 ELSEVIER, Chief K.D. Baker, M. Pantic Editors

[3] "Experimental Assessment of Probabilistic Integrated Object Recognition and Tracking Methods"Proc.14th Iberoamerican Congress on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, México, Springer,LNCS-5856.

[4] R. Alquézar, N. Amézquita Gómez, F. Serratosa, "Tracking deformable objects and dealing with same class object occlusion", in: Proc. Fourth Int. Conf. On Computer Vision Theory and Applications (VISAPP 2009), Lisboa, Portugal.

[5] [Chen 2003] F-S. Chen, C-M. Fu and C-L. Huang, Hand Gesture recognition using a real-time tracking method and Hidden Markov Models, Image and Vision Computing, vol. 21, pp: 745-758, 2003.

[6] [Cremers, and Schnorr, 2003] Cremers, D. and Schnorr, C. 2003. Statistical shape knowledge in variational motion segmentation. I.Srael Nent. Cap. J. 21, 77–86.

[7] [Comaniciu, 2002] Comaniciu D. and Meer P., Mean shift: a robust approach toward feature space analysis. IEEE Trans. Patt. Analy. Mach. Intell. 24, 5, 603–619, 2002.

[8] [Fiesler E ,1997]. Fiesler E. and Beale R. (eds.), Handbook of Neural Computation, IOP Publishing Ltd and Oxford University Press, 1997.

[9] [Foresti ,1999]. G. L. Foresti, "Object recognition and tracking for remote video surveillance", IEEE Trans. on Circuits and Systems for Video Technology 9 (1999) 1045-1062.

[10] [Grewe, 1995] Grewe, L. and Kak, A. 1995. Interactive learning of a multi-attribute hash table classifier for fast object recognition. Comput. Vision Image Understand. 61, 3, 387–416.

[11] [Hariharakrishnan,2005] K. Hariharakrishnan, D. chonfeld, "Fast object tracking using adaptive block matching", IEEE Trans. Multimedia 7 (2005) 853-859.

[12] [Kerner 2004] B.S. Kerner, H. Rehborn, M. Aleksic and A. Haug, Recognition and Tracking of spatial-temporal congested traffic patterns on freeways, vol 12, pp: 369- 400, 2004.

[13] [Lee, 2005] K-C. Lee, J. Ho, M-H. Yang, D. Kriegman, Visual Tracking and Recognition using Probabilistic appearance manifolds, Computer Vision and Image Understanding, vol. 99, pp: 303-331, 2005.

[14] [Mutch, 2006] Mutch J. and Lowe D.G., Multiclass Object Recognition with Sparse, Localized Features, Proceedings CVPR'06, New York, pp. 11 – 18, June 2006.

[15] [Hu, 1962] Hu M.K., Visual pattern recognition by moment invariants, *IRE Trans. On Information Theory*, Vol. 8 (2), pp.179-187, 1962.

[16] [Huttenlocher, 1993] Huttenlocher, D., Noh, J., and Rucklidge, W. 1993. Tracking nonrigid objects in complex scenes. In *IEEE International Conference on Computer Vision (ICCV)*. 93–101.

[17] [Ito,2001] K. Ito, S. Sakane, "Robust view-based visual tracking with detection of occlusions", in: *Proc. Int. Conf. Robotics Automation*, 2001, vol. 2, pp. 1207-1213.

[18] [Jepson,2003] A.D. Jepson, D.J. Fleet, T.F. EI-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (2003) 1296- 1311.