# SIFT-Based Segmentation of Multiple Instances of Low-Textured Objects

Paolo Piccinini, Andrea Prati, and Rita Cucchiara

Abstract—This paper proposes a simple yet effective approach for segmenting multiple instances of the same object for a pick-and-place application. The considered objects present several challenges, such low as texture. semi-transparent container, moving parts, and severe occlusions. Real-time constraints must be met, calling for a good trade-off between accuracy and efficiency. For all these reasons, the proposed approach is based on SIFT features and a suitable modification of the 2NN matching procedure to increase the number of available matches. Moreover, in order to reduce false segmentations, ad-hoc algorithms based on overlap detection and color similarity are used.

*Index Terms*—Machine vision, pick-and-place automation, object multiple instance detection, SIFT.

## I. INTRODUCTION

Information technologies have become in the last decades a fundamental aid for helping the automation of everyday people life and industrial processes. Among the many different disciplines contributing to this process, machine vision and pattern recognition have been widely used for industrial applications and especially for robot vision. A typical need is to automate the *pick-and-place* process of *picking* up objects, possibly performing some tasks, and then *placing* down them on a different location. Object picking can be very complicated if the scene is not well structured and constrained and as a first step it requires object recognition and localization.

Vision-guided pick-and-place processes often require to work with many different types of objects of different sizes and complexity. In particular, this paper considers a very challenging category of objects: low-textured, deformable and self-occluded objects. Some examples are reported in Fig. 1. In many previous works about object recognition, query objects are well textured and composed of fixed parts, such as toy cars, books, or shoes. Conversely, our approach also considers objects with low-textured, semi-transparent parts which can possibly move each other, such as the syringes in Figs. 1(a) and 1(b), where plugs and pistons can move, rotate and assume different relative positions. Moreover, these

Manuscript received June 15, 2012; revised August 10, 2012. This work was supported by Marchesini Group Spa, Pianoro (BO), Italy.

Paolo Piccinini is with Department of Information Engineering, University of Modena and Reggio Emilia, StradaVignolese, 905, I-41125 Modena, Italy (email: paolo.piccinini@unimore.it)

Andrea Prati is with Department of Design and Planning of Complex Environments, University IUAV of Venice, Santa Croce 1957, I-30135 Venice, Italy (email: andrea.prati@iuav.it)

Rita Cucchiara is with Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, StradaVignolese, 905, I-41125 Modena, Italy (email: rita.cucchiara@unimore.it) objects are often inserted in *flowpacks* for hygienic reasons and this contributes to the creation of reflexes and varying appearance which complicate object segmentation.



Fig. 1. Examples of objects considered in this paper.

Moreover, a very useful characteristics of these systems is to not require ordering and regular disposal of objects, but to allow their random disposal. The ultimate goal can be to work directly in bins (problem known as *bin picking* [1]), for saving time and/or for hygienic reasons. Finally, the required working speed is typically very high: a fast yet accurate detection technique should be adopted to work more than a hundreds of objects per minute.

This paper presents a supervised method which tackles three main challenges: multiple instance segmentation (all the or most of the instances of the query object must be segmented), the aleatory aspect of the object (due to moving parts and flowpacks) and the real-time requirement. With regards to the multiple instance segmentation, it can be seen as an extension of the one-to-one object correspondence problem to the one-to-many case. This problem can be turned into the problem of matching a feature from the model with a number of features in the current image. The current literature reserved very few works to the one-to-many matching of features, whereas lot of papers addressed many-to-many feature correspondences, but only for the recognition of the same object at different scale using graph matching [2].

The segmentation method proposed in this paper is based on SIFT features and the Two-Nearest-Neighbor (2NN) matching method proposed in [3]. However, while 2NN demonstrated impressive accuracy in dealing with one-to-one correspondence, its performance decreases dramatically when multiple correspondences are searched. This is due to the fact that every feature of the model is matched with only the best candidate feature in the current image, leaving all the other same features on the other instances without any correspondence. To avoid this loss in performance a modification in 2NN matching is here proposed. This modification increases the correct match rate when multiple mostly-identical features are present, but also introduces wrong segmentations that are properly managed with a specific false detection algorithm. Moreover, the real-time requirement is fulfilled thanks to the use of a version of SIFT algorithm implemented on a GP-GPU (General Purpose Graphical Processing Unit) [4].

The paper is structured as follows: the next Section will present works related to fast and accurate object segmentation with multiple instances; Section 3 will present our method for solving the problem and Section 4 reports the tests performed to evaluate its performances.

## II. RELATED WORKS

Approaches to object recognition and detection can be categorized by the fact if they use either global or local features. Global methods were the first used in object recognition, and exploit a global description of the object for the matching task. With these methods, objects are represented with features like color histogram [5], shape [6] or shocks graphs [7], and in the test image the same features are computed and compared with a proper similarity measure. Although these methods are interesting under the theoretical point of view, in real applications they have been demonstrated sensitive to background clutter and object occlusions that make them unreliable for the most of the industrial applications.

In local methods, instead, such those proposed by [3,8], objects are represented as a set of local features. In order to be useful for object recognition tasks, these local features must be invariant to scale, light changes and reasonable changes in the point of view. Usually these local methods are composed of two steps: feature localization and feature descriptor computation, employed on both the model and the current image. Then, with the help of a proper similarity measure, the features from these two images are matched to estimate the object pose in the current image. In these methods the model is given by an image of the query object taken on a plain background or in a cluttered image where the query object is somehow bounded, so that all the features useless for the recognition are discarded. The successive step of classification can be accomplished by either supervised or unsupervised methods [9,10,11].

Whichever feature descriptor is chosen, local methods always need a way to match features between two images or between a model and the current image. This matching is based on a proper similarity measure and a correct policy for selecting the best match. Among the possible solutions, SIFT feature descriptors and 2NN matching (based on Euclidean distance between descriptors) are the most used in the literature. However, Ferrari *et al.* [12] and Cho *et al.* [13] have already noticed the 2NN matching weakness for multiple identical objects in the scene, but they overcome the problem using a matching based on prefixed thresholds. Their proposals increase the number of correct matches but increase at the same time the number of wrong correspondences. Both these proposals get rid of the problem using a validation step.

In [12], Ferrari *et al.* proposed an object recognition and segmentation method based on affine invariant regions. After producing a large set of matches, the method iteratively explores the surrounding areas increasing matched regions and deleting the wrong ones. The method also deals with multiple model views for integrating the contributions of the views at recognition time. Authors state that the method is useful also for segmenting different instances of the same object, but failed in providing sufficient evidence of this. Additionally, the method is too slow for our requirements since it requires roughly 5 seconds to process a single image.

Cho *et al.* [13] investigated the detection of identical objects in the same image without any supervision, achieving good results. They proposed a match-growing algorithm similar to [12] and estimated the geometric relationship between object entities by means of *object correspondence networks*. The method is robust to scale changes and to small rotations. However, this approach cannot recognize two instances of the same object but different faces or views.

We tested both these solutions and they resulted unsatisfactory for our purposes due to the excessive number of wrong segmentations that they produce.

Finally, authors in [14] proposed a method based on dominant orientation for the real-time detection of texture-less objects. The key point of this method is the template representation that has been designed to be robust to small image transformations. The method seems reliable for non-deformable objects, but its behavior has not been discussed when the objects are severely occluded. Since the template is modeled with gradient orientations its behavior becomes unpredictable in the case of deformable objects, which makes this method unsuitable for the objects considered in this paper.

The method proposed in this paper is meant to tackle all these issues using a simple yet effective method that we called *Partitioned 2NN* (P-2NN), where the image is partitioned into P areas and the SIFT+2NN method is applied for each area separately. Proper actions are then taken to prevent increasing false or poor segmentations.

# III. THE PARTITIONED 2NN METHOD

The method proposed in this paper is an improvement of [15], with several additions aiming at increasing the accuracy in object segmentation in the case of occluded and low-texture objects. The final objective of the system is to segment as many object instances as possible in cluttered scenes such as those reported in Fig. 1. In order to segment multiple instances in a cluttered scene, the first step is to collect several models (or templates) of the query object. Since objects may have different faces/views and given our

requirement of random object disposal, different models for each face must be acquired. Each object model consists simply in an image containing a single object on a plain background. All the models are taken under free environmental illumination and using different object orientations in order to be robust to the reflexes created by a possible transparent container.



(b) P-2NN P=2



(c) P-2NN P=4

Fig. 2. Examples of feature matching between one model (as a proof-of-concept - left top) and the current image, at different value of P).

After this step, we obtain a complete set of models M for a given object, composed of N images, where  $N = \sum_{i=1}^{FN} N^i$ , with FN the total number of faces and  $N^i$  the number of models used for the face i. We will indicate the model  $M_j^i \in M$  as the *j*-th model for the *i*-th face, with i = 1, ..., FN and  $j = 1, ..., N^i$ .

Each considered model  $M_j^i$  generates a set of SIFT keypoints  $\mathcal{K}_j^i$ . The keypoint sets of a given face *F* are grouped together  $\mathcal{K}^F = \{\mathcal{K}_1^F, ..., \mathcal{K}_{N^F}^F\}$  and compared with the set *K* of the keypoints extracted from the current image. It is worth noting that each set  $\mathcal{K}_j^F$  is matched separately with *K*, resulting in the set of matches  $\mathcal{M}^{F,j}$ . This choice is very important: in fact, given several models of the same face, it is very likely that there are strong keypoints which are found on all the models (even though rotated and translated); keeping them separated in the matching will result on multiple matches on the same strong point in the current image (possibly one for each object instance) and it increases the number of useful matches for the segmentation of multiple instances.

The obtained matches  $\mathcal{M}^{F,j}$  are merged to form  $\mathcal{M}^F = \bigcup_{i=1}^{N^F} \mathcal{M}^{F,j}$ . Then, for each matched feature a vector of

distances between its position and a certain number of control points defined by the user (including the center of the object) is computed. The control points are chosen so to define a boundary of the object shape (the actual segmentation of the object). The estimated positions of the control points for each instance in the current image (projected assuming a pure Euclidean transformation, which is a suitable hypothesis with far, down-looking camera as typical in pick-and-place applications) are used to obtain a rough yet accurate segmentation of the object instances using mean shift to cluster close points. Please refer to [15] for further details about the method.

The basic step of this method is, however, the matching of features between the models and the current image. If this is unreliable or if too few matches are provided, the projected control points result unstable and imprecise and many false or missed segmentations arise. In particular, the low texture of our objects produces few features to match with, even after the use of multiple models per face. To worse this situation, the standard 2NN approach embedded in the SIFT procedure allows only one (the best) match for each feature. As shown in Fig. 2(a). the standard 2NN leaves some instances with too few matches for a good segmentation (blue squares indicate missing matches). In fact, although two models can have a SIFT feature in the same position, it is not guaranteed that the two descriptors will be identical, and thus they can be matched with two different features in the current image.

In this paper, we propose a new method which aims at further increasing the number of matches between the models and the current image. We called this method partitioned 2NN (or P-2NN) since it is based on partitioning the image in P parts and applying the 2NN matching over each partition separately. The matching on each partition p with the model *j* of face *F* produces the set of matches  $\mathcal{M}^{F,j,p}$  =  $\left\{\mathcal{M}_{1}^{F,j,p}, \dots, \mathcal{M}_{m^{F,j,p}}^{F,j,p}\right\}$ , where  $m^{F,j,p}$  is the number of matches on that specific partition. In this way, each feature in a model can be matched, at most, P times, which contributes to increase the global number of matches. Fig. 2(b) shows how dividing the image in two columns (P = 2) the same feature on the model is matched twice, one for each partition. By increasing the number of partitions to 4 (Fig. 2(c)) the number of available matches for that feature increases as well. Obviously, if more than two very similar features lay in the same partition (as in the case of top right partition in Fig. 2(c)), only one can be matched.

One may argue that partitioning the image can result in having objects which are split between two adjacent regions (see Fig. 2). For this reason, the matching procedure is carried out separately for each partition p but the obtained matches are then merged together to provide, similarly to [15], the set  $\widetilde{\mathcal{M}}^F$  of matches for the face F. Differently from [15], in *P*-2NN this set can be written as:  $\widetilde{\mathcal{M}}^F = \bigcup_{i=1}^{N^F} \bigcup_{p=1}^{P} \mathcal{M}^{F_i j, p}$ .

The cardinality of  $\widetilde{\mathcal{M}}^F$  is typically bigger than that of  $\mathcal{M}^F$  defined above and this improves the accuracy in the segmentation, as will be shown in Section IV.

The increase in the matched features does not bring, however, only benefits, since it is likely to generate also several wrong matches. This is due to the fact that by reducing the considered area the 2NN selects weaker matches. This increase of wrong matches contributes to the creation of false or erroneous segmentations, which can be mainly of two types: the first type (that we called *shadow segmentations*) is strictly related to a correct segmentation but producing a duplicate of it which is slightly translated and/or rotated; the second type consists of the actual *wrong segmentations*. Fig. 3 shows an example of segmentation: Fig. 3(a) shows the case of classical 2NN with few yet correct segmentations, while Fig. 3(b) the case of *P-2NN* which contains both shadow (in blue) and wrong segmentations (in red).



Fig. 3. Example of segmentations (green=correct, blue=shadow, red=wrong).

In order to get rid of shadow segmentations the following consideration is drawn: if two segmentations overlap for more than 30% of their area and have almost the same orientation (within an approximation of about 20 degrees), they are likely to regard the same object instance. Thus, only the segmentation obtained with the highest number of matches (i.e., with the highest score, which in some sense means that the segmentation is more reliable) is retained, while the others are classified as shadow segmentations and removed. Fig. 3(c) shows the benefit of this simple heuristic rule which works quite well but tends to fail when two segmentations of the same instance have equal score.

To account for this latter case and for the actual wrong segmentations, we developed a method based on the color similarity between the segmented object and the model. The color similarity can be expressed in many ways, but given our tight time constraints we prefer to use simple color histograms. Moreover, instead of employing a single 3-D histogram in RGB (as proposed in [5]), three separate 1-D histograms (one for each color channel) are used. After smoothing the histograms with a running average (similarly to what is done in computing the SIFT keypoint orientations), the positions of the  $n_p$  peaks of each of the histograms are computed (where  $n_p$  is a fixed parameter, equal to 3 in our experiments). The similarity S between the segmented object and the model is:

$$S = \frac{1}{\frac{1}{\frac{1}{3}\sum_{c \in R, G, B} \sum_{r=1}^{n_p} w_r^c \cdot |o_r^c - m_r^c|}}$$
(1)

where  $w_r^c$  is the magnitude of the  $r^{th}$  peak in the histogram of color channel  $c \in R, G, B$ , and  $o_r^c$  and  $m_r^c$  are the position of the peak for the segmented object and the model, respectively. If the similarity *S* is lower than a preselected threshold  $\tau$ , the segmentation is discarded. In the case of multiple models, the similarity *S* is averaged among the models. Fig. 3(d) demonstrates how this additional check allows to remove all the false segmentations introduced by P-2NN (Fig. 3(b)).

# IV. EXPERIMENTAL RESULTS

In order to demonstrate the advantages of the proposed method, we build an experimental campaign which aims at evaluating three different aspects (as a function of the number of partitions *P*):

- the percentage of keypoints of the current image correctly matched with the model(s);
- [2] the number of correctly segmented objects;
- [3] the accuracy in the segmentation (to account not only for segmented objects but rather for *well segmented* objects).

It is worth noting that since we are experimenting with a real on-line working system the results can be heavily dependent on the illumination and the exact time in which they are performed. For this reason, all the experiments are repeated five times and the values reported in the following are the result of the averaging on these five runs.

In Table I the number of matches, which percentage of them are correct and the percentage of keypoints correctly matched as a function of P are reported (P = 1 corresponds to the classical 2NN). These values have been computed in fairly complicated scenes.

TABLE I: AVERAGE NUMBER OF MATCHES, PERCENTAGE OF CORRECT MATCHES AND OF KEYPOINTS CORRECTLY MATCHED AS A FUNCTION OF P.

Р	Average # of matches	% of correct matches	% of keypoints correctly matched
1	48.2	77.21%	23.41%
2	75.8	75.73%	35.99%
4	123.2	66.38%	50.79%
6	137.0	62.01%	52.16%
9	147.6	70.01%	63.14%

As expected, the number of matches increases with the number of partitions, even though the percentage of correct matches slightly decreases (from 77.21% to 62.01%), due to the fact that the image partitioning increases the number of false matches. The percentage of correctly-matched keypoints raises from 23.41% to a maximum of 63.14%. As successive tests will make clearer, the increase in the number of matches will produce better segmentations.

TABLE II: PRECISION AND RECALL AT OBJECT LEVEL.

	False segmentation disabled		False segmentation enabled	
Р	Precision	Recall	Precision	Recall
1	80.00%	42.86%	100.00%	34.29%
2	58.26%	51.43%	81.67%	34.29%
4	56.17%	65.71%	95.00%	45.71%
6	45.00%	65.71%	93.33%	65.71%
9	41.50%	57.14%	86.00%	65.71%

Table II shows the system performance in terms of precision and recall at object level. These two values help in evaluating how good the system is in detecting correct objects while minimizing false positives (i.e., false segmentations). The values have been computed by disabling or enabling the procedure described in Section 3 for removing shadow and wrong segmentations (here generally called *false segmentations*). As it was foreseeable, the recall value increases when *P* increases, that means that more

correct segmentations are obtained. At the same time, however, the precision decreases due to the increased number of wrong matches. By comparing the results obtained when the false segmentation algorithm is enabled with those when it is disabled, it is evident that this algorithm allows to increase significantly the precision. On the other hand, when the algorithm is enabled, the recall is generally lower because some good segmentations are rejected together with wrong ones.

Table I and II confirm that our approach produces in general more matches and consequently the number of correct segmentations increases (Table I). It also increases the number of false segmentations which can be, however, handled well using our false segmentation algorithm (Table II).

Results so far demonstrated that the proposed methods can augment the number of correct segmentations, by keeping low the false ones. However, whether a segmentation is correct or not is often a subjective choice, directly related to the overlap of the obtained segmentation with the real object. It may happen that the system produces more correct segmentations but at the cost of lower accuracy, which can be a problem for object picking.

The last experiment aims at evaluating also the level of accuracy of the obtained segmentations, by considering the overlapping between the segmentation obtained by the system and that given by the manual ground truth. Table III summarizes the results in terms of precision and recall at pixel level.

	False segmentation disabled		False segmentation enabled	
Р	Precision	Recall	Precision	Recall
1	83.52%	92.83%	83.52%	92.72%
2	80.37%	86.54%	80.20%	87.52%
4	84.05%	92.15%	86.54%	92.33%
6	79,86%	86,75%	82.55%	90.54%
9	81.41%	89.39%	82.76%	89.44%

TABLE III: PRECISION AND RECALL AT PIXEL LEVEL.

Looking at Table III it seems that the false segmentation algorithm has no clear influence on the accuracy of the segmentation. This happens because the precision and the recall are computed only for correct segmentations (which are globally increased using the false segmentation algorithm). As a further proof of the goodness of the system, Fig. 4 shows some visual results of the segmentation in complex scenes.



Fig. 4. Examples of segmentation in complex scenes.

Finally, one may argue that increasing the number of partitions can slow down the system since it requires

additional and redundant matches. This is surely true, but our experiments demonstrate that the increase in computational time is negligible. In fact, in our GPU-based implementation the processing time raises from 0.05 seconds per image when P = 1 to 0.074 seconds in the case of P = 9. It is also worth noting that these computational efforts are very limited and that, considering to pick up a single object per every processed image (which is a worst-case scenario), the required speed of hundreds of objects per second is met.

#### V. CONCLUSIONS

In conclusion, this paper presents a suitable modification to the 2NN matching procedure to increase the number of matches but also keeping low the number of false segmentations using two effective heuristic rules which evaluate pairwise the amount of overlap between segmentations (discarding weaker segmentations in the case of significant overlap) and the color similarity with the model to discard wrong segmentations. Reported results are promising, also given the very efficient solution provided (more than 100 images per second).

#### REFERENCES

- K. Rahardja and A. Kosaka, "Vision-based binpicking: recognition and localization of multiple complex objects using simple visual cues," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots* and Systems, pp. 1448–57, 1996.
- [2] A. Shokoufandeh, Y. Keselman, M. F. Demirci, D. Macrini, and S. J. Dickinson, "Many-to-many feature matching in object recognition," *Cognitive Vision Systems*, pp. 107–125, 2006.
- [3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110 2004.
- [4] S. N. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Gen, "Gpu-based video feature tracking and matching," in *Proc. of Workshop on Edge Computing Using New Commodity Architectures*, 2006.
- [5] M. J. Swain and D. H. Ballard, "Color indexing," International Journal of Computer Vision, vol. 7, no. 1, pp. 11–32 1991.
- [6] C. M. Cyr and B. B. Kimia, "3D object recognition using shape similiarity-based aspect graph," in *Proc of International Conference* on Computer Vision, pp. 254–261 2001.
- [7] A. Bataille and S. Dickinson, "Coarse-to-fine object recognition using shock graphs," *Graph-Based Representations in Pattern Recognition*, vol. 3434 of Lecture Notes in Computer Science, edited by L. Brun and M. Vento, pp. 203-212. Springer Berlin Heidelberg, 2005.
- [8] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in Proc of the 7th European Conference on Computer Vision-Part I (ECCV), pp. 128–142, 2002.
- [9] L. Karlinsky, M. Dinerstein, D. Levi, and S. Ullman, "Unsupervised classification and part localization by consistency amplification," in *Proc of the 10th European Conference on Computer Vision*-Part II (ECCV), pp. 321–335 2008.
- [10] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," in *Proc of the IEEE International Conference on Computer Vision* (ICCV) 2005.
- [11] S. Todorovic and N. Ahuja, "Unsupervised category modeling, recognition, and segmentation in images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 12, pp. 2158–2174, 2008.
- [12] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Simultaneous object recognition and segmentation from single or multiple model views," *International Journal of Computer Vision*, vol. 67, no. 2, pp. 159–188, 2006.
- [13] M. Cho, Y. M. Shin, and K.. M. Lee, "Unsupervised detection and segmentation of identical objects," in *Proc of IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR), 2010.
- [14] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *Proc of IEEE International Conference on Computer Vision and Pattern Recognition* (CVPR), 2010.

[15] P. Piccinini, A. Prati, and R. Cucchiara, "A fast multi-model approach for object duplicate extraction," in *Proc of Workshop on Applications* of Computer Vision (WACV) 2009.



Paolo Piccinini was born in Modena, Italy, in September,10, 1980.He received the MS degree in 2007 andthe PhD degree in 2011.Currently he is a computer vision engineer at Marchesini Group Spa developing computer vision systems for automation application. As PhD student worked at a regional research project, in summer2008 attended at ICVSS summer school and in December 2009 participated at the conference WACV.His research interest concerns

object detection, pattern recognition,3D reconstruction, code optimization and parallel programming withCUDA.



Andrea Prati received the MS degree in 1998and the PhD degree in 2001. He is an associateprofessor currently on the Department of Design and Planning of Complex Environments of the University IUAV ofVenice. He collaborates on research projects atthe regional, national, and European level. He isan author or coauthor of more than 100 papersin national and international journals and conferenceproceedings, he has been an invitedspeaker, organizer of workshops

and journal'sspecial issues, and a reviewer for many international journals in

the field of computer vision and multimedia. He has also been the program chairof the 14th International Conference on Image Analysis and Processing (ICIAP), held in September 2007 in Modena, and of the ACM/IEEEInternational Conference on Distributed Smart Cameras (ICDSC) 2011 and 2012. He is a senior member of the IEEE, and a member of the IEEE Computer Society, the ACM, and the GIRPR (IAPR Italy).



**Rita Cucchiara** received the MS degree in 1989 and the PhD degree in 1992. She has been a fullprofessor at the University of Modena andReggio Emilia since 2005. Since 2008, she hasbeen deputy dean of the Faculty of Engineeringand heads the Imagelab laboratory (http://imagelab.ing.unimore.it). Since 2010, she hasbeen Scientific Responsible of the ICT Platformin the High Technology Network of

EmiliaRomagna. Her research interests regard computervision, pattern recognition, and multimedia systems. She isresponsible of many research projects (national and EU projects), especially in people surveillance, urban security, and human-centeredmultimedia search of images and videos. She is the author of more than200 papers in journals and international proceedings, and she acts as areviewer for several international journals. She is in the EB of the MTAPand MVA journals, chair of several workshops on surveillance, trackchair for ICPR 2012, and general chair of ICIAP 2007 and AI\*IA 2009.She is a member of the IEEE, the IEEE Computer Society, and theACM. Since 2006, she has been a fellow of the IAPR.