International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:6, 2007

Object Recognition in Color Images by the Self Configuring System MEMORI

Michela Lecca

Abstract**—** System MEMORI automatically detects and recognizes rotated and/or rescaled versions of the objects of a database within digital color images with cluttered background. This task is accomplished by means of a region grouping algorithm guided by heuristic rules, whose parameters concern some geometrical properties and the recognition score of the database objects. This paper focuses on the strategies implemented in MEMORI for the estimation of the heuristic rule parameters. This estimation, being automatic, makes the system a self configuring and highly user-friendly tool.

Keywords**—** Automatic Object Recognition, Clustering, Content-based Image Retrieval System, Image Segmentation, Region Adjacency Graph, Region Grouping.

I. Introduction

OBJECT recognition plays a crucial role in Computer Vision, especially in the semantic description of visual content [5], [9]. Although object recognition has been intensely studied, it still remains a hard and computational expensive problem. The main difficulty in the description of image content is the lack of information about the kind and the number of objects possibly present. Moreover, objects can appear at different locations in the image and they can be deformed, rotated, rescaled, differently illuminated or also occluded with respect to a *reference* view. The problem is to build a system able to recognize, as much as possible in an automatic way, the objects under various conditions. In order to simplify object detection and to reduce computational cost, many systems (e.g. [6], [7], [10]) limit the recognition to specific classes of objects. In these cases, a priori knowledge permits to select the most descriptive features for the objects at hand and to circumscribe the search space. However, even under this restriction, high classification performance is seldom reached. Moreover, many object recognition systems rely on user interaction to label as wrong or correct the returned items or to improve system response [11].

MEMORI (MEMory-based Object Recognition in digital Images) [3], [14] is a system for the detection and recognition of rescaled and/or rotated versions of a set of known objects within digital color images. Unlike other systems [10], [15], tailored to the recognition of specific categories of objects (e.g. airplanes or hand tools), MEMORI is independent of the set of objects to be recognized. In this framework, the object database itself is an input parameter of the system. Object detection and recognition are performed by (a) segmenting the input image in regions; (b) applying a region grouping algorithm producing a set of object candidates; (c) filtering the candidate list. The final output is a list of image portions, each one associated to the most similar object view O_r, along with its recognition score, its scale factor and orientation in the image plane with respect to O_r.

Some parameters related to geometric properties and recognition score of the objects in the database are employed in the region grouping strategy as well in the filtering step. The system automatically estimates these parameters or a range of their variability. It also evaluates its own recognition performance on the database images, and its accuracy on the determination of the scale factor and the orientation angle of the objects recognized. These capabilities makes MEMORI a self configuring system with respect the input database. Moreover, no user interaction is requested, except for the selection of an object database, of an input image, and for setting up some thresholds related to the desired recognition accuracy.

This paper presents the methods and the algorithms developed for the automatic computation of the heuristic rule parameters.

Section II briefly describes the working phases of MEMORI. Section III presents the methods for the automatic parameter estimation. The detection and recognition performances obtained by running MEMORI on a set of color images containing objects of the database COIL-100 [16] are discussed in Section IV. Section V reports some conclusions and future planes.

II. MEMORI

MEMORI works through of three main operating phases: *pre-processing*, *processing* and *post-processing* (Fig. 1).

The *pre-processing phase* is devoted to the database description and to the computation of the heuristic rules parameters; information about the recognition performance on the images of the input database are provided.

MEMORI manages databases in which each object is described by many 2D images representing different views. Each view is described by means of a normalized vector of low-level features, such as color, texture and shape, along with log-polar transformation, area and elongation axes (the sides of the minimum bounding rectangle of the view region). The computation of the feature vector is carried out by the modified version [1] of the content-based image retrieval system COMPASS [4]. Visual similarity between image regions is expressed as the L^1-distance between the corresponding feature vectors. Since the low-level descriptors are invariant by rescaling and rotating and composition by thereof, the system is able to recognize also rescaled and/or rotated database objects.

The user input of the *processing phase* is a color image. Its segmentation is computed by the system by the algorithm [8],

Manuscript received on 20th April 2006. This work has been funded by the European Union under the Project VIKEF (*Virtual Information and Knowledge Environment Framework*). Michela Lecca is with ITC - irst, Istituto Trentino di Cultura, Centro per la Ricerca Scientifica e Tecnologica, 38050 - Povo, Trento, Italy (e-mail: lecca@itc.it).

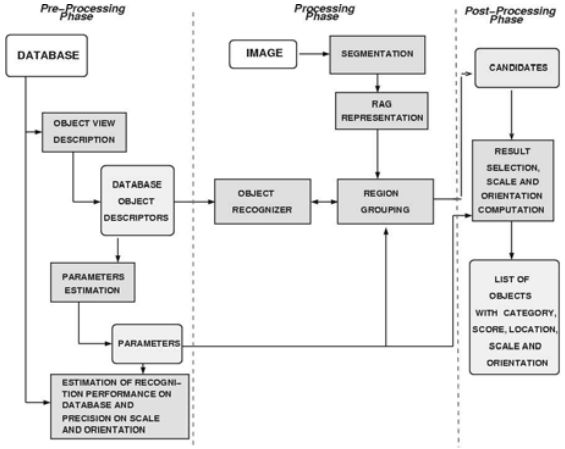

Fig. 1. MEMORI structure. The three main operating phases (pre-processing, processing, and post-processing) are separated by the dotted lines. User inputs, operating components, and system outputs are highlighted respectively by rounded white rectangles, filled rectangles and filled rounded rectangles. The arrows indicate the data flow, the bidirectional arrow denotes the mutual interaction between the object recognizer and the region grouping module.

and then it is represented by a region adjacency graph (RAG), i.e. an undirected graph whose vertices are the image regions produced by the segmentation and whose edges represent their adjacency relationships.

The core of the object detection and recognition procedure is the synergy between the *object recognizer* and the *region grouping* module.

The object recognizer describes each image region group R with the same features used for the object views description. It computes the visual similarity between R and each database item as L^1-distance between their corresponding vectors of features, excluding the shape descriptors. The object recognizer calculates the smallest distance $d(R)$ of R from the item of the database, and if $d(R)$ is smaller than a threshold, it determines the object view O_r of the database such that its distance $D(R)$ from R calculated by considering also the shape features, is minimum. The distances $d(R)$ and $D(R)$, named *recognition distances*, are then used to establish if R is a rotated and/or rescaled version of a database object view (*object hypothesis*, see Subsection II-A).

The idea underlying the recognition procedure is to generate all the possible groupings of adjacent regions, i.e. all the connected vertices sets of the RAG, to compare them with the objects in the database by means of the object recognizer, and to select the groups that are object hypotheses. In real cases, when the RAG is large, the generation of all connected vertices sets is impractical. For this reason, some heuristics mainly depending on the similarity of the region groups to the objects in the database are introduced. The RAG is used to represent in an efficient way the adjacency relationships among the image segments and for the computation of the connected vertices sets. Each segment S of the input image is the root of a tree $T(S)$, whose nodes are connected subgraphs containing S. The set of trees rooted in the image segments, are then organized as well in a tree $\mathbf{T}$, incrementally built following an expansion strategy dependent on the heuristics. A simple example is illustrated in Fig. 2. More details about this data structure is provided in [2] and [12].

TABLE I
SUMMARY OF THE HEURISTIC RULE PARAMETERS.

Parameter	Description
α_1, α_2	scale parameters
A_{min}, A_{max}	area parameters
ϵ_1, ϵ_2, τ	elongation parameters
L	segmented region parameter
d_o, d_g, D_g, f, k	distance parameters
s, N_s, K, M, M_o	exploration parameters

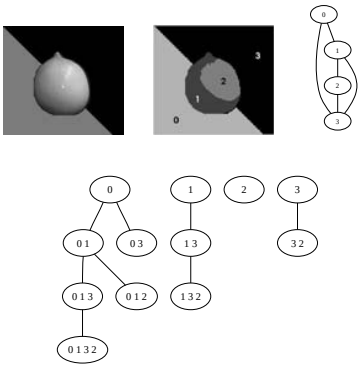

Fig. 2. At top, from left to right: an image, its segmentation and its RAG. At bottom: the trees rooted at the segments and employed in the generation of the region groups. These trees are not unique, but depend on the order of exploration of the image segments (see [2] for more details).

The output of the processing phase is a list of region groups, each associated to the most similar object view O_r and its recognition distances from it.

In the *post-processing* phase, a filtering procedure selects from this list the most reliable hypotheses and for each of them, the system estimates the scale factor and the orientation in the image plane with respect to the associated object view. This is done by comparing the log-polar transformation of each hypothesis with that of the correspondent object view [3].

The subsections II-A and II-B briefly explain the heuristic rules (*constraints*) employed during the region grouping algorithm, and the criteria of the post-processing phase. A more detailed description is provided in [13]. All the used parameters are summarized in Table I.

A. Heuristic rules

Hereafter, we denote the distance of a region group R from the most similar object view calculated including the shape features by using capital letters (e.g. $D(R)$) and the distance calculated without the shape descriptors by using lowercase letters (e.g. $d(R)$). The tree which R belongs to, the area and the elongation axes of R are denoted by $T(R)$, $A(R)$, $e_1(R)$, $e_2(R)$, with $e_1(R) \leq e_2(R)$, respectively.

R is said an *object hypothesis* iff $d(R) \leq d_g$ and $D(R) \leq D_g$. The object view associated to an object hypothesis R by the object recognizer will be indicated by O_r.

- **Constraints on area and elongation axes** (α_1, α_2, $A_{min}, A_{max}, \epsilon_1, \epsilon_2$): the system is able to recognize a database view rescaled by a factor α and/or rotated by an angle θ in the image plane, only if the used features are invariant by the change of scale factor α and by the rotation with angle θ. The system estimates the domain Ω of invariance of the features with respect to rescaling and rotation. It calculates the minimum and maximum values (α_1 and α_2, respectively) of the scale factor in Ω, and it uses them to estimate the variability ranges of the area and of the elongation axes ($[A_{min}, A_{max}]$, $[\epsilon_1, \epsilon_2]$) of the rescaled versions of the object views that can be present in the images. In particular, these parameters are used to avoid the expansion of R if $A(R) \geq A_{max}$ or $e_2(R) \geq \epsilon_2$, and the computation of the feature vector of R if $A(R) \leq A_{min}$ and $e_1(R) \leq \epsilon_1$.
- **Constraint on the number of object regions** (L): the system stops the expansion of R if it is composed by more than L regions.
- **Constraints on distance** (f, k): let d_b be the minimum recognition distance along the path from the node of R to the root of $T(R)$. The expansion of R is stopped if $d(R) \geq f d_b$ with $f \in$ [0,1], and the same happens for at least k direct ancestors of R; this means that the addition of k regions to R worsts the recognition distance (see Fig. 3 for an example).

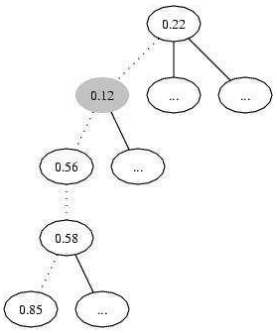

Fig. 3. Example of application of the constraint on distance: let k be 3. The expansion of the dotted branch is stopped because after 3 steps the distance $d_b = 0.12$ (filled node) doesn't decrease. The number inside the nodes is its recognition distance (computed without considering the shape descriptors), whereas the nodes with label "..." are not considered in this example.

- **Subtree constraints** (s, N_s, M): the system postpones the expansion of a region group if it belongs to a proper subtree of $\mathbf{T}$ that contains (i) s object hypotheses found consecutively; (ii) M region groups that are not object hypotheses. The constraint (i) is activated after the expansion of N_s region groups (see Fig. 4 for an example).
- **Tree constraint** (K): the expansion of R is stopped if the number of nodes in $T(R)$ exceeds a threshold K.
- **Non-Overlap constraint** (M_o): the object hypotheses resulting from the region grouping procedure may overlap each other. In order to limit the generation of too many overlapping object hypotheses, MEMORI postpones the expansion of nodes corresponding to object hypothesis if they contain a segment which belongs to at least M_o object hypotheses.

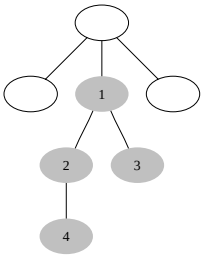

Fig. 4. The colored nodes are object hypothesis found subsequently. If N_s = 1 and s = 4, the expansions of the nodes of this tree are postponed.

The parameters s, N_s, M, K and M_o, named *exploration parameters*, are used to reduce the search space by pruning less promising branches, and to speed up the region grouping procedure. The order for the expansion of the region groups is established by using different priority queues. In particular, MEMORI expands firstly the nodes whose distance $d(R)$ is smaller than a threshold d_o, then the nodes whose area belongs to the range $[A_{min}, A_{max}]$, then the nodes with area smaller than A_{min} and, finally, those satisfying the subtree constraints.

B. Post-Processing

Let α be the scale factor such that $A(R) = \alpha^2 A(O_r)$. Three filtering criteria are applied:

1) **Filter on area:** the solution R is discarded if $A(R) \notin [\alpha_1^2 A(O_r), \alpha_2^2 A(O_r)]$.
2) **Filter on elongation:** the solution R is discarded if $|e_1(R) - \alpha e_1(O_r)| > \tau$ or $|e_2(R) - \alpha e_2(O_r)| > \tau$, where τ is a tolerance factor which depends on the noise along the border of the object caused by the segmentation algorithm.
3) **Filter on overlapping solutions:** if different solutions overlap, only the one with the lowest distance $D(R)$ is retained.

III. Automatic Parameters Estimation

The structure of the parameter estimation phase is depicted in Fig. 5, and consists of four modules, described in the following subsections.

A. Module 1

The first module performs the selection from the input database of the *most relevant* views in terms of recognition accuracy.

Let O be an object, let $O_1, \ldots, O_m$ indicate its views in the input database, and let a be the accuracy of the recognition of O computed by means of a leave-one-out criterion.
The views $O_{i_1}, \ldots, O_{i_k}$ with $i_1, \ldots, i_k \in \{1, \ldots, m\}$ are τ_a-*relevant* for O with respect to a parameter τ_a in $[0, 1]$ iff by replacing in the database $O_1, \ldots, O_m$ with $O_{i_1}, \ldots, O_{i_k}$, the new recognition accuracy a' is such that $a' \geq a - \tau_a$.

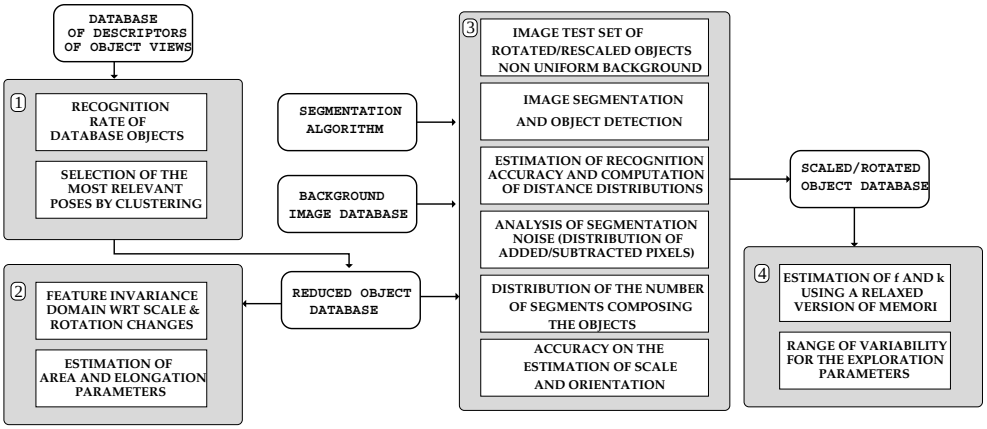

Fig. 5. The four modules of the parameter estimation phase.

We refer to the τ_a-relevant views as *the most relevant* if they represent the smallest subset, that we call *reduced database*.

The selection of the most relevant views permits to reduce the number of items of the database and to speed up the estimation of the parameters (see [13]). It is particularly important when the system has to manage large databases and/or when the object views are very similar to each other, as in the case of objects with many symmetries. Two examples are shown in Fig. 6.

The reduced database is computed by the Algorithm 1. For each object, the set of the most relevant views is extracted by clustering their descriptor vectors by the k-means algorithm. Given an object O, it groups its views in k clusters, with $1 \leq k \leq m$. The number k of clusters is not fixed a priori, but it initialized to 1 (c = 1) and its incrementation depends on the *cluster distortion*, defined as follows: let C be a cluster with center $\mathbf{c}$ and elements $\mathbf{e_1}, \ldots, \mathbf{e_z}$. The *distortion of* C is defined as

$$\Delta_C = \max_{i \in \{1,\ldots,z\}} \parallel \mathbf{c} - \mathbf{e_i} \parallel,$$

where $\|\|$ indicates the L^1 norm.

Algorithm 1 View Selection

```
for all object class O do
  a' = 0, t = 0;
  while (a' < a − τ_a and t < 1.0) do
    c = 1;
    while (c ≤ m and Δ_C > Δ_O) do
      k-means (O, c, Δ_O);
      Compute Δ_C;
      c = c + 1;
    end while
    Compute a';
    Increment t;
    Update(Δ_O, t);
  end while
end for
```

The range of variability of the threshold Δ_O for the cluster distortion is computed automatically and it is defined as follows: let O_i be a view of an object O. Let O_i^{ok} indicate the first view of O returned by the object recognizer, such that $O_i \neq O_i^{ok}$ and let d_i^{ok} be its recognition distance. Let W_i^w indicate the first view not belonging to the object O and returned by the object classifier, and let d_i^w be its recognition distance. The variability set of Δ_O is the range $[d_s, d_e]$, where

$$d_s = \min\{d_O^{ok}, d_O^w\}, \qquad d_e = \max\{d_O^{ok}, d_O^w\},$$
$$d_O^{ok} = \min_i\{d_i^{ok}\}, \qquad d_O^w = \min_i\{d_i^w\}$$

and $i = 1, \ldots, m$.

In the selection of the most relevant views, Δ_O assumes the initial value d_e and it is decremented by $(d_e - d_s)t$ with $t = 0, 1/h, 2/h, \ldots, 1.0$, and h an integer number (by default, h = 10). This is implemented by function `Update`(Δ_O, t). The clustering is re-calculated until the new recognition accuracy a' is greater than $a - \tau_a$.

For each object, the system returns the values of the accuracies a and a'. Moreover, it computes the number of object views of the input database that are correctly recognized by using the reduced database as reference set.

B. Module 2

In Module 2, the system tests the capability of the object recognizer in recognizing rescaled/and or rotated versions of the database elements, and estimates the parameters A_{min}, A_{max}, ϵ_1, ϵ_2.

Let $\varphi_{\alpha,\theta}$ be the composition of the change of scale by a factor α and the rotation in the image plane of angle θ. The system applies to each relevant view a set of 121 transformations $\varphi_{\alpha,\theta}$, with $\alpha \in \{0.2 + 0.18i\}_{i=0,\ldots,10}$, $\theta \in \{5^o + 35^o j\}_{j=0,\ldots,10}$.

The transformation of the view O_j of object O by $\varphi_{\alpha,\theta}$ is *recognized* iff $\varphi_{\alpha,\theta}(O_j)$ is *classified* as a view of O, i.e. iff the database element returned by the object recognizer is a view of O. Indicated by ρ the recognition rate, the system computes the domain Ω of the feature invariance with respect to changes of scale factor and/or orientation in the image plane, defined as

$$\Omega = \{(\alpha, \theta) : \rho \geq 1 - \varepsilon\},$$

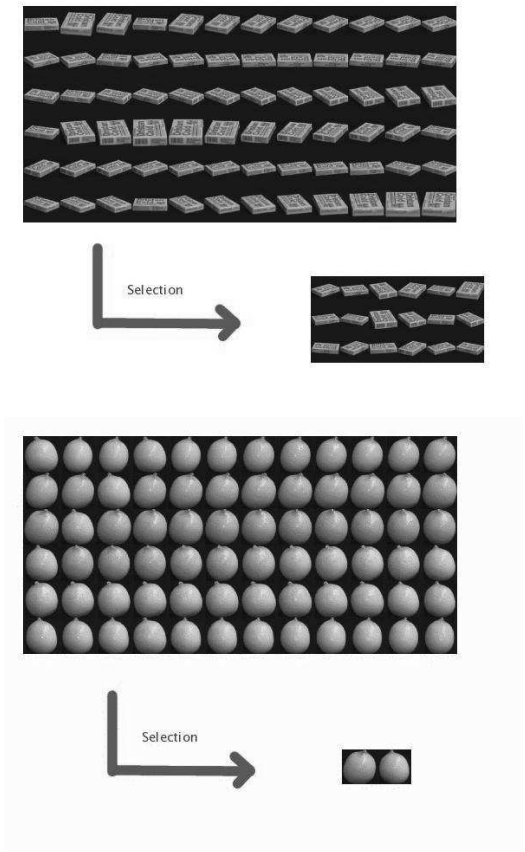

Fig. 6. Two examples of selection of the most relevant views.

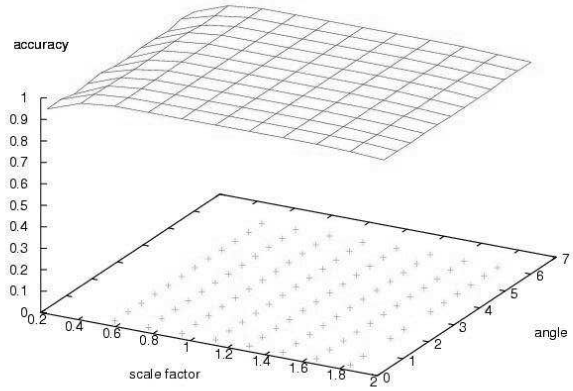

Fig. 7. Accuracy of recognition of the objects transformed by scale change and rotation for the objects of COIL database [16]. Ω is indicated by the symbol "$\times$" on the x-y plane.

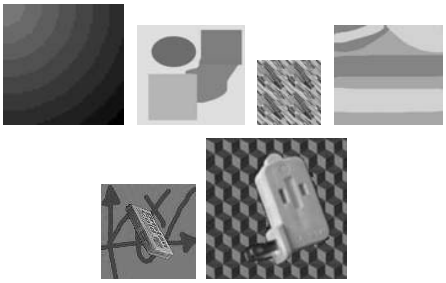

Fig. 8. Some examples of synthetic non uniform background used for testing the robustness of the descriptors with respect to segmentation noise (first row), with an embedded COIL object (second row).

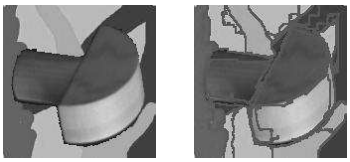

Fig. 9. A view of an object is firstly rotated and rescaled, then displayed on a cluttered background (on left) and finally it is segmented (on right) in order to extract the number of regions composing the transformed segmented object view.

where ε is a real number in [0,1] that can be selected by the user. The parameters α_1 and α_2 are set to the minimum and the maximum value of α in Ω.
The elongation parameters ϵ_1 and ϵ_2 are set to maximum value of $\alpha_1 e_1(O_j)$ and $\alpha_2 e_2(O_j)$, respectively, over the views O_j and the database objects.

C. Module 3

Module 3 is devoted to: (i) the computation of the distributions of L, d_o, d_g, D_g, (ii) the evaluation of the segmentation noise and of τ, (iii) the estimation of the recognition performance on the database, and of the accuracy of determination of scale and orientation of the recognized objects.

The segmentation [8] often returns over-segmented objects, differing from the original database object by the addition or/and subtraction of some pixels along the borders. Therefore it is necessary to evaluate the robustness of the descriptors with respect to this noise introduced by the segmentation.

This analysis is carried out by considering a set of non uniform background images (Fig. 8) on which the objects may be placed. For each relevant object view O_j, a set of 100 transformations $\varphi_{\alpha,\theta}$ with α and θ randomly chosen in Ω, is computed. Each transformed object view is laid on a background image, randomly selected. Let $\mathcal{I}$ indicate the set of images built in this way. At this stage, a priori knowledge about the scenario where the objects can appear, could be used to select the most appropriate backgrounds, or even to insert a new suitable background.

Each image of $\mathcal{I}$ is segmented and the transformed object view $\varphi_{\alpha,\theta}(O_j)$ contained in is generally divided in more parts (see Fig. 9). Let $\mathcal{P} = \{P_1, \ldots, P_m\}$ be the set of regions obtained from the segmentation of an image. The transformed segmented view is obtained by merging the regions $P_{i_1}, \ldots, P_{i_y}$ of $\mathcal{P}$, maximizing the *weighted overlap index*

$$\frac{A(Q)}{A(Q) + w_1 + w_2} \tag{1}$$

with $Q := \varphi_{\alpha,\theta}(O_j) \cap P$, and $P = \cup_{j=1}^{y} P_{i_j}$.
We define then the sets $D_1 := \varphi(O_j) - P$, $D_2 := P - \varphi_{\alpha,\theta}(O_j)$, whose cardinality is the numbers of the pixels subtracted from and added to the object view it by the segmentation. The weights w_1 and w_2 measure the integral of the distance of the points of D_1, D_2 from the border $\partial\varphi_{\alpha,\theta}(O_j)$ of $\varphi_{\alpha,\theta}(O_j)$: the closer to the borders are the pixels of D_i, the smaller is w_i, $i = 1, 2$. More precisely,

$$w_i = \sum_{x \in D_i} \nabla(\partial\varphi_{\alpha,\theta}(O_j), x), \quad i = 1, 2$$

where $\nabla(\partial\varphi_{\alpha,\theta}(O_j), x)$ is the city-block distance of pixel x from the $\partial\varphi_{\alpha,\theta}(O_j)$.

Recognition distances and performance with respect to segmentation noise: the views transformed and separated from background by the maximization of the overlap index (1), are submitted to the object recognizer. The system computes the distributions of its distances (with and without shape features) for the views correctly classified. The parameters d_o, d_g, D_g, are set by selecting a percentile of their distributions; default values for the percentiles are the 75th percentile for d_o, and the 99th percentile for d_g and D_g. The system computes also the percentage of recognized transformed object views.

Analysis of the segmentation noise : the segmentation noise is measured by computing the distributions of the percentage of the pixels added and subtracted to the objects in the images of $\mathcal{I}$. The parameter τ is determined as the maximum between two percentiles of these two distributions (99.8th percentiles as default).

Maximum number of regions composing an object: the distribution of the number of segments composing the transformed object views in the images of $\mathcal{I}$ is returned and L is fixed as a percentile (the 99th as default) of such distribution.

Accuracy on scale factor and orientation: for each object hypothesis R, the scale factor and the rotation angle in the image plane with respect to O_r are computed. This is done by calculating the vertical and the horizontal shifts s_v and s_h of the log-polar transformation $\mathcal{L}(\varphi_{\alpha,\theta}(O_r))$ of O_r on the log-polar transformation $\mathcal{L}(R)$ of R, such that $\mathcal{L}(\varphi_{\alpha,\theta}(O_r))$ and $\mathcal{L}(R)$ match as well as possible. The anti-transform of s_v and s_h give the rotation angle and the scale of R. The errors on scale and rotation angle for an object view $\varphi_{\alpha,\theta}(O_r)$ contained in an image of $\mathcal{I}$, are defined as

$$\begin{aligned} E_{scale}^{\varphi_{\alpha,\theta}(O_r)} &= |\alpha - \alpha_{lp}| \\ E_{angle}^{\varphi_{\alpha,\theta}(O_r)} &= \min\{|\theta - \theta_{lp}|, 2\pi - |\theta - \theta_{lp}|\}, \end{aligned}$$

where α_{lp} and θ_{lp} are the scale factor and the angle computed by aligning the log-polar transformations. These error distributions provide an indication of the accuracy of such estimation.

D. Module 4

Once the value of d_o and A_{min} have been estimated, the system computes the distributions of f and k by running on the images of $\mathcal{I}$ a *relaxed* version of the region grouping algorithm, in which only the constraint on area is activated.

Fig. 10. Two objects of COIL-100.

For each image in $\mathcal{I}$, let $\mathcal{B}$ be the path of the tree $\mathbf{T}$ connecting the region group correspondent to the correct object view to the root. For each node N on $\mathcal{B}$, $f_b = d(R)/d_b$ and the number k_b of ancestors of R such that $f_b \geq f$ are computed. f and k are then set by choosing a percentile of the distributions of f_b and k_b (99th percentile as default for both parameters).

The ranges of variability for the exploration parameters are:

$$\begin{aligned} K \geq L, N_s \in \{1, \ldots, K\}, s \in \{L - s, \ldots, K\} \\ M \in \{1, \ldots, K\}, M_o \in \{1, \ldots, KN_{seg}\} \end{aligned}$$

where N_{seg} is the number of segments output by the segmentation of the input image.

In summary, the user has to provide only the following data: the database of objects to be recognized, ε, τ_a, the percentiles for fixing the parameters L, d_o, d_g, D_g, f, k, and the values of the exploration parameters (in the provided ranges).

IV. Experimental Results

The database of objects chosen for testing the recognition and detection performances of MEMORI with the values of the parameters estimated automatically, is the COIL-100 database [16]. It consists of 7200 images of 100 objects, 72 different views for each object. Some examples of object views are shown in Fig. 10. The accuracy of the object recognizer on the images of COIL-100 is 0.9983 (12 errors). The recognition rate on the database object views by using the most relevant views as reference database rises to only 1 error.
The feature invariance domain Ω has been computed by setting $\varepsilon = 0.01$ (Fig. 7). The most relevant views have been selected by using the default value $\tau_a = 0.01$, and it contains 942 images. The recognition rate on the $94,200$ images in $\mathcal{I}$, segmented by algorithm [8], is 0.9648.

Table II reports the values of the system parameters used in the experiments. Values for L, d_o, d_g, D_g, f, k and τ have been selected by using the default percentiles reported above. The distributions used to compute the parameters, the distribution of the segmentation noise and the error on the estimation on scale factor and orientation, have been reported in Fig. 11, 12, 13.

A test set of 100 synthetic images [17] has been created by drawing rotated and rescaled version of some COIL objects on a non uniform background (see Fig. 14). The objects inserted in the images (735 in total) have been randomly chosen as well as the rotation angle and the scale (in Ω).

MEMORI performances are tested by evaluating the detection and recognition accuracy. A view O_j contained in a ground truth image is *well detected* with respect to a real parameter

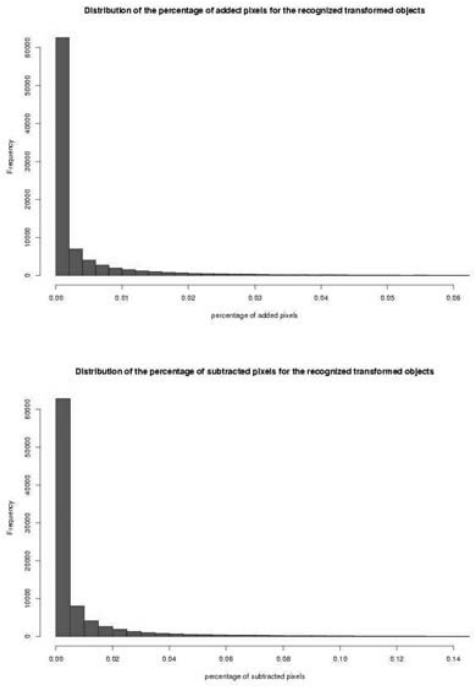

Fig. 11. Pixel addition (top) and subtraction (bottom) due the segmentation [8].

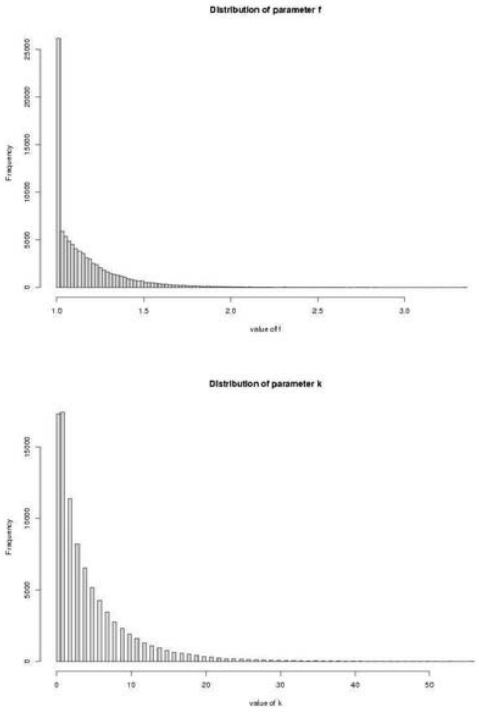

Fig. 13. Distribution of parameters f and k.

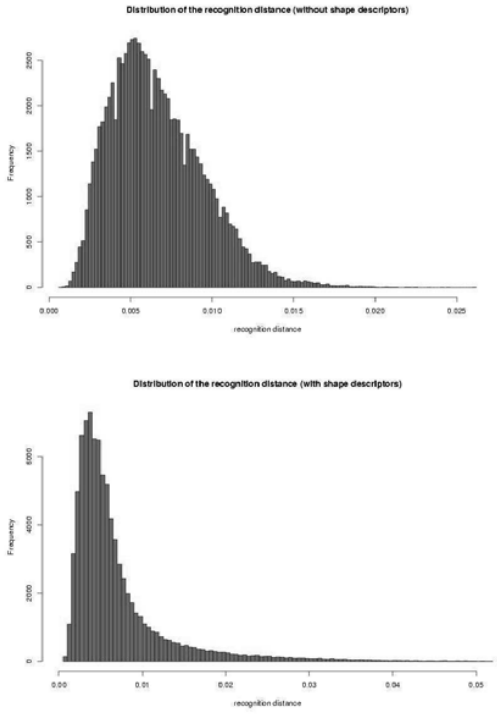

Fig. 12. Distribution of the recognition distances without shape descriptors (top) and with shape descriptors (bottom).

TABLE II
VALUES OF THE SYSTEM PARAMETERS.

Parameter	Value
α_1	0.56
α_2	1.82
A_{min}	588.31
A_{max}	50477
ϵ_1	229.32
ϵ_2	297.54
L	44
τ	20
d_o	0.009

Parameter	Value
d_g	0.015
D_g	0.033
f	2.04
k	30
s	43
N_s	1
K	120
M	15
M_o	20

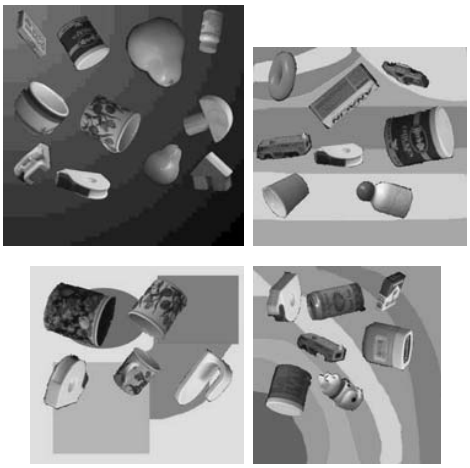

Fig. 14. Some images of the test set [17] used in the experiments.

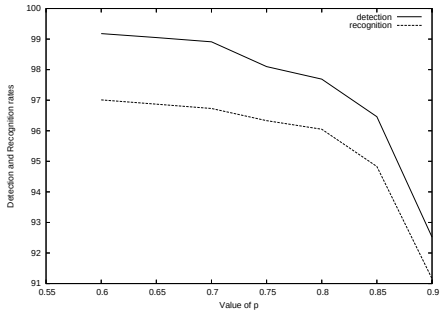

Fig. 15. Detection and recognition percentage versus p for the test images.

TABLE III
SYSTEM PERFORMANCES ON THE 100 IMAGES TEST SET AT DIFFERENT VALUES OF p: D = DETECTION, R = RECOGNITION, PD = PARTIAL DETECTION, M = MISS DETECTION, FP = FALSE POSITIVE, E_α, E_θ = AVERAGE ERRORS IN THE ESTIMATION OF SCALE AND ROTATION

p	D	R	PD	M	FP	E_{scale}	E_{angle}
0.60	99.18	97.01	0.54	0.27	0	0.0132	0.0278
0.65	99.05	96.87	0.68	0.27	0	0.0129	0.0270
0.70	98.91	96.73	0.82	0.27	0	0.0128	0.0269
0.75	98.10	96.33	1.63	0.27	0	0.0125	0.0265
0.80	97.69	96.05	2.04	0.27	0	0.0123	0.0256
0.85	96.46	94.83	3.27	0.27	0	0.0117	0.0245
0.90	92.52	91.16	7.21	0.27	0	0.0105	0.0186

$p \in [0, 1]$ if there exists an object hypothesis R such that the overlap index

$$\mu := \frac{A(O_j \cap R)}{A(O_j \cup R)}$$

is greater than p. The closer μ is to one, better the detection is. If O_j contains R, or R contains O_j but their overlap factor is lower than p, the object O_j is only *partially detected*. Otherwise O_j is *missed*. O_j is *recognized* if it is well detected and the object view corresponding to the result R is a view of O.

The results of the experiments are reported in Table III as a function of the parameter p. The percentages of detection and recognition in the test images are also plotted in Fig. 15. The average error in the estimation of the scale factor (E_{scale}) and of the orientation angle (E_{angle}, in radians) of the recognized objects are reported in Table III.

The dependency of the system performance on the exploration parameters is shown in the Figures 16, 17, 18, 19. It shows that the system performances are relatively robust with respect to changes in the exploration parameters.

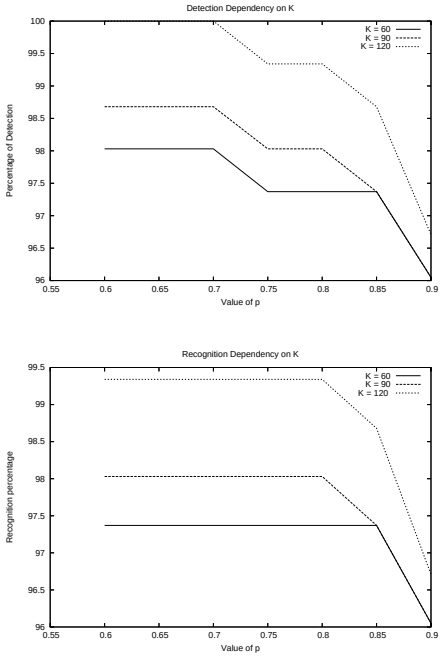

Fig. 16. Graphs of Detection and recognition dependency on K.

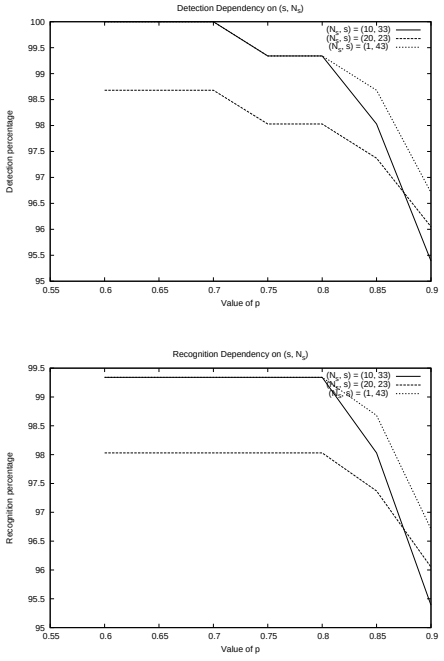

Fig. 17. Graphs of Detection and recognition dependency on (s, N_s).

V. CONCLUSIONS

The experiments show that MEMORI (whose interface is shown in Fig. 20) is a promising tool for the detection and recognition of the objects of a database within color digital images. The main advantages in the use of MEMORI are the adaptability to the object database, the capability to recognize rotated and/or rescaled objects, and the reduced user interaction. In comparison with the previous versions of the system [3], [12], the current one (MEMORI 1.1) automatically estimates the rules parameters and it is remarkably faster without to affect the performances. MEMORI 1.1 provides also information about its performances in object recognition and determination on scale factor and orientation.

Our future work will target the elaboration of a strategy to determine more restrictive values for the exploration pa-

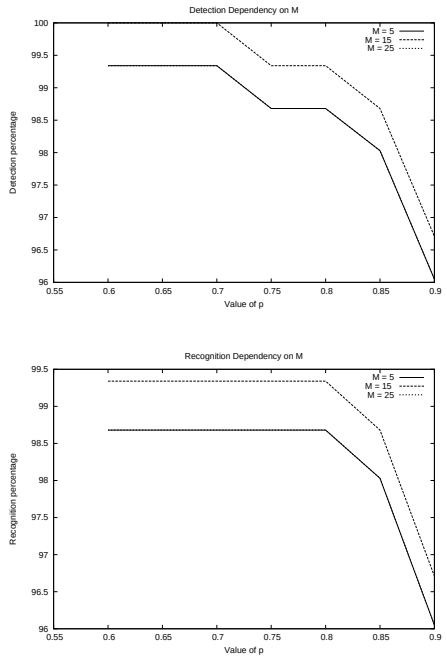

Fig. 18. Graphs of Detection and recognition dependency on M.

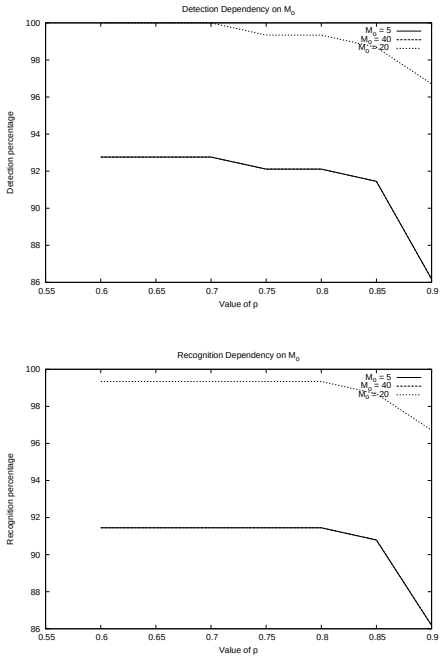

Fig. 19. Graphs of Detection and recognition dependency on Mo.

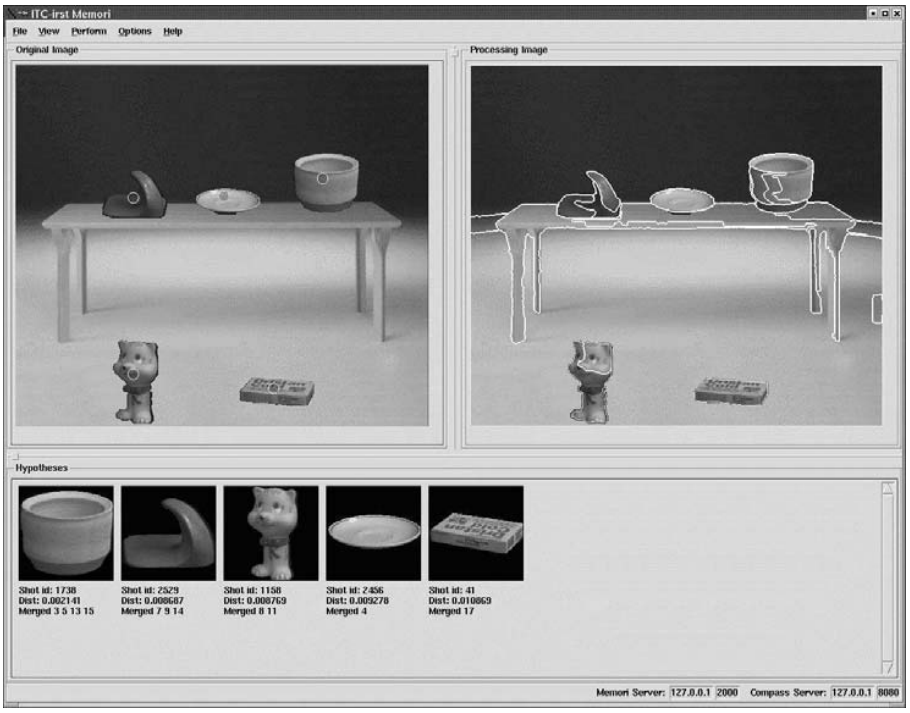

Fig. 20. MEMORI's interface: on left, the input image with the recognized objects highlighted by spots; on right, the segmented image, which it is possible to visualize the region groups corresponding to the recognized objects in; on bottom, the recognized objects are shown along with their identifier number in the input database, the recognition distance computed by considering all the descriptors, and the sequence of merged regions; the scale factor and the orientation angle can be visualized by using the menu options.

rameters, the introduction of new techniques to make the system robust with respect to illumination changes and able to recognize partially occluded objects. MEMORI will then be used as an indexing tool in a novel and effective content-based image retrieval system.

ACKNOWLEDGMENT

M. Lecca thanks S. Messelodi, R. Brunelli, and C. M. Modena for the many discussions and suggestions about this work.

REFERENCES

[1] C. Andreatta. CBIR techniques for object recognition. Technical Report T04-12-01, ITC-irst, Povo, Trento, Italy, December 2004.
[2] C. Andreatta, M. Lecca, and S. Messelodi. Memory-based object recognition in images. Technical Report N. T04-12-06, ITC -irst, December 2004
[3] C. Andreatta, M. Lecca, and S. Messelodi. Memory-based object recognition in images. In *10th International Fall Workshop - Vision, Modelling, and Visualization - VMV 2005*, 2005.
[4] R. Brunelli and O. Mich. Image retrieval by examples. *IEEE Transactions on Multimedia*, 2(3):164–171, 2000.
[5] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object Recognition as Machine Translation: Learning a lexicon for a fixed image vocabulary. In *European Conference on Computer Vision (ECCV) Copenhagen*, 2002.
[6] D. I. Moldovan, and C.-I. Wu. A Hierarchical Knowledge Based System for Airplane Classification. IEEE Transactions on Software Engineering, 2004, Vol. 14, N. 12, pp. 1828 – 1834
[7] O. Carmichael, and M. Hebert. Shape-based Recognition Of Wiry Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, Vol. 26, pp. 1537–1552
[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, 2004.
[9] D. A. Forsyth and J. Ponce. *Computer Vision: a modern approach*. Prentice Hall, 2002.
[10] A. Hoogs, R. Collins, R. Kaucic, and J. Mundy. A common set of perceptual observables for grouping, figure - ground discrimination, and texture classification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, (4):458–474, 2003.
[11] B. Ko and H. Byun. Extracting Salient Regions And Learning Importance Scores In Region-Based Image Retrieval. *International Journal of Patter Recognition and Artificial Intelligence*, (17(8)):1349–1367, 2003.
[12] M. Lecca. MEMORI - version 1.0. Technical Report T05-10-01, ITC - irst, Centro per la Ricerca Scientifica e Tecnologica, October 2005.
[13] M. Lecca. A new method for the automatic estimation of the heuristic rule parameters for MEMORI 1.0. Technical Report T05-12-01, ITC - irst, Centro per la Ricerca Scientifica e Tecnologica, December 2005.
[14] M. Lecca. A Self Configuring System for Object Recognition in Color Images. Proceedings of 12th International Conference on Computer Science, March 2006
[15] D. I. Moldovan and C.-I. Wu. A hierarchical knowledge based system for airplane classification. *IEEE Transactions on Software Engineering*, 14(12):1829–1834, 1988.
[16] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-100). In *Technical Report CUCS-006-96, Columbia University*, 1996.
[17] M. Lecca. Test Set GroundTruth100-for-COIL, `http://tev.itc.it/DATABASES/objects.html`