Electronic Letters on Computer Vision and Image Analysis 8(2):68-83, 2009

An Attention Module for Object Detection in Cluttered Images

Michela Lecca*

* *Fondazione Bruno Kessler, Center for Scientific and Technological Research, via Sommarive 18, Povo, 38100 Trento, Italy*

Received 12th January 2009; accepted 18th July 2009.

Abstract

In this paper, we propose a visual attention module that automatically detects the regions of an input previously unseen image, which are *more likely* occupied by a known object. The module can be integrated in many object recognition systems for reducing the image space in which to search the object, and the computational costs. The strategy has been tested on two public real-world image databases showing good performances. Moreover, we measured the usefulness of this selective visual attention by comparing the performances of the SIFT recognition algorithm with and without the proposed attention module.

1 Introduction

The recognition of the objects of a given database in an unknown image consists in finding a correspondence between features describing the objects and the image. In the last years many approaches have been developed, e.g. [17], [20], [18], [19], [22], [30], but a completely automatic efficient system for object recognition is not yet available. One of the main difficulties in automatic object recognition is the lack of information about the presence, the position and the number of occurrences of the objects in the image. This results in the necessity of extending the object search to the whole image and consequently in a long time for the feature extraction and matching. Pruning strategies - typically relating to geometric and appearance-based constraints - are particularly necessary when the algorithm complexity depends on the number of image pixels or regions to be analyzed [1], [17], [19], [18], [20].

In the human visual system, a *visual attention* mechanism allows to rapidly detect the location of the most interesting components of the seen scene [5], [14], [28]. Human visual attention consists of two complementary processes, which generally work in parallel. In the first one, named *bottom-up* attention, human attention is involuntarily attracted by some visually *salient* features, like contrasts, luminance, brilliant colors, direction and speed of the motion. A *salient map* encoding these conspicuous stimuli is automatically computed in less than 50ms in the early visual cortex area. In the second process, called *top-down* attention, the human visual system focuses on specific locations or objects in the scene, depending on the task at hand, like for instance to establish the presence and the position of a certain object in a room or to recognize a person. The top-down attention is controlled by a complex brain network, that connects the cognitive areas to the early visual cortex, and differently from the bottom-up attention it requires voluntary efforts and more time (about 200 ms per scene). Understanding how the visual attention mechanism works is an attractive still unsolved challenge not only in Neuroscience but also in Computer Vision, where tasks such as surveillance, object recognition or semantic image annotation, could take advantages by simulating this human capability [2], [7], [27]. Therefore, many computational models of visual attention have been developed [11], and many works show that simulating

Correspondence to: <lecca@fbk.eu>

Recommended for acceptance by <Giorgio Fumera>
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

visual attention improves the accuracy and the efficiency of the software tools for object recognition in images [3], [9], [24].

In this work, we present a novel simple top-down attention module, that can be easily integrated in any system for the detection and recognition of objects in color images with cluttered background. The module selects the image portions where to find the objects is more probable. Even if the problem has been addressed by other methods, as far as we know, the exact approach we present here has never published previously. We represent each object O to be detected by a multi-view model, i.e. by many color pictures portraying it from different points of view. For each of these pictures, the set of pixels corresponding to the object (i.e. the view) is specified by a binary mask. We describe each object view and the input (previously unseen) image I by parts, called *covers*, and we compute a *correspondence map*, that assigns to each cover of I a *confidence* to be occupied by a cover of O. The confidence is expressed in terms of visual similarity between the considered covers. More precisely, each view and image cover is described by a vector of low-level features encoding information about color, texture and edge distributions. The correspondence map associates each cover C of I to the object cover C_O that is visually the most similar to C. The visual similarity between the covers is defined as the L^1 distance between their descriptor vectors. A threshold τ on the visual similarity is then used to find out the regions (if any) of I where O is more likely positioned.

Three are the novelties of our method: firstly, we represent the objects and the image by parts *topologically* defined and we describe each part by *global* low-level descriptors; secondly, we define an object-image *correspondence map*, that specifies for each object a *confidence level* to occupy a given image part; finally, in order to reduce the user interaction, we propose an *automatic estimate* of the main parameters used in our approach.

The description of an object or image by parts is not new: many popular methods, e. g. [8], [19], [22], [23] represent the objects and the image by sets of pixels, termed *interest* or *key points*. These are characterized by *salient* features that allow a good recognition of the object which they belong to, like pixel brightness, color, high curvature, gradient orientation. In some cases [6], [8], [13], [15], [22], groups of adjacent interest points (*interest regions*) are considered. The detection of the key points simulates a bottom-up mechanism tailored to object recognition, but this method is strongly dependent on the choice of the salient features, that are not easy to be determined. Often supervised learning strategies are namely employed for their selection [16], [26].
Unlike the saliency-based methods, our approach does not implement any bottom up attention strategy, as it avoids the definition and the extraction of the features that are the most relevant for the recognition task. In fact, the covers are not defined by their visual properties, but by the following *topological* condition: a cover of an object view (or of an image) is the intersection set between a circle and the view (or of the image). Changing the radius and the center coordinates of the circles, we can define many different sets of covers. Although the radius and the center as well as the threshold τ can be entered by the user, we developed and implemented a technique to estimate them automatically. Therefore, the task to direct the procedure to efficiently recognize the objects is not completely left to the user, that often does not have a precise idea about the optimal values of the system parameters. The correspondence map we compute makes the recognition process more efficient: it allows to circumscribe the detection process to some image areas, and establishes a priority on the order of the image portions where the objects have to searched, and of the objects to be detected (from the most to the less probable). The experiments carried out on two public real-world datasets shows that on average our approach restricts the search for an object to the 50 % of the area of the input image. This finally results in a reduction of the computational time for the recognition.

In order to demonstrate the effectiveness of our visual attention module, we have integrated it into the well known algorithm SIFT [19]. This achieves object recognition by selecting and matching scale-, illuminant- and noise- invariant key points extracted from the objects as well as from the input image. Our selection of the image regions with high confidence to contain an object reduces of the 47% the number of key points of the image to be matched with the key points of the object, without affecting the SIFT performances.

Synopsis - Section 2 explains how to compute and to describe the object and image covers, while Section 3 defines the correspondence map. Section 4 illustrates the automatic estimates of the parameters for the cover computation. Section 5 presents the performance evaluation experiments. Section 6 proposes some applications

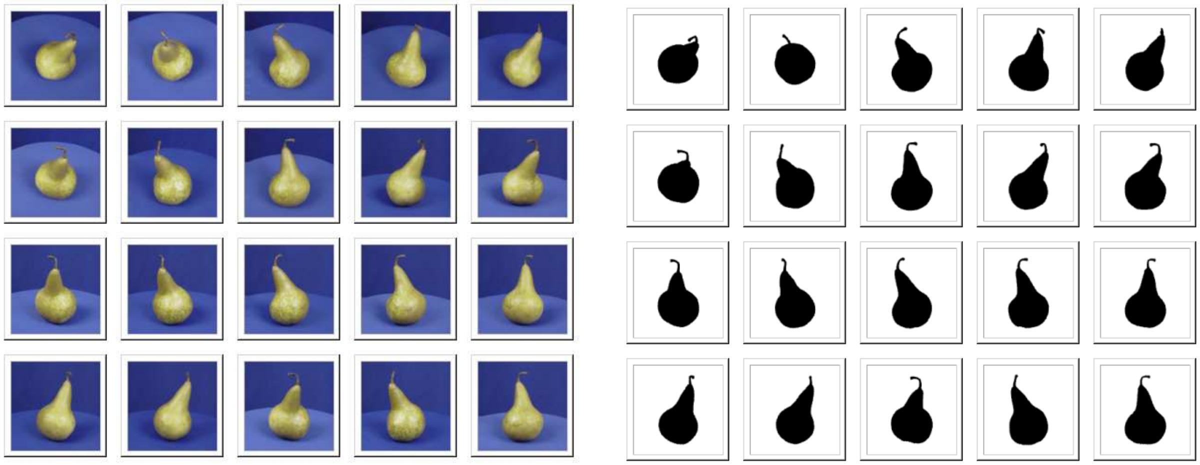

Figure 1: An object (here a pear) is represented by many 2D images depicting it from different viewpoints. For each image, the part actually correspondent to the object is defined by a binary image: the black pixels belong to the object and form the support of the view, while the white pixels belong to the background.

of our approach in object recognition. Section 7 presents a comparison of our approach with one saliency-based method. Finally, Section 8 contains our concluding remarks and outlines our future work.

2 Computation and Description of Object and Image Covers

Let D be the database of the objects to be searched for. Each object O in D is modeled through a set of 2D color pictures portraying it from different points of view. The object view depicted in one of these pictures is the portion of the picture that actually corresponds to the object, while the rest of the picture is regarded as transparent. In practice, to each picture is associated a binary image, where the black pixels refer to the object view and the white pixels to the background. Figure 1 shows such a model of an object.
Hereafter we refer to the set of coordinates of a certain portion P of an image I as the *support* of P. Therefore, the support of a view v, here denoted by supp(v), is the set of coordinates of the black region in the correspondent binary image, while the support of an image I is the set of coordinates of the pixels of I.

A cover of an object view (or of an image) is defined as the portion of the view (or of the image) whose support is the intersection set between a circle and the support of the view (or of the image). A circle intersecting a support of a view is considered in the cover generation only if (R1) the circle does not contain the whole view support and (R2) the ratio between the overlapped area and the cover area is greater than a threshold μ. These conditions constrain the length of the radius and the position of the center of the cover. In particular, rule (R1) forbids the generation of a trivial cover whose support coincides with the whole view support, while rule (R2) controls the percentage of area of the view support that is covered by the circle and then avoids the creation of too small covers. The radius R of the circle is input by the user or estimated automatically by the procedure detailed in Section 4 and it is the same for each database view. Since an object can appear in an image with a size different from that of the model in the database, covers with different radii $\lambda_j R$, j = 1, $\ldots$, m, are generated, where λ_j is a scale factor.
A cover of an image is similarly defined by the rules (R1) and (R2). In this case, rule (R1) states that the circle must not contain the whole image support. The radius of the image covers is R.
We note that the covers of an object view or of an image are general not disjoint, i.e. they do not form a partition of the view or of the image.
Each view and image cover is described by a vector of low-level descriptors, so that the visual similarity

between a cover of a view and a cover of an image is expressed by the L^1 distance between their feature vectors. The more probable positions of an object in an image are determined by comparing each image cover with the multi-scale covers of the object to be searched for.

Subsections 2.1 and 2.2 contain the formal definitions of object and image covers, while Subsection 2.3 explains how the covers are described.

2.1 Object Covers

Let O be an object of D, and let v be a reference view of O. A cover of v is the intersection set between a circle and supp(v). More precisely, the cover of v with center p_k and radius R is the portion of v whose pixels belong to the set $B(p_k, R) \cap$ supp(v) where $B(p_k, R)$ is the circle with center p_k and radius R and $B(p_k, R) \cap$ supp$(v) \neq \emptyset$. Moreover, by rule (R1), supp(v) is not properly included in $B(p_k, R)$, i.e. supp$(v) \not\subset B(p_k, R)$.

The computation of a set $\mathcal{C}(v)$ of covers for each object view v is implemented as follows:

1. Compute the minimum bounding rectangle MBR(supp(v)) of supp(v), i.e. the rectangle including supp(v) and having the smallest area;

2. Fix a regular grid $G(v)$ = $\{p_k \in$ MBR(supp(v))$\}_{k \in K}$ on MBR(supp(v)), with K a set of indices;

3. Define $\mathcal{C}(v)$ = {portion of v with support $B(p_k, R) \cap$ supp(v) : $p_k \in G(v), \mu_k \geq \mu\}_{k \in K}$, where

$$\mu_k := \frac{A(B(p_k, R) \cap \text{supp}(v))}{A(B(p_k, R))} = \frac{1}{\pi R^2} A(B(p_k, R) \cap \text{supp}(v)), \tag{1}$$

   $A(\cdot)$ denotes the area of the region specified within the brackets and μ is a real number in (0,1].

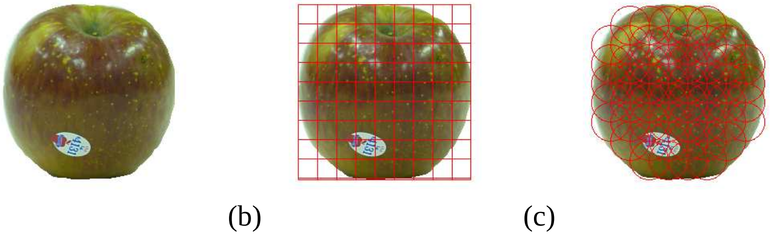
(a) (b) (c)

Figure 2: Steps for the computation of the cover of the reference view (a): a grid is fixed on the MBR of the view support and superimposed to the view (b); the circular regions extracted are shown in (c).

An example of covers for an object view is shown in Figure 2.

The cover set of a view v depends on the values of μ and R and on the grid superimposed on MBR(supp(v)). These parameters are user inputs and determine the existence of a cover set for v. For instance, large values of R make the condition $\mu_k \geq \mu$ false for each node on the grid, so that no covers can be generated. For $\mu = 1.0$, only circles completely included in supp(v) are taken as covers.

Hereafter we assume that the objects in the database have similar aspect ratios. Otherwise, choosing a unique radius R for the covers of all the database objects is not recommendable, because thin objects can be covered only by covers with circles with diameter equal to the object thickness. By using the radius of these circles for covering bigger objects, we would obtain a lot of covers, and the features describing the most of them could not allow a good object detection. To overcome this problem, when a database contains objects with different aspect ratios, we cluster the items by the aspect ratio and we consider each cluster as a database of objects with an uniform aspect ratio. Given an input image, we compute a correspondence map for each of these databases.

However, in this work, we focus on databases with homogeneous aspect ratio.
The value μ_k defines the rule (R2). It measures the percentage of area of v covered by the circle $B(p_k, R)$. If μ_k is zero, the circle $B(p_k, R)$ does not intersect supp(v); a small value of μ_k indicates that $B(p_k, R)$ covers only a small part of supp(v), like in the case of circles intersecting parts of supp(v) near to its boundary. The threshold μ is an user input, and it is used to exclude the covers with a small intersection area with the view. These covers are in fact not relevant for the detection task. A typical variability range of μ is [0.7, 1.0]. In this work, we set μ = 0.75.
In principle, the user could choose any non zero positive value for R, but, as discussed above, the existence of a database cover set is not guaranteed for each value of R. In this work we propose two formulas for computing R in order to generate non empty cover set for the objects. The formulas are relating to the geometric properties of the objects in the database.
For databases whose objects have an aspect ratio ρ close to 1.0 (in this work, $\rho \geq 0.6$), the value of the cover radius R is computed by the following formula:

$$R = \frac{1}{N} \min_{O \in D} \left\{ \sqrt{\frac{A(\text{supp}(v))}{\pi}}, v \text{ view of } O \right\} \tag{2}$$

where N is an integer strictly positive number that is fixed by the user. The smaller N is, the larger the number of covers for the object views is. In case of thin objects (here $\rho < 0.6$), the radius is set up as

$$R = \frac{1}{2N} \min_{O \in D} \left\{ t(\text{supp}(v)), v \text{ view of } O \right\} \tag{3}$$

and $t(\text{supp}(v))$ is the average thickness of v.
Specifying a value for R such that the existence of a cover set and good detection performances are guaranteed could be a difficult task for an user, especially without having information about the objects. Formulas (2) and (3) constraint the value of R to the aspect ratio and to the area or thickness of the database views. By these equations, the user fixes the value of R through N as portion of a function of the area or of the thickness of the database views. Qualitatively, for the user, fixing N instead of R is simpler than entering a numerical value for R: for $N = 1$, R is the radius of the circle covering the smallest object view (by formula (2)) or one half of the smallest average thickness (by formula (3)). The user can set up qualitatively the value of N by looking at the database views or just at the smallest and at the greatest views (in terms of area or thickness) and then to fix the value of N that he/she retains the most adequate. However, in Section 4 we present also a method for estimating N and hence R automatically.
In order to deal with changes of scale factor, *multi-resolution covers* of each database object are computed. The user specifies a discrete set $\{\lambda_0, \ldots, \lambda_m\}$ of scale factors with λ_0 = 1.0 and for each j = 0, …, m, the view covers with radius $\lambda_j R$ are computed.
To have the most complete description of v, the nodes $p_k := (x_k, y_k)$ of the grid have to be chosen such that v is almost entirely covered by the union of its covers. We say *almost* because the threshold μ avoids the generation of covers intersecting small portions of v, generally close to the view boundary. In this work, we fix the grids such that

$$|x_{k+1} - x_k| = |y_{k+1} - y_k| = \lambda_j R, \tag{4}$$

for each j = 0, 1, …, m.
Other grids can be considered. In Section 5 we use a coarser grid and we analyze the dependency of the detection accuracy on the grid resolution.

The covers of the object views having radius $\lambda_j R$ with j = 0, …m are said *object covers with basis* R. In the following, we denote by $\mathcal{C}(O)$ the set of all the computed covers of the views of O.

The cardinality of the set of the object covers depends on the number of reference views of the objects and on the parameters $\{p_k\}_k$, μ and R (or N). When many reference views are used in the object representation, the memory space requested for storing the object covers can be very large. In order to limit it and also to speed up the description and matching of the covers, we reduce the number of views modeling each object by means of the clustering algorithm described in [17]. The views of each object are grouped by a k-means algorithm and the centroids of the obtained clusters identify the *relevant views*. Their number varies from object to object and it is determined automatically. Here we do not illustrate this technique, but we remind to [17] for more details.

Figure 3 shows the multi-resolution covers computed for a view of a keyboard. In this example R = 28, λ_0 = 1.0 for the coverage in the middle, λ_1 = 0.7 and λ_2 = 1.3 for the covers on left and right respectively.

2.2 Image Covers

Let D be the database of objects whose covers have basis R and let I be an unknown image where the objects have to be detected. The covers of I are the circles B with radius R, centered at the nodes of a regular grid $G(I)$ fixed on $\mathbf{R^2}$ and such that

$$\mu_I := \frac{A(B \cap \text{supp}(I))}{A(B)} = \frac{1}{\pi R^2} A(B \cap \text{supp}(I)) \geq \mu. \tag{5}$$

The parameter μ is the same used for the computation of the object covers. The parameter μ_I is 1.0 for all the circles B entirely contained in supp(I). It is smaller than 1.0 for the circles that intersect the image support on its borders and zero for those non intersecting the image support. The generation of the border covers is controlled by the threshold μ, that in this work is the same as that used for computing the object covers.

We note that it is not necessary that the nodes of the grid $G(I)$ are spaced like those of the grid used for the database objects. Generally, when a fine (*coarse*, *resp.*) grid has been used in the object covers computation, a coarse (*fine*, *resp.*) grid is computed on the image.

Figure 4 (left) shows a cover set for a picture containing the keyboard of Figure 3. In this case, R = 28 but the grid used for the computation of the image covers is coarser than that one used for covering the model of the keyboard.

2.3 Cover Description

Each cover of an object view or of an image is described by the following low-level features: (*i*) *color*: histograms of hue, intensity, and distribution of the saturation with respect to the hue; (*ii*) *edges*: distribution of the module of the edges detected by the image gradient; (*iii*) *texture*: distributions of bi-dimensional co-occurrence matrices of hue and intensity.

The computation of these descriptors is completely automatic and no user interaction is requested. The features are encoded in a vector and the visual similarity between two covers is measured by the L^1 distance between the correspondent feature vectors. In [4] it has been reported that, for the considered descriptors, this distance gives the best performances in terms of recognition accuracy and computational time. The considered features are invariant to rescaling and in-plane rotations, so that rescaled and/or rotated versions of the database objects can be detected. The visual similarity is normalized in order to range [0,1]. The closer to 0 the distance is, the more similar the compared covers are.

3 Correspondence Maps

The correspondence map relates a cover set of an image with the cover sets of the objects in the database. As mentioned in Section 1, it specifies for each object the confidence to occupy a certain cover of the image.

Let D be a database with n objects O_1, ..., O_n, and let $\mathcal{C}(O_1)$, $\mathcal{C}(O_2)$, ..., $\mathcal{C}(O_n)$ be their cover sets with basis R. Let I be an image and let $\mathcal{C}(I)$ be its cover set with radius R. For each cover c of $C(I)$ we define the *distance* of c from the object O_j (j = 1, $\ldots n$), as

$$d(c, O_j) = \min\{\delta(c, c_v) : c_v \in \mathcal{C}(O_j)\} \tag{6}$$

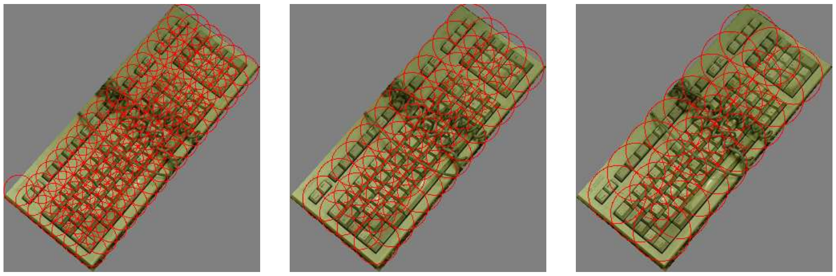

Figure 3: Three coverages at different resolutions (i.e. with different basis) of a view of a keyboard.

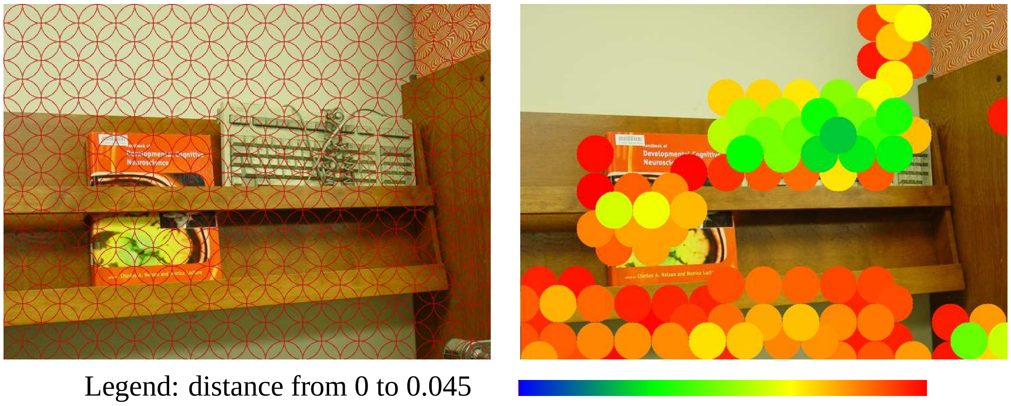

Figure 4: Left: The coverage of a picture depicting a scene with the keyboard of Figure 3 and Right: the confidence regions selected by our algorithm.

where $\delta(c, c_v)$ denotes the L^1 distance between the features vector of c and c_v. As mentioned in Subsection 2.3, δ measures the visual similarity between the covers c and c_v, and it ranges over [0, 1], therefore d too ranges over [0, 1].

The *correspondence map* relating the cover sets $\mathcal{C}(I)$ and $\mathcal{C}(O_j)$, j = 1, . . . , n, is the function $\mathcal{M} : \mathcal{C}(I) \times \Pi_{j=1}^{n} \mathcal{C}(O_j) \longrightarrow [0,1]^n$ such that $\forall$ $c \in \mathcal{C}(I)$

$$\mathcal{M}(c, O_1, \ldots, O_n) = (d(c, O_1), \ldots, d(c, O_n)).$$

The complexity of the computation of $\mathcal{M}$ is $O(qz)$ where z is the number of object covers and q is the cardinality of $\mathcal{C}(I)$.

Distance $d(c, O_j)$ provides a *confidence measure* of the possibility that a cover c of $C(I)$ is occupied by the object O_j. The higher the distance $d(c, O_j)$ is, the lower is the possibility that O_j is located in the image part covered by c.

The τ-*confidence region* of I for the object O_j is the portion of I composed by the image covers $c_1, \ldots, c_h$ such that $d(c_k, O_j) \leq \tau$, for each k = $1, \ldots, h$ and τ is a threshold ranging over [0, 1]. Mathematically:

$$\Omega(O_j) = \cup_{k=1}^{h} \{c_k \in \mathcal{C}(I) : d(c_k, O_j) \leq \tau\}. \tag{7}$$

Figure 4 shows on left a grid superimposed to an image, where an instance of the keyboard of Figure 3 is depicted. On right the τ-confidence regions of the keyboard are highlighted. In this example, τ = 0.045.

4 Automatic Estimate of Cover Basis and Similarity Threshold

The user inputs of our approach are: the database of the objects to be searched, an input (unseen) image, the parameters μ, R, λ_1, . . . , λ_m and the threshold τ.
Here we propose a strategy to set up automatically R and τ. It consists of the following steps:

1. We randomly transform each view of the objects in D by changing its size and its in-plane orientation. In this work, the scale factor and the orientation are chosen in [0.5, 1.5] and [0, 2π] respectively.

2. We compute w bases $R_1, \ldots, R_w$ by considering w different values of N in the formula (2) or (3), depending on the object database.

3. For each radius R_i (i = 1, …, w), we estimate a value τ_i for the similarity threshold as follows:
1. we compute the covers with basis R_i of the reference views and of their transformed versions;
2. we compute the ratio ν_i between the number of transformed views with non empty cover set and the total number of views in the database, i.e.

$$\nu_i = \frac{\#\{T(v) : v \text{ is an object view in } D \text{ and } \exists\, \mathcal{C}(T(v)) \neq \emptyset\}}{\#\{\text{ object views in } D\}} \tag{8}$$

where # indicates the cardinality of the subsequent set and $T(v)$ is the change of scale and in-plane orientation applied to v;
3. for each view v we compute the distances

$$d(c, T(v)) = \min\{\delta(c, c_T) : c \in \mathcal{C}(v), c_T \in \mathcal{C}(T(v))\}$$

and we fix τ_i as the value of the distances $d(c, T(v))$ averaged on the number of views v in D.

The value of ν_i varies in [0,1] and it is used to set up the value of R and τ: in fact, a small value ν_i indicates that many set of covers with radius R_i are empty, and therefore this value R_i must be not considered. By default, the values of R and τ are given by the values of the pair (R_i, τ_i) with the smallest radius and maximum ν_i. By this estimate, the parameters the user must set up, are the number μ, a range of variability of N (i.e. the number w), and the scale factors λ_i, i = 1, …, m.

5 Performance Evaluation

Here we illustrate the tests we carried out on the two public real-world databases PONCE-DB [25] and PM-GADGETS-03 of the Caltech Computational Vision Group. Both of them consist of a database of different kinds of objects represented by a multi-view model and of a set S of test pictures containing the objects.

The performances of our visual attention module are evaluated by measuring how many objects and how much part of them are recovered in the images of S and the percentage of image area where the object search can be focused. Since the considered datasets contain non thin objects, the radius R has been computed by the equation (2). The performance analysis has been repeated for three different values of R obtained by setting up N = 1, 2, 3 and the four values 0.02, 0.03, 0.04 and 0.05 for the threshold τ. In addition, for each dataset we also considered the value of R and τ estimated automatically. The multi-resolution covers have been generated by using the two scale factors λ_1 = 0.7 and λ_2 = 1.3. The radius length is measured in pixels.
More precisely, for each pair (R, τ), for each image I in S and for each instance ω of an object O portrayed in I, we compute the τ-confidence region Ω of O. We evaluate the detection accuracy by the quantities Θ, $\overline{\Theta}$ and Σ defined as follows:

1. $\Theta(\omega) = \frac{A(\omega \cap \Omega)}{A(\omega)}$: the rate of the area of ω covered by Ω:

2. $\overline{\Theta}(\omega) = 1 - \frac{A(\omega \cap \Omega)}{A(\Omega)}$: the rate of the area of Ω not belonging to ω;

3. $\Sigma(\Omega) = \frac{A(\Omega)}{A(I)}$: the rate of the image area selected by Ω, i.e. the rate of image area to be explored for searching O.

In Tables 2, 3 and 5, the values of Θ, $\overline{\Theta}$ and Σ are averaged on the number Q of object instances actually portrayed in the test images. Moreover we compute the *detection rate* defined as the ratio between the number of object instances whose τ-confidence region has $\Theta > \sigma$, and Q. The parameter σ varies in [0, 1], and in the following tests, σ = 0.00, 0.10, 0.50.

Let us now describe how false positives and misclassification cases are measured. For each object $\overline{O}$ not present in I, let $\overline{\Omega}$ be its τ-confidence region. The false positives are quantified by $\Sigma(\overline{\Omega})$, that is the percentage of the area of I covered by $\overline{\Omega}$. The values of $\Sigma(\overline{\Omega})$ reported in Tables 2, 3 and 5 are averaged on the number $\overline{Q}$ of the objects not contained in the test pictures and they are simply indicated by $\overline{\Sigma}$.

To quantify the misclassification cases, for each object $\overline{O}$ not present in I, we compute the percentage of $\overline{\Omega}$ overlapping a portion of I where an object $O \neq \overline{O}$ is depicted, i.e. the portion of $\overline{\Omega}$ that intersects an object instance ω. This measure is indicated by M. Its values, averaged on $\overline{Q}$, are reported in Tables 2, 3 and 5.

For the basis R giving the best performances (R_{best}), we report also the mean rate Γ of image area composed by the union of the τ-confidence regions of each database object (present or not in the image). In these experiments we varied the values of τ.

The experiments presented here have been carried out on a Pentium4 CPU 2.80 GHz. On average, the computation and description of the cover set of a view take about 0.04 seconds, while the mean time for the computation of the correspondence map is about 46 seconds.

5.1 Experiments on PONCE-DB

PONCE-DB database has been built up by the Robotics and Computer Vision Laboratory Beckman Institute (Illinois, USA). It consists of 161 references images depicting 8 objects in different poses against an almost uniform background and of 51 test pictures containing rescaled, rotated, partially occluded and differently illuminated instances of the objects. Objects, test images and ground-truth information are available at http://tev.fbk.eu/DATABASES/objectsPonce.html.

We applied the clustering algorithm reported in [17] to reduce the storage memory space and to speed up the computation of the correspondence maps, so that the total number of relevant views is 24. Table 1 shows the three bases, the correspondent number of object covers, and the parameters ν and τ automatically estimated. The number of covers and the detection rate decrease by increasing the length of the basis. The worst results are obtained for the coarsest object covers (N = 1). In the other cases, no empty covers are generated.

Tables 2 (a), (b), (c) show the detection performances for each pair (R, τ). The best results are obtained for the values of R and τ estimated automatically (R = 24, τ = 0.0391, in the last row of Table 2 (a)). In this case, the mean percentage of object area detected is about 87%, while the percentage of image area to be explored is about 43%. This means that the 57% of the image can be excluded from the object search. The detection rate is very high for each value of σ, and the values of M and $\overline{\Sigma}$ are smaller than 30%. The mean value of Γ is in this case about 74%.

In order to analyze the dependency of the detection performances on the number of object covers, for R = 24, we repeat the experiments by using a coarser grid for the generation of object covers, so that their final number is 3126 instead of 6225. On the contrary, the image grid has not been modified. The results, reported in Table 3, show a noticeable decrement (about the 20%) of the detection rate.

Figure 5 shows the τ-confidence regions of each database object computed by using τ = 0.03, R = 20.

R	N. of Covers	ν	τ
24	6225	1.0000	0.0391
37	2384	1.0000	0.0333
73	487	0.8447	0.0246

Table 1: PONCE-DB: Results of the automatic set up of the cover radius and of the visual similarity threshold.

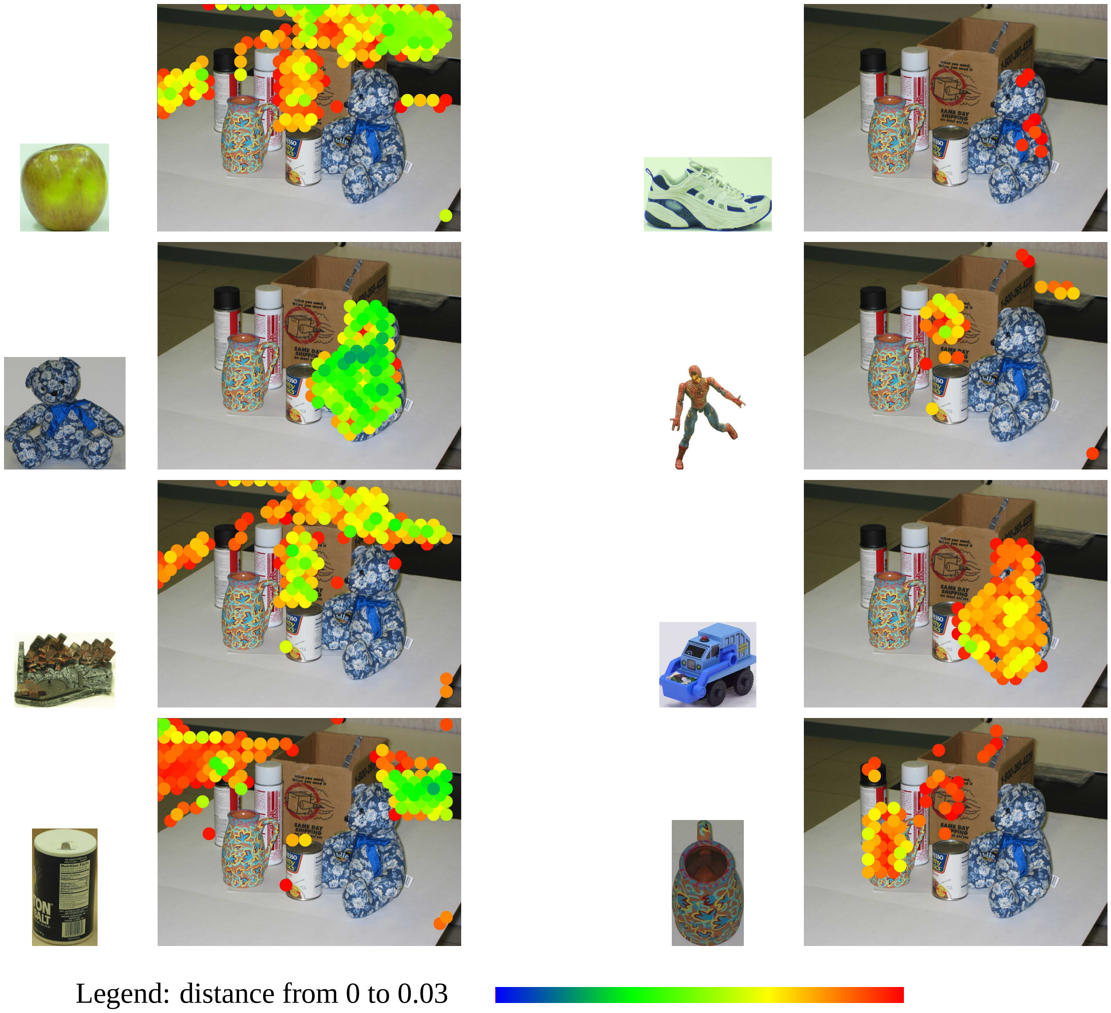

Figure 5: Detection of the τ-confidence regions of the objects in a test image of PONCE-DB. The color of each image cover represents the value of the distance (6) from the object covers. Blue corresponds to distance 0, while red corresponds to the similarity threshold τ = 0.03. Here R = 20.

5.2 Experiments on PM-GADGETS-03

The PM-GADGETS-03 database has been built up by the Computational Vision Group of Caltech, (California, USA). It consists of 48 references images portraying 36 objects under different poses. These images are available along with two different test sets at http://www.vision.caltech.edu/html-files/archive.html.
Here we consider the test set named `TestScenes`, including 45 test pictures with some rescaled, rotated, partially occluded and re-lighted instances of the objects. Since the number of views used for representing each object is very low, no clustering process is necessary. The values of R considered in these experiments are shown in Table 4 with the correspondent number of object covers, and with the parameters ν and τ automatically estimated. As for PONCE-DB, the best results are obtained for the finest object covers (N = 3).
The detection performances obtained by varying R and τ are presented in the Tables 5 (a), (b), (c). The pair (R, τ) estimated automatically allows a very high detection rate (greater than 0.98 for σ = 0.50) and the detection of the 97% of the area of the objects depicted in the test images, but the image area excluded by the object search is only the 21 % about, while the values measuring the false positive detection rate are higher (60% about). Good results are obtained also by using R = 20, and τ = 0.03. In this case the mean percentage of object area detected is about 88%, while the percentage of image area to be explored is about 57 %. The

(a) R = **24** (R_{best})

τ	Θ	$\overline{\Theta}$	Σ	Detection Rate			M	$\overline{\Sigma}$	Γ
				$\sigma = 0.00$	$\sigma = 0.10$	$\sigma = 0.50$			
0.02	0.3611	0.4393	0.0601	0.7976	0.6517	0.3933	0.02596	0.0221	0.2071
0.03	0.6769	0.6719	0.2306	0.9888	0.9213	0.7416	0.0921	0.1246	0.4927
0.04	0.8827	0.7505	0.4393	1.0000	1.0000	0.9326	0.2165	0.3202	0.7607
0.05	0.9576	0.8070	0.5799	1.0000	1.0000	0.9888	0.3875	0.5749	0.8750
0.0391	0.8711	0.7466	0.4245	1.0000	1.0000	0.9326	0.2022	0.2987	0.7453

(b) R = **37**

τ	Θ	$\overline{\Theta}$	Σ	Detection Rate			M	$\overline{\Sigma}$
				$\sigma = 0.00$	$\sigma = 0.10$	$\sigma = 0.50$		
0.02	0.3053	0.3249	0.0388	0.6292	0.5169	0.3483	0.0157	0.01413
0.03	0.6031	0.6021	0.1667	0.9326	0.8539	0.6517	0.0621	0.0920
0.04	0.8584	0.7369	0.4023	1.0000	0.9888	0.9213	0.1660	0.2638
0.05	0.9543	0.7923	0.5534	1.0000	1.0000	1.0000	0.3361	0.5208
0.0333	0.7022	0.6725	0.2471	0.9775	0.9213	0.7865	0.0899	0.1385

(c) R = **73**

τ	Θ	$\overline{\Theta}$	Σ	Detection Rate			M	$\overline{\Sigma}$
				$\sigma = 0.00$	$\sigma = 0.10$	$\sigma = 0.50$		
0.02	0.1679	0.1430	0.0206	0.3146	0.2809	0.2359	0.0046	0.0059
0.03	0.3935	0.4147	0.0966	0.5955	0.5730	0.4382	0.0335	0.0406
0.04	0.7436	0.6866	0.3354	0.9326	0.9101	0.8089	0.1029	0.1563
0.05	0.9137	0.7767	0.515	0.9888	0.9888	0.9889	0.2432	0.3609
0.0246	0.2397	0.2045	0.0348	0.3820	0.3708	0.3034	0.0129	0.01154

Table 2: PONCE-DB: Detection Performances obtained for different values of R and for different values of τ. The last row reports the results obtained by using the parameters estimated automatically.

τ	Θ	$\overline{\Theta}$	Σ	Detection Rate			M	$\overline{\Sigma}$
				$\sigma = 0.00$	$\sigma = 0.10$	$\sigma = 0.50$		
0.02	0.2404	0.5204	0.0700	0.6517	0.5169	0.2247	0.0604	0.1251
0.03	0.5128	0.6361	0.2184	0.8315	0.7865	0.5393	0.2021	0.3604
0.04	0.7264	0.7123	0.4469	0.8427	0.8202	0.7753	0.4035	0.6156
0.05	0.8147	0.7431	0.6299	0.8427	0.8427	0.8427	0.5929	0.7664

Table 3: PONCE-DB: Detection performances obtained by using a coarser grid with respect to that fixed by default (see formula (4)) and R = 24.

detection rate is very high for each value of σ, and the values of M and $\overline{\Sigma}$ are smaller than 26%. The value of Γ is for this dataset very high, because many background parts of the test images are similar to some object covers, especially in the case of the glass bottle.

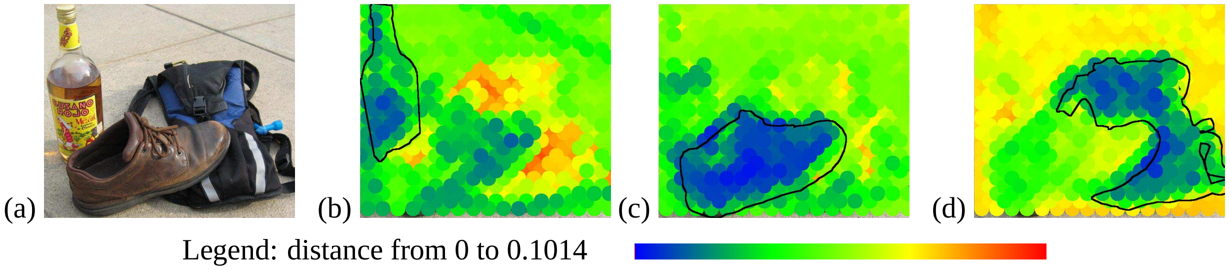

Figure 6: PM-GADGETS-03: Correspondence maps of the objects of the image (a). The object contour has been highlighted in the maps (b), (c), (d). Blue corresponds to the maximum similarity (distance is 0), while red corresponds to 0.1014, that is the maximum distance between the image covers and the object covers.

R	N. of Covers	ν	τ
20	20900	1.0000	0.0464
30	8534	0.9375	0.0423
60	1654	0.8334	0.0321

Table 4: PM-GADGETS-03: Results of the automatic set up of the cover basis and of the visual similarity threshold.

(a) R = **20** (R_{best})

τ	Θ	$\overline{\Theta}$	Σ	Detection Rate			M	$\overline{\Sigma}$	Γ
				$\sigma = 0.00$	$\sigma = 0.10$	$\sigma = 0.50$			
0.02	0.6418	0.5746	0.3280	0.9588	0.9339	0.7190	0.0770	0.065	0.8060
0.03	0.8761	0.6870	0.5714	1.0000	0.9835	0.9339	0.2597	0.2449	0.9600
0.04	0.9499	0.7487	0.7199	1.0000	0.9917	0.9835	0.4546	0.4696	0.9666
0.05	0.9760	0.7845	0.8229	1.0000	1.0000	0.9835	0.6159	0.6619	0.9666
0.0464	0.9700	0.7743	0.7920	1.0000	1.0000	0.9835	0.5604	0.5962	0.9666

(b) R = **30**

τ	Θ	$\overline{\Theta}$	Σ	Detection Rate			M	$\overline{\Sigma}$
				$\sigma = 0.00$	$\sigma = 0.10$	$\sigma = 0.50$		
0.02	0.5736	0.4987	0.2552	0.9008	0.8678	0.6281	0.0425	0.0399
0.03	0.8387	0.6322	0.4831	0.9669	0.9587	0.9174	0.1805	0.1855
0.04	0.9375	0.7333	0.6726	1.0000	0.9917	0.9752	0.3674	0.4111
0.05	0.9658	0.7789	0.7883	1.0000	1.0000	0.9917	0.5361	0.6137
0.0423	0.9483	0.7481	0.7006	1.0000	0.9917	0.9834	0.4079	0.4610

(c) R = **60**

τ	Θ	$\overline{\Theta}$	Σ	Detection Rate			M	$\overline{\Sigma}$
				$\sigma = 0.00$	$\sigma = 0.10$	$\sigma = 0.50$		
0.02	0.4548	0.3216	0.1887	0.7355	0.7273	0.4876	0.0103	0.0131
0.03	0.7366	0.5695	0.3934	0.9008	0.9008	0.8347	0.1015	0.1182
0.04	0.8677	0.6976	0.5977	0.9422	0.9339	0.9174	0.2648	0.3183
0.05	0.9242	0.7539	0.7276	0.9752	0.9669	0.9504	0.4278	0.5227
0.0321	0.78237	0.5989	0.4348	0.9174	0.9091	0.8843	0.1331	0.1539

Table 5: PM-GADGETS-03: Detection Performances obtained for different values of R and for different values of τ. The last row reports the results obtained by using the parameters estimated automatically.

Figure 6 shows the correspondence maps of the objects of a picture of PM-GADGETS-03. The maximum distance between the image and object's covers is 0.1014.

6 Integration in an Object Recognition System

As mentioned in Section 1, our visual attention module can be integrated in different ways in an object recognition system to reduce the complexity of the search and to improve its performances.

Firstly, the correspondence map establishes an exploration order on the regions of the input image: from the portion where it is more probable to find the objects to that where this probability is lower. The map also determines an order on the objects to be searched for: from the most to the less probable. Finally, using the threshold τ for the selection of the τ-confidence regions allows to circumscribe the object search in specific image portions and to restrict their matching to a subset of database objects.

By the priorities on the region exploration and on the object matching, the detection of the objects more likely present in the image is put before the detection of the others. This could help to discard false positives. In systems like [17], where the object search is stopped when a number of object hypotheses in a certain image portion is found, these ordering criteria could be used to improve the performance in terms of time. The hierarchies imposed by the correspondence map could be also integrated in semi-automatic systems for object recognition, where the user stops the search process when the objects are found or filters manually the object hypotheses output by the system, like [10].

In order to measure the benefit of the integration of our attention module in object recognition, we compare the performances of the well known object recognition algorithm SIFT [19] with and without the selection of the τ-confidence regions.

SIFT assumes the multi-view model for the object representation and describes the database views and every input image by key points, characterized by scale-, noise-, occlusion-, and illuminant- invariant features. Typically, the number of key points in an image is very large, so that their extraction and matching are remarkably time consuming. For PONCE-DB, the mean number of the key points is 545 for the object views, and 1782 for the test images. On average, selecting the τ-confidence regions object by object reduces the number of key points of the test images to 831.

SIFT achieve the object recognition by comparing the key points of the test image with those of the given objects and using a nearest-neighbor technique for retaining only the most reliable matches. Although some reliable matches are lost due to the selection of the τ-confidence regions, for 87 of the 89 object instances contained in the test images of PONCE-DB, the use of the τ-confidence regions does not affect the SIFT recognition performances. However, as shown in Figure 7, in the two cases where the matches are discarded, the SIFT key points are very close to the selected image portions.

Finally, we notice that there are 6 cases where no reliable matches are found by SIFT, while our module detects more that the 70% of the area of the object instances contained in those images. This suggests that our approach could be also employed as an additional source in other tools for object detection.

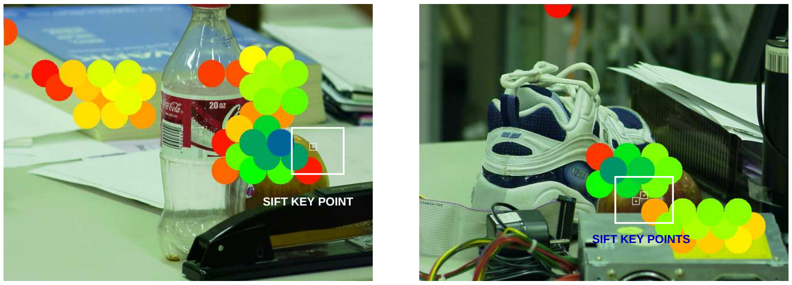

Figure 7: PONCE-DB: for these images the SIFT matches (highlighted by the white rectangles) are not included in the area selected by the τ-confidence regions, but they are very close to them.

7 A comparison

As outlined in Section 1, our confidence region selection as guidance for object search does not simulate the human bottom-up visual attention mechanism. This relies namely on the perception and extraction of conspicuities, like high curvature points or constrasts, which are then processed and tied together by the top-down human mechanism for interpreting the visual scene. In many object recognition approaches, the conspicuities are defined by interest operators [21] or salient feature detectors [13]. Here we compare our SIFT-integrated approach with the method described in [29] where the authors use a saliency-based model of bottom-up attention for showing the usefulness of attention in object recognition. We choose this work among the others because it has been tested on a small but public dataset (102TestImages) available at

https://netfiles.uiuc.edu/walther/www/wapcv04/

Object	Method [29]		Our Method	
	Hits	Missed	Hits	Missed
box	21	2	22	1
book	14	10	22	2

Table 6: 102TestImages: Comparison of our SIFT-integrated approach with the method in [29]. For these experiments, $\tau = 0.045$.

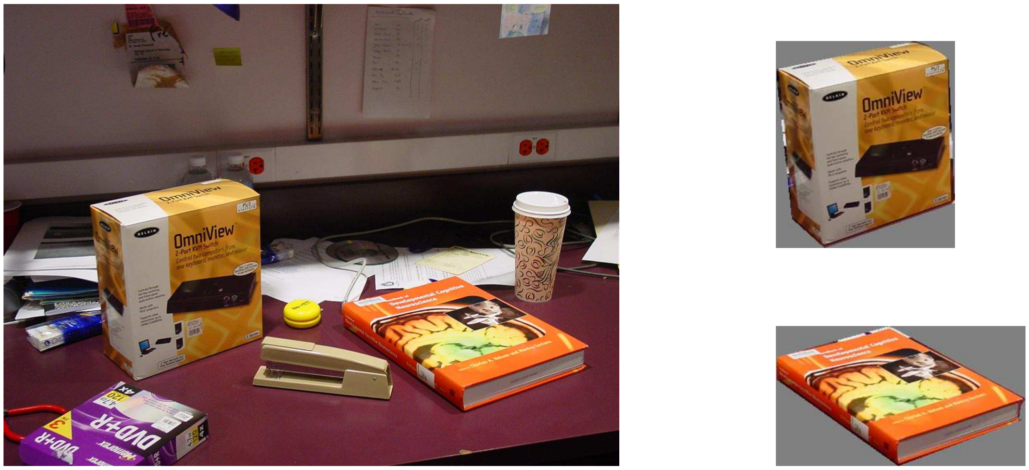

Figure 8: 102TestImages: on left, the image containing the views of the book and the box taken as model for these objects. On right, the two views cropped from the background.

and consisting on 102 real-world pictures with different objects.

In [29], for each object to be recognized and for every input image where the object has to be searched, they compute the saliency map of Itti and Koch [12] and cluster the obtained key points in salient regions. The object recognition is therefore reduced to matching the key points of the salient regions of the object and of the image.

This selective attention algorithm has been applied in [29] for learning and recognizing objects in video sequences and in static images. Recognition is performed by describing and matching the selected key points by SIFT. In [29], the authors use their approach for recognizing the book and the box of 102TestImages shown in the picture reported in Figure 8 within the other 101 images of the database. 23 of these contain the box and 24 the book, and among these, four contain both objects. Table 6 shows that on this dataset our approach performs better: the book is recognized in 24 on 23 image, while the box is recognized in 22 on 23 pictures. The results we achieve are really impressive, especially because in this case the object model is very poor, consisting just of one view for object. Unfortunately, no results with other objects are available for this dataset in [29]. Comparing our approach with others in terms of performances is quite difficult because of the lack of common public datasets. However, we are currently conceiving and designing experiments for further comparisons.

8 Concluding Remarks and Future Work

In this paper we presented a top-down attention module for selecting automatically the portions of an unseen image more likely occupied by a known object. This task is accomplished by describing the object to be searched for and the input image by parts, called covers, and by comparing each object cover with each image cover in terms of visual global features. A correspondence map based on the visual similarity between the covers of the image and of the object defines for each image cover a confidence measure to be occupied by a part of the object. A strategy for estimating automatically some parameters used for the cover computation and for the selection of the confidence regions is also proposed, limiting the user interaction. The experiments we illustrated here show good results: on average, about the 80% of the area of the database objects present in the test images is detected and the percentage of the image area to be explored is reduced to the 50%. The integration of our approach in the SIFT recognition algorithm has resulted in a remarkably reduction of the number of the key points to be matched and consequently of the computational time, without affecting the recognition performances.

The main drawback of our approach is due to the low robustness of the cover descriptors to illuminant changes. Therefore, future work will consist in introducing a color constancy algorithm to make the used features invariant to changes of light and to make automatic also the choice of the scale factors involved in the multi-scale generation of the object covers. We are currently conceiving and designing experiments to compare our approach with others on common databases. Finally we also aim to integrate it in the object recognition system MEMORI [17].

Acknowledgments

The author would like to acknowledge Dr. Carla M. Modena for her inspiring suggestions and fruitful discussions. Moreover, the author would like to thank the anonymous reviewers for their thoughtful work.

References

[1] A. Ahmadyfard and J. Kittler. Colour-based model pruning for efficient ARG object recognition. In *Proc. of ICPR*, 2002.

[2] Brandon Bennett. Combining logic and probability in tracking and scene interpretation. In *Logic and Probability for Scene Interpretation*, Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[3] E. Bermudez-Contreras, H. Buxton, and E. Spier. Attention can improve a simple model for object recognition. *Image Vision Comput.*, 26(6):776–787, 2008.

[4] R. Brunelli and O. Mich. Image retrieval by examples. *IEEE Transactions on Multimedia*, 2(3), 2000.

[5] C. Bundese and T. Habekost. *Principles of visual attention: Linking mind and brain*. Oxford university Press, 2008.

[6] H. Deng, W. Zhang, E. N. Mortensen, T. G. Dietterich, and L. G. Shapiro. Principal curvature-based region detector for object recognition. In *CVPR*, 2007.

[7] C. Elfers, O. Herzog, A. Miene, and T. Wagner. Qualitative abstraction and inherent uncertainty in scene recognition. In *Logic and Probability for Scene Interpretation*, number 08091 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008.

[8] V. Ferrari, T. Tuytelaars, and L. Gool. Simultaneous object recognition and segmentation from single or multiple model views. *Int. J. Comput. Vision*, 67(2), 2006.

[9] S. Frintrop and E. Rome. Simulating visual attention for object recognition. In *Proc. of the Workshop on Early Cognitive Vision*, 2004.

[10] Md. A. Hossain, R. Kurnia, A. Nakamura, and Y. Kuno. Interactive object recognition through hypothesis generation and confirmation. *IEICE - Trans. Inf. Syst.*, E89-D(7):2197–2206, 2006.

[11] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.

[12] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, 1998.

[13] T. Kadir and M. Brady. Saliency, scale and image description. *Int. J. Comput. Vision*, 45(2):83–105, 2001.

[14] N. Kanwisher and E. Wojciulik. Visual attention: Insights from brain imaging. *Nature Reviews Neuroscience*, (1):91–100, 2000.

[15] B. Ko and H. Byun. Extracting Salient Regions And Learning Importance Scores In Region-Based Image Retrieval. *Int. Journal of Pattern Recognition and Artificial Intelligence*, (17(8)):1349–1367, 2003.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *Proc. of BMCV*, 2004.

[17] M. Lecca. Object recognition in color images by the self configuring system MEMORI. *Int. Journal of Signal Processing*, 3(3), 2006.

[18] H. Lei, C. Han, B. Everding, and W. Wee. Graph matching for object recognition and recovery. *Pattern Recognition*, 37, 2004.

[19] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 20:91–110, 2003.

[20] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *Proc. of CVPR*, 2007.

[21] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004.

[22] S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *Proc. of BMVC*, 2002.

[23] B. Platel, E. Balmachnova, L. Florack, and B. M. ter Haar Romeny. Top-points as interest points for image matching. In *ECCV (1)*, volume 3951 of *Lecture Notes in Computer Science*, pages 418–429. Springer, 2006.

[24] A. L. Rothenstein and J. K. Tsotsos. Attention links sensing to recognition. *Image Vision Comput.*, 26(1):114–126, 2008.

[25] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, 66(3), 2006.

[26] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Int. J. Comput. Vision*, 37(2):151–172, 2000.

[27] A. Torralba. Contextual priming for object detection. *Int. J. Comput. Vision*, 53(2):169–191, 2003.

[28] J. K. Tsotsos, L. Itti, and G. Rees. *A Brief and Selective history of Attention, Neurobiology of Attention*. Elsevier/Academic Press, 2005.

[29] D. Walther, U. Rutishauser, C. Koch, and P. Perona. On the usefulness of attention for object recognition. In *Proc. of the 2nd international Workshop on Attention and Performance on Computational Vision*, 2004.

[30] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. Journal of Computer Vision*, 73(2):213–238, 2007.