Dynamic Visual Search Using Inner-Scene Similarity: Algorithms and Inherent Limitations

Tamar Avraham and Michael Lindenbaum

Computer Science Department
Technion, Haifa 32000, Israel
tammya,mic@cs.technion.ac.il

Abstract. A dynamic visual search framework based mainly on inner-scene similarity is proposed. Algorithms as well as measures quantifying the difficulty of search tasks are suggested. Given a number of candidates (e.g. sub-images), our basic hypothesis is that more visually similar candidates are more likely to have the same identity. Both deterministic and stochastic approaches, relying on this hypothesis, are used to quantify this intuition. Under the deterministic approach, we suggest a measure similar to Kolmogorov's ϵ-covering that quantifies the difficulty of a search task and bounds the performance of all search algorithms. We also suggest a simple algorithm that meets this bound. Under the stochastic approach, we model the identities of the candidates as correlated random variables and characterize the task using its second order statistics. We derive a search procedure based on minimum MSE linear estimation. Simple extensions enable the algorithm to use top-down and/or bottom-up information, when available.

1 Introduction

Visual search is required in situations where a person or a machine views a scene with the goal of finding one or more familiar entities. The highly effective visual-search (or more generally, attention) mechanisms in the human visual system were extensively studied from psychophysics and physiology points of view. Yarbus [24] found that the eyes rest much longer on some elements of an image, while other elements may receive little or no attention. Neisser [11] suggested that the visual processing is divided into pre-attentive and attentive stages. The first consists of parallel processes that simultaneously operate on large portions of the visual field, and form the units to which attention may then be directed. The second stage consists of limited-capacity processes that focus on a smaller portion of the visual field. Triesman and Gelade (*feature integration theory* [19]) formulate an hypothesis about how the human visual system performs pre-attentive processing. They characterized (qualitatively) the difference between search tasks requiring scan (serial) and those which do not (parallel, or pop-out). While several aspects of the Feature Integration Theory were criticized, the theory was dominant in visual search research and much work was carried out based on its premises, e.g. to understand how feature integration occurs

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 58–70, 2004.

(some examples are [8,23,21]). Duncan and Humphreys rejected the dichotomy of parallel vs. serial search and proposed an alternative theory based on similarity [3]. According to their theory, two types of similarities are involved in a visual search task: between the objects in the scene, and between the objects and prior knowledge. They suggest that when a scene contains several similar structural units there is no need to treat every unit individually. Thus, if all non-targets are homogeneous, they may be rejected together resulting in a fast (pop-out like) detection, while if they are heterogeneous the search is slower.

Several search mechanisms were implemented, usually in the context of HVS (human visual system) studies (e.g. [8,21,23,5]). Other implementations focused on computer vision applications (e.g. [7,17,18]), and sometimes used other sources of knowledge to direct visual search. For example, one approach is to search first for a different object, easier to detect, which is likely to appear close to the sought for target ([15,22]). Relatively little was done to quantitatively characterize the inherent difficulty of search tasks. Tsotsos [20] considers the complexity of visual search and proves, for example, that spatial boundedness of the target is essential to make the search tractable. In [22], the efficiency of indirect search is analyzed.

This work has two goals: to provide efficient search algorithms and to quantitatively characterize the inherent difficulty of search tasks. We focus on the role of inner-scene similarity. As suggested in [3], the HVS mechanism uses similarity between objects of the same identity to accelerate the search. In this paper we show that computerized visual search can also benefit from such information, while most visual search application totally ignore this source of knowledge. We take both deterministic and stochastic approaches. Under the deterministic approach, we characterize the difficulty of the search task using a metric-space cover (similar to Kolmogorov's ϵ-covering [9]) and derive bounds on the performance of all search algorithms. We also propose a simple algorithm that provably meets these bounds. Under the stochastic approach, we model the identity of the candidates as a set of correlated random variables taking target/non-target values and characterize the task using its second order statistics. We propose a linear estimation based search algorithm which can handle both inner-scene similarity and top-down information, when available.

Paper outline: The context for visual search and some basic intuitive assumptions are described in Sect. 2. Sect. 3 develops bounds on the performance of search algorithms, providing measures for search tasks' difficulty. Sect. 4 describes the VSLE algorithm based on stochastic considerations In Sect. 5 we experimentally demonstrate the validity of the bounds and the algorithms' effectiveness. [1]

2 Framework

2.1 The Context – Candidate Selection and Classification

The task of looking for object/s of certain identity in a visual scene is often divided into two subtasks. One is to *select* sub-images which serve as *candidates*. The

[1] A preliminary version of the VSLE algorithm was presented in [1].

other, the *object recognition* task, is to decide whether a candidate is a sought for object or not. The candidate selection task can be performed by a segmentation process or even by a simple division of the image into small rectangles. The candidates may be of different size, bounded or unbounded [20], and can also overlap. The object recognizer is usually computationally expensive, as the object appearance may vary due to changes in shape, color, pose, illumination etc. The recognizer may need to recognize a category of objects (and not a specific model), which usually makes it even more complex.

The object recognition process gets the candidates, one by one, after some ordering. An efficient ordering, which is more likely to put the real objects first, is the key to high efficiency of the full task. This ordering is the attentional mechanism on which we focus here.

2.2 Sources of Information for Directing the Search

Several information sources enabling more efficient search are possible:
Bottom-up saliency of candidates - In modelling HVS attention, it is often claimed that a saliency measure, quantifying how every candidate is different from the other candidates in the scene, is calculated. ([19,8,7]). Saliency is important in directing attention, but it can sometimes mislead or not be applicable when, say, the scene contains several similar targets.
Top-down approach - When prior knowledge is available, the candidates may be ranked by their degree of consistency with the target description ([23,6]). In many cases, however, it is hard to characterize the objects of interest in a way which is effective and inexpensive to evaluate.
Mutual similarity of candidates - Usually, a higher inner-scene visual similarity implies a higher likelihood for similar (or equal) identity ([3]). Under this assumption, after revealing the identity of one (or a few) candidates, it can effect the likelihood of the remaining candidates to have the same/different identity.

In this paper we focus on (the less studied) mutual similarity between candidates, and assume that no other information is given. Nevertheless, we show how to handle top-down information and saliency, when available.

To quantify similarity, we embed the candidates as points in a metric space with distances reflecting dissimilarities. We shall either assume that the distance between two objects of different identities is larger than a threshold (deterministic approach), or that the identity correlation is a monotonically descending function of this distance (stochastic approach).

2.3 Algorithms Framework

The algorithms we propose share a common framework. They begin from an initial priority map, indicating the prior likelihood of each candidate to be a target. Iteratively, the candidate with the highest priority receives the attention. The relevant sub-image is examined by a high-level recognizer, which we denote the *recognition oracle*. Based on the *oracle*'s response and the previous priority map, a new priority map is calculated, taking into account the similarities.

Usually, systems based on bottom-up or top-down approaches suggest calculating a saliency map before the search starts, pre-specifying the scan order. This *static* map may change only to inhibit the return to already attended locations [8]. The search algorithms proposed here, however, are *dynamic* as they change the priority map based on the results of the object recognizer.

2.4 Measures of Performance

For quantifying the search performance, we take a simplified approach and assume that only the costs associated with calling the recognition oracle are substantial. Therefore, we measure (and predict) the number of queries required to find a target.

3 Deterministic Bounds of Visual Search Performance

In this section we analyze formally the difficulty of search tasks. Readers interested only in the more efficient algorithms based on a stochastic approach can skip this section and continue reading from section 4.1.

Notations. We consider an abstract description of a search task as a pair (X, l), where $X = \{x_1, x_2, \ldots, x_n\}$ is a set of partial descriptions associated with the set of candidates, and $l : X \to \{T, D\}$ is a function assigning identity labels to the candidates. $l(x_i) = T$ if the candidate x_i is a target, and $l(x_i) = D$ if x_i is a non-target (or a distractor). An attention, or search algorithm, A, is provided with the set X, but not with the labels l. It requires $\text{cost}_1(A, X, l)$ calls to the recognizer oracle, until the first target is found. We refer to the set of partial descriptions $X = \{x_1, x_2, \ldots, x_n\}$ as points in a metric space (S, d), $d : S \times S \to \mathbb{R}^+$ being the metric distance function. The partial description can be, for example, a feature vector, and the distance may be the Euclidian metric.

A Difficulty Measure Combining Targets' Isolation and Candidates' Scattering. We would like to develop a search task characteristic which quantifies the search task difficulty. To be effective, this characteristic should combine two main factors:

1. The feature-space-distance between target and non-target candidates.
2. The distribution of the candidates in the feature space.

Intuitively, the search is easier when the targets are more distant from non-targets. However, if the non-targets are also different from each other, the search again becomes difficult. A useful quantification for expressing a distribution of points in a metric space uses the notion of a metric cover [9].

Definition 1. *Let $X \subseteq S$ be a set of points in a metric space (S, d). Let 2^S be the set of all possible subsets of S. $\mathcal{C} \subset 2^S$ is 'a cover' of X if $\forall x \in X \exists C \in \mathcal{C}$ s.t. $x \cap C \neq \emptyset$.*

Definition 2. *$\mathcal{C} \subset 2^S$ is a 'd_0-cover' of a set X if $\mathcal{C}$ is a cover of X and if $\forall C \in \mathcal{C}$ diameter$(C) < d_0$, where diameter(C) is* $\max_{c_1,c_2 \in C} d(c1, c2)$.

Definition 3. *A 'minimum-d_0-cover' is a d_0-cover with a minimal number of elements. We shall denote a minimum-d_0-cover and its size by $\mathcal{C}_{d_0}(X)$ and $c_{d_0}(X)$, respectively.*

If, for example, X is a set of feature vectors in a Euclidian space, $c_{d_0}(X)$ is the minimum number of m-spheres with diameter d_0 required to cover all points in X.

Definition 4. *Given a search task (X, l), let the 'max-min-target-distance', denoted d_T, be the largest distance of a target to its nearest non-target neighbor.*

Theorem 1. *Let $\mathcal{X}_{d_0,c}$ denote all the family of search tasks (X, l) for which d_T,the max-min-target-distance, is bounded from below by some d_0 ($d_T \geq d_0$) and for which the minimum-d_0-cover size is c ($c_{d_0}(X) = c$). The value c quantitatively describes the difficulty of $\mathcal{X}_{d_0,c}$ in the sense that:*

1. *Any search algorithm A needs to query the oracle for at least c candidates in the worst case before finding a target. ($\forall A\ \exists (X, l) \in \mathcal{X}_{d_0,c}$; $cost_1(A, X, l) \geq c$)*
2. *There is an algorithm that, for all tasks in this family, needs no more than c queries for finding the first target.($\exists A\ \forall (X, l) \in \mathcal{X}_{d_0,c}$ $cost_1(A, X, l) \leq c$)*

Proof: 1. We first provide such a 'worst case' X, and then choose the labels l depending on the algorithm A. Choose c points in the metric space, so that all the inner-point distances are at least d_0. Choose the n candidates to be divided equally among these locations. Until a search algorithm finds the first target, it receives only *no* answers from the recognition oracle. Therefore, given a specific algorithm A and the set X, the sequence of attended candidates may be simulated under the assumption that the oracle returns only *no* answers. Choose an assignment of labels l that assigns T only to the group of candidates located in the point whose first appearance in that sequence is last. A will query the oracle at least c times before finding a target.
2. We suggest the following simple algorithm, which suffices for the proof:
FLNN- Farthest Labeled Nearest Neighbor: Given a set of candidates $X = \{x_1, \ldots, x_n\}$, randomly choose the first candidate, query the oracle and label this candidate. Repeat iteratively, until a target is detected: for each unlabeled candidate x_i, compute the distance dL_i to the nearest labelled neighbor. Choose the candidate x_i for which dL_i is maximum. Query the oracle to get its label.
Let us show that FLNN finds the first target after at most c queries for all search tasks (X, l) from the family $\mathcal{X}_{d_0,c}$: Take an arbitrary minimum-d_0-cover of X, $\mathcal{C}_{d_0}(X)$. Let x_i be a target so that $d(x_i, x_j) \geq d_0$ for every distractor x_j (such a x_i exists since $d_T \geq d_0$). Let C be a covering element($C \in \mathcal{C}_{d_0}(X)$) so that $x_i \in C$. Note that all candidates in C are targets. Excluding C, there are $(c-1)$ other covering elements in $\mathcal{C}_{d_0}(X)$ with diameter $< d_0$. Since C contains a candidate whose distance from all distractors $\geq d_0$, FLNN will not query two distractor-candidates in one covering element (whose distance $< d_0$), before it

queries at least one candidate in C. Therefore, a target will be located after at most c queries. (It is possible that a target that is not in C will be found earlier, and then the algorithm stops even earlier.) ∎

Note that no specific metric is considered in the above claim and proof. However, the cover size and the implied search difficulty depend on the partial description (features), which may be chosen depending on the application.

Note also that FLNN does not need to know d_0 and performs optimally (in the worst case) relative to the (unknown) difficulty of the task.

Note that $c_{d_T}(X)$ is the tightest suggested upper-bound on the performance of FLNN for a task (X, l) for which its max-min-target-distance is d_T. Given a search task, naturally, we do not know who the targets are in advance and do not know d_T. Nevertheless, we might know that the task belongs to a family of search tasks for which d_T is greater than some d_0. In this case we can compute $c_{d_0}(X)$, and predict an upper-bound on the queries required for FLNN.

The problem of finding the minimum cover is NP-hard. Gonzalez [4] proposes a 2-approximation algorithm for the problem of clustering a data set minimizing the maximum inner-cluster distance, and proves it is the best approximation possible if $P \neq NP$. In our experiments we used a heuristic algorithm that provided tighter upper bounds on the minimum cover size. Note also that according to the theorem, FLNN's worst cases' results may serve as a lower bound on the minimum cover size as well.

Since computing the cover is hard, we also suggest a more simple measure for search difficulty. Given a bounded metric-space containing the candidates, cover all the space with covering elements with diameter d_0. (For the m-dimensional bounded Euclidean metric space $[0, 1]^m$, there are $\lceil \frac{\sqrt{m}}{d_0} \rceil^m$ such elements.) The number of non-empty such covering elements is an upper-bound on the minimal cover size. See [2] for more results and a more detailed discussion.

4 Dynamic Search Algorithm Based on a Stochastic Model

The FLNN algorithm suffers from several drawbacks. It relates only to the nearest neighbor, which makes it non-robust. A single attended distractor close to an undetected target, reduces the priority of this target and slows the search. Moreover, it does not extend naturally to finding more than one target, and to incorporating bottom-up and top-down information, when available. The alternative algorithm suggested below addresses these problems.

4.1 Statistic Dependencies Modelling

Taking a stochastic approach, we model the object identities as binary random variables with possible values 0 (for non-target) or 1 (for target).

Recall that objects associated with similar identities tend to be more visually similar than objects which are of different identities. To quantify this intuition, we set the covariance between two labels to be a monotonic descending function

γ of the feature-space-distance between them: $cov(l(x_i), l(x_j)) = \gamma(d(x_i, x_j))$, where $X = \{x_1, x_2, \ldots, x_n\}$ is a set of partial descriptions (feature vectors) associated with the set of candidates, $l(x_i)$ is the identity label of the candidate x_i, and d is a metric distance function. In our experiments we use an exponentially descending function ($e^{-d(x_i,x_j)/d_{max}}$, where $d_{\max}$ is the greatest distance between feature-vectors), which seems to be a good approximation to the actual dependency (see Sect. 5.2).

4.2 Dynamic Search Framework

We propose a greedy approach to a dynamic search. At each iteration, estimate the probability of each unlabelled candidate to be a target using all the knowledge available. Choose the candidate for which the estimated probability is the highest and apply the object recognition oracle on the corresponding sub-image.

After the $m-$th iteration, m candidates, $x_1, x_2, \ldots, x_m$, were already handled and m labels, $l(x_1), l(x_2), \ldots, l(x_m)$ are known. We use these labels to estimate the conditional probability of the label $l(x_k)$ of each unlabelled candidate x_k to be 1.

$$p_k \triangleq p(l(x_k) = 1 \mid l(x_1), \ldots, l(x_m)). \tag{1}$$

4.3 Minimum Mean Square Error Linear Estimation

Now, note that the random variable l_k is binary and, therefore, its expected value is equal to its probability to take the value 1. Estimating the expected value, conditioned on the known data, is generally a complex problem and requires knowledge about the labels' joint distribution. We use a linear estimator minimizing the mean square error criterion, which needs only second order statistics.

Given the measured random variables $l(x_1), l(x_2), \ldots, l(x_m)$, we seek a linear estimate $\hat{l_k}$ of the unknown random variable $l(x_k)$, $\hat{l_k} = a_0 + \sum_{i=1}^{m} a_i l(x_i)$, which minimizes the minimum mean square error $e = E((l(x_k) - \hat{l_k})^2)$. Solving a set of (Yule-Walker) equations [13] provides the following estimation:

$$\hat{l_k} = E[l(x_k)] + \boldsymbol{a}^t(\boldsymbol{l} - E[\boldsymbol{l}]), \tag{2}$$

where $\boldsymbol{l} = (l(x_1), l(x_2), \ldots, l(x_m))$ and $\boldsymbol{a} = R^{-1} \cdot \boldsymbol{r}$. R_{ij}, $i, j = 1, \ldots, m$ and r_i, $i = 1, \ldots, m$ are given by $R_{ij} = \mathrm{cov}(l(x_i), l(x_j))$ and $r_i = \mathrm{cov}(l(x_k), l(x_i))$.

$E(l_k)$ is the expected value of the label l_k, which is the prior probability for x_k to be a target. If there is no such knowledge, $E(l_k)$ can be set to be uniform, i.e., $\frac{1}{n}$ (where n is the number of candidates). If there is prior knowledge on the number of targets in the scene, $E(l_k)$ should be set to $\frac{m}{n}$ (where m is the expected number of targets).

The estimated label $\hat{l_k}$ is the conditional mean of a label $l(x_k)$ of an unclassified candidate x_k, and, therefore, may be interpreted as the probability of $l(x_k)$ to be 1

$$p_k = p(l(x_k) = T \mid l(x_1), \ldots, l(x_m)) \sim \hat{l_k}.$$

4.4 The Algorithm: Visual Search Using Linear Estimation – VSLE

- *Given a scene image, choose n sub-images to be candidates.*
- *Extract the set of feature vectors* $X = \{x_1, x_2, \ldots x_n\}$.
- *Calculate pairwise feature space distances and the implied covariances.*
- *Select the first candidate/s randomly (or based on some prior knowledge).*
- *In iteration* $m + 1$*:*
  - *For each candidate* x_k *out of the* $n - m$ *remaining candidates, estimate* $\hat{l_k} \in [0, 1]$ *based on the known labels* $l(x_1), \ldots, l(x_m)$ *using equation 2.*
  - *Query the oracle on the candidate* x_k *for which* $\hat{l_k}$ *is maximum.*
  - *If enough targets were found - abort.*

Our goal is to minimize the expected search time, and the proposed algorithm, being greedy, cannot achieve an optimal solution. It is, however, optimal with respect to all other greedy methods (based on second order statistics), as it uses all the information collected in the search to make the decision.

Note that clustered non-targets accelerate the search and even let the target pop-out when there is only a single non-target cluster. Clustered targets are found immediately after the first target is detected.

As the covariance decreases with distance, estimating the labels only from their nearest (classified) neighbors is a valid approximation which accelerates the search.

4.5 Combining Prior Information

Bottom-up and top-down information may be naturally integrated by specifying the prior probabilities (or the prior means) according to either the saliency or the similarity to known models. Moreover, if the top-down information is available as k model images (one or more), we can simply add them as additional candidates that were examined before the actual search. Continuing the search from this point is naturally faster; see end of Sect.5.2.

5 Experiments

In order to test the ideas described so far, we conducted many experiments using images of different types, using different methods for candidates selection, and different features to partially describe the candidates. Below, we describe a few examples that demonstrate the relation between the algorithms' performance and the tasks' difficulty.

5.1 FLNN and Minimum-Cover-Size

The first set of experiments considers several search tasks and focus on their characterization using the proposed metric cover. Because calculating the minimal cover size is computationally hard, we suggest several ways to bound it

from above and from below and show that combining these methods yields a very good approximation. In this context we also test the FLNN algorithm and demonstrate its guaranteed performance. Finally we provide the intuition explaining why indeed harder search tasks are characterized by larger covers.

The first three search tasks are built around the 100 images corresponding to the 100 objects in the COIL-100 database [12] in a single pose. We think of these images as candidates extracted from some larger image. The extracted features are first, second, and third Gaussian derivatives in five scales [14] resulting in feature vectors of length 45. A Euclidian metric is used as the feature space distance. The tasks differ in the choice of the target which was cups (10 targets), toy cars (10 targets) and toy animals (7 targets) in the three search tasks.

The minimal cover size for every task is bounded as follows: First the minimal target-distractor distance, d_T, is calculated. We developed a greedy heuristic algorithm which prefers sparse regions and provide a possibly non-tight but always valid d_T- cover ; see [2] for details. For the cups search task the cover size was, for example, 24. For all tasks, this algorithm provided smaller (and tighter) covers than those obtained with the 2-approximation algorithm suggested by Gonzalez [4], which for the cups task gave a cover of size 42. Both algorithms provide upper bounds on the size of the minimal cover. See table 1 for cover sizes. Being a rigorous 2-approximation, half of the latter upper bound value (42/2=21 for the cups) is also a rigorous lower bound on the minimal cover size. Another lower bound may be found by running the FLNN algorithm itself, which, by theorem 1, needs no more than $c_{d_T}(X)$ queries to the oracle. By running the algorithm 100 times, starting from a different candidate each run and taking the largest number of queries required (18 for the cups task), we get the tightest lower bound; see table 1 where the average number of queries required by the FLNN is given as well.

Note that the search for cars was the hardest. While the car targets are very similar to each other (which should ease the search), finding the first car is hard due to the presence of distractors which are very similar to the cars (d_T is small). The cups are also similar to each other, but are dissimilar to the distractors, implying an easier search. On the other hand, the different *toy animals* are dissimilar, but as one of them is very dissimilar from all candidates, the task is easier as well. Note that the minimal cover size captures the variety of reasons characterizing search difficulty in a single scalar measure.

We also experimented with images from the Berkeley hand segmented database [10] and used the segments as candidates; see Fig.1. Small segments are ignored, leaving us with 24 candidates in the *elephants* image and 30 candidates in the *parasols* image. The targets are the segments containing elephants and parasols, respectively. For those colored images we use color histograms as feature vectors. In each segment (candidate), we extract the values of $\frac{b}{r+g+b}$ and $\frac{r}{r+g+b}$ from each pixel, where r, g, and b are values from the RGB representation. Each of these two dimensions is divided into 8 bins, resulting a feature vector of length 64. Again, we use Euclidean metric for distance measure. (Using other histogram comparison methods, such as the ones suggested in [16] the results

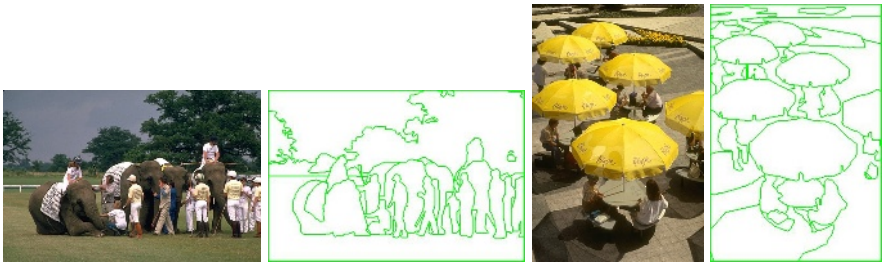


Fig. 1. The *elephants* and *parasols* images taken from the Berkeley hand segmented database and the segmentations we used in our experiments. (colored images)

Table 1. Experiment results for FLNN and cover size. The real value of minimal cover size is bounded from below by 'FLNN worst' and the half of '2-Approx. cover size', and bounded from above by 'Heuristic cover size' and '2-Approx. cover size'. The rightmost column shows that VSLE improves the results of FLNN for finding the first target.

Search task	# of cand.	# of targets	FLNN worst	FLNN mean	Heuristic cover size	2-Approx. cover size	Real cover size	VSLE worst
cups	100	10	18	8.97	24	42	21-24	15
cars	100	10	73	33.02	79	88	73-79	39
toy animals	100	7	22	9.06	25	42	22-25	13
elephants	24	4	9	5.67	9	11	9	8
parasols	30	6	6	3.17	8	13	7-8	4

were similar.) See the results in Table 1. Although the mean results are usually not better than the mean results of a random search, the worst results are much better.

5.2 VSLE and Covariance Characteristics

The VSLE algorithm described in Sect.4 was implemented and applied to the same five visual search tasks described in Sect.5.1. See Fig.2 for part of the results. Unlike FLNN which deals only with finding the first target, VSLE continues and aims also to find the other targets. Moreover, in almost all the experiments we performed, VSLE was faster in finding the first target (both in the worst and the mean results). See the rightmost column in table 1.

VSLE relies on the covariance between candidates' labels. We use a covariance function that depends only on feature-space-distance, and argue that for many search tasks this function is monotonic descending in this distance. To check this assumption we estimate the covariance of labels vs. feature-space-distance of search tasks and confirmed for its validity; see Fig.2 and [2].

We experimented with a preliminary version of integrated segmentation and search. An input image (see Fig.3) was segmented using k means clustering in the RGB color space (using 6 clusters). All (146) connected components larger than 100 pixels served as candidates. The VSLE algorithm searched for the (7)

faces in the image, using a feature vector of length 4: each segment is represented by the mean values of red green and blue and the segment size.

No prior information on size, shape color or location was used. Note that this search task is hard due to the presence of similarly colored objects in the background, and due to the presence of hands which share the same color but are not classified as targets. Note that in most runs six of the seven faces are detected after about one-sixth of the segments are examined. We deliberately chose a very crude segmentation, demonstrating that very good segmentation is not required for the proposed search mechanism.

Using the method suggested in Sect.4.5, we incorporate top-down information and demonstrate it on the *toy cars* case: 3 toy cars which do not belong to the COIL-100 database are used as model targets. The search time was significantly reduced as expected; see Fig.4.

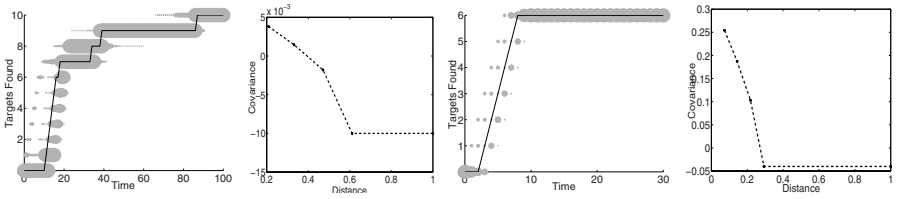
6 Discussion

In this paper we considered the usage of inner-scene similarity for visual search, and provided both measures for the difficulty of the search, and algorithms for implementing it. We took a quantitative approach, allowing us not only to optimize the search but also to quantitatively predict its performance.

Interestingly, while we did not aim at modelling the HVS attention system, it turns out that it shares many of its properties, and in particular, is similar to Duncan and Humphreys's model [3]. As such, our work can be considered as a quantification of their observations. Not surprisingly, our results also show that there is a continuity between the two poles of 'pop-out' and 'sequential' searches.

While many search tasks rely only on top down or bottom up knowledge, inner scene similarities always help and may become the dominant source of knowledge when less is known about the target. Consider, for example the *parasols* search task (Sect. 5). First, note that the targets take a significant image fraction, and cannot be salient. Then, the parasols are similar and different from the non-targets in their color, but if this color is unknown, they cannot be searched using top-down information. More generally, considering a scene containing several objects of the same category, we argue that their sub-images are more similar than images of such objects taken in different times and places. This happens because the imaging conditions are more uniform and because the variability of objects is smaller in one place. (e.g, two randomly chosen trees are more likely to be of the same type if they are taken from the same area.)

We are now working on building an overall automatic system that will combine the suggested algorithms (extended to use bottom-up and top-down information) with grouping and object recognition methods. We also intend to continue analyzing search performance. We would like to be able to predict search time for the VSLE algorithm, for instance, in a manner similar to that we have achieved for FLNN. While the measure of minimal cover size as a lower bound for the worst cases holds, we aim to suggest a tighter bound for cases that are statistically more common.

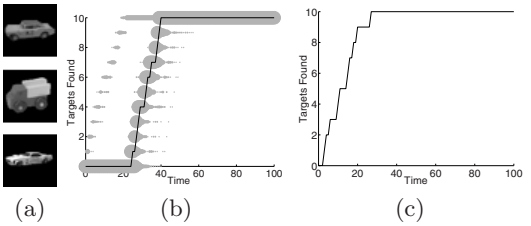


Cups VSLE (a) *Cups* covariance (b) *Parasols* VSLE (c) *Parasols* covariance (d)

Fig. 2. VSLE and covariance vs. distance results. (a) VSLE results for the *cups* search task.The solid lines describe one typical run. Other runs, starting each time from a different candidate, are described by the size of the gray spots as a distribution in the (time, number of targets found) space. It is easy to find the first cup since most cups are different from non-targets. Most cups resemble and follow pretty fast, but there are three cups (two without a handle and one with a special pattern) that are different from the rest of the cups, and are found rather late. (b) Estimate of labels covariance vs. feature-space-distance for the *cups* search task. (c) VSLE results for the *parasols* search task. All the parasols are detected very fast, since their color is similar and differs from that of all other candidates. (d) Estimate of labels covariance vs. feature-space-distance for the *parasols* search task.

(a) (b) (c)

Fig. 3. VSLE applied on an automatic-color-segmented image to detect faces. (a) The input image (colored image) (b) Results of an automatic crude color-based segmentation (c) VSLE results (see caption of figure 2 for what is shown in this graph).



(a) (b) (c)

Fig. 4. VSLE using top-down information for the *toy cars* search task. (a) The three model images. (b) VSLE results without using the models, (c)results of extended VSLE using the model images.

References

1. T. Avraham and M. Lindenbaum. A Probabilistic Estimation Approach for Dynamic Visual Search. *Proceedings of International Workshop on Attention and Performance in Computer Vision (WAPCV)*, 1–8, 2003.
2. T. Avraham and M. Lindenbaum. CIS Report #CIS-2003-02, 2003. Technion - Israel Institute of Technology, Haifa 32000, Israel.
3. J. Duncan and G.W. Humphreys. Visual search and stimulus similarity. *Psychological Review*, 96:433–458, 1989.
4. T.F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(2-3):293–306, June 1985.
5. G.W. Humphreys and H.J. Muller. Search via recursive rejection (serr): A connectionist model of visual search. *Cognitive Psychology*, 25:43–110, 1993.
6. L. Itti. Models of bottom-up and top-down visual attention. *Thesis*, January 2000.
7. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259, November 1998.
8. C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural vircuity. *Human Neurobiology*, 4:219–227, 1985.
9. A.N. Kolmogorov and V.M. Tikhomirov. *epsilon*-entropy and *epsilon*-capacity of sets in functional spaces. *AMS Translations. Series 2*, 17:277–364, 1961.
10. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th ICCV*, volume 2, pages 416–423, July 2001.
11. U. Neisser. *Cognitive Psychology*. Appleton-Century-Crofts, New York, 1967.
12. S. Nene, S. Nayar, and H. Murase. Columbia object image library (coil-100). *Technical Report CUCS-006-96, Department of Computer Science, Columbia University*, February 1996.
13. A. Papoulis and S.U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, NY, USA, fourth edition, 2002.
14. R.P.N. Rao and D.H. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78(1–2):461–505, 1995.
15. R.D. Rimey and C.M. Brown. Control of selective perception using bayes nets and decision theory. *International Journal of Computer Vision*, 12:173–207, 1994.
16. M.J. Swain and D.H. Ballard. Color indexing. *IJCV*, 7:11–32, 1991.
17. H. Tagare, K. Toyama, and J.G. Wang. A maximum-likelihood strategy for directing attention during visual search. *IEEE PAMI*, 23(5):490–500, 2001.
18. A. Torralba and P. Sinha. Statistical context priming for object detection. In *Proceedings of the 8th ICCV*, pages 763–770, 2001.
19. A. Treisman and G.Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.
20. J.K. Tsotsos. On the relative complexity of active versus passive visual search. *IJCV*, 7(2):127–141, 1992.
21. J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y. Lai, N. Davis, and F.J. Nuflo. Modeling visual attention via selective tuning. *Artificial intelligence*, 78(1-2):507–545, 1995.
22. L.E. Wixson and D.H. Ballard. Using intermediate objects to improve the efficiency of visual-search. *IJCV*, 12(2-3):209–230, April 1994.
23. J.M. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1(2):202–238, 1994.
24. A.L. Yarbus. *Eye Movements and Vision*. Plenum Press, New York, 1967.