Object-based regions of interest for image compression

Sunhyoung Han
Electrical and Computer Engineering
University of California, San Diego
s1han@ucsd.edu

Nuno Vasconcelos
Electrical and Computer Engineering
University of California, San Diego
nuno@ucsd.edu

Abstract

A fully automated architecture for object-based region of interest (ROI) detection is proposed. ROI's are defined as regions containing user defined objects of interest, and an efficient algorithm is developed for the detection of such regions. The algorithm is based on the principle of discriminant saliency, which defines as salient the image regions of strongest response to a set of features that optimally discriminate the object class of interest from all the others. It consists of two stages, *saliency detection* and *saliency validation*. The first detects salient points, the second verifies the consistency of their geometric configuration with that of training examples. Both the saliency detector and the configuration model can be learned from cluttered images downloaded from the web. Learning and ROI detection are optimal in the minimum probability of error (MPE) sense, and computationally efficient. This enables interactive user training of ROI-based image coders, with minimal amounts of manual supervision. Experimental results are presented for images of complex scenes, containing both objects and background clutter, and demonstrate good object-based ROI image compression performance.

1 Introduction

Many applications in image processing require the variable assignment of resources to different image regions. For example images can be displayed with spatially variable resolution to enable faster browsing or manipulation, image regions can be subject to variable degrees of error correction to achieve an optimum trade-off between transmission reliability and efficiency, or images can be encoded with spatially varying bit rates so as to guarantee higher fidelity in the regions that are deemed more importance to their viewers [1, 2]. The automatic identification of these *regions of interest* (ROI's) is a non-trivial problem, since they are determined by the perceptual mechanisms of visual attention. While these mechanisms are still poorly understood, it is known that attention is driven by two complementary mechanisms. The first, usually referred as *bottom-up*, is very fast and completely stimulus driven, i.e. active even when a subject is not actively pursuing a task. For example a, vividly colored, “DANGER” sign posted on a white wall attracts attention, even when subjects are not looking for signs of danger. The second, usually referred to as *top-down*, is slower and task dependent. When asked to identify a face, subjects spend most of the time analyzing regions that contain features useful for facial identification, such as eyes, nose, or mouth [13].

Various models of attention have been proposed in the psychology literature [14, 15, 16, 5], and offer plausible explanations for human visual search strategies. However, most of these models do not suggest a computational architecture that can be directly applied to image processing. On the other hand, the computational models which have been proposed in the literature [3, 4], and applied to problems such as compression and image understanding, belong to the bottom-up class. This leads to ROI definitions that cannot be made task specific, i.e. tuned to the specific application or the specific subject at the end of the image processing chain. Designing ROI detectors tunnable to either an application or a subject is, however, not simple. Since it is a-priori unknown what subjects may be interested in, the detector should be generic enough to handle large numbers of object categories. While fast and highly accurate object detectors are currently available for some categories, such as faces and cars [6], their application to other classes is problematic. The main bottleneck is the complexity of training, which requires careful manual assembly of very large training sets, composed of precisely cropped and aligned example images. The resulting cost, in terms of both time and manual labour, makes the user-guided training of these detectors infeasible.

In this work, we pursue an alternative strategy for the design of ROI detectors. This strategy is of a top-down nature, but trades the emphasis on highly accuracte classification (characteristic of classical detector design) for an emphasis on 1) weak supervision and 2) learning efficiency. It consists of two stages: *saliency detection* and *saliency validation*. In the first, given an object class of interest, we search a predefined dictionary of features (or filters) for the subset that best discriminates between 1) the examples from that class and 2) a set of generic images, representative of the distribution of all natural imagery. This can be done very efficiently, with recourse to an information theoretic feature selection criterion based on the maximization of mutual information [7]. Filtering an image with this discriminant filter bank produces a *discriminant saliency map*, which has high magnitude at the locations of the class of interest and low magnitude elsewhere. The price paid for the gain in learning efficiency is some loss of classification accuracy. This is addressed by the introduction of a *saliency validation* stage, inpired by recent developments in computer vision, which have shown the benefits of representing objects as constellations of "parts" [17]. The basic idea is to equate locations of maximal saliency with object "parts", and build a model for the spatial configurations of these parts in the class of interest. The salient configurations of the image under analysis are then rated according to how well they are explained by this model. This imposes a constraint of geometric consistency between ROI's and the training examples, and is a powerful filter for the rejection of false-positives. We develop a computationally efficient validation step by approximating all saliency maps by Gaussian mixtures and using fast hierarchical inference procedures for model-building. While our geometric constraints are quite simple (much simpler than those commonly used in vision [17, 18]), their combination with discriminant saliency is rather effective.

Overall, because we search for both the features and configurations that are more common in the object class of interest than in the generic class of natural images, the proposed strategy is fairly robust to the presence of clutter in the training images. This makes it possible to learn without the requirement for manual segmention or alignment of examples during the assembly of the training set. The cost of tailoring the architecture for a new class is therefore quite low, making it possible for users to define new ROI detectors. It is shown that good results, in terms of both ROI accuracy and compression efficiency, can be achieved with training sets automatically downloaded from the web, without any manual processing. Furthermore, the architecture is completly generic, and applicable to any object category.

2 Top-down saliency

The first step in the proposed ROI detection algorithm is the detection of top-down salient regions. This is accomplished with a discriminate formulation of saliency introduced in [8], which we now briefly review.

2.1 Discriminant Saliency and feature selection

Discriminant saliency defines salient regions in decision theoretic terms. It assumes a binary classification problem, which opposes the visual class of interest to a null hypothesis composed of the set of all natural images. Salient regions are those containing visual features which can be assigned to the class of interest with minimum probability of error (MPE). Their computation consists of two steps: feature selection and salient point detection.

The feature selection stage identifies the visual features that best discriminante between the class of interest and the null hypothesis. Defining a binary random variable Z such that $Z = 0$ for the null hypothesis and $Z = 1$ for the class of interest, and assuming that the feature vectors are drawn from a random process $\mathbf{X} = (X_1, \ldots, X_n)$, the saliency of each feature is measured by the mutual information between the feature and the class label

$$I(X_k; Z) =< KL[P_{X_k|Z}(x|i)||P_{X_k}(x)] >_Z, \tag{1}$$

where $KL[p||q] = \int p(x) \log \frac{p(x)}{q(x)} dx$ is the Kullback-Leibler divergence between the distributions $p(x)$ and $q(x)$ and $< f(i) >_Z = \sum_i P_Z(i) f(i)$. The salient features for the class of interest are those that maximize this mutual information. In our experience, the precise choice of the feature dictionary does not have a major impact on saliency judgments. We have tested various frequency decompositions including Gabor and Haar wavelets, and the discrete cosine transform (DCT), with similar results[1]. More important is to collect features at various image scales, since this enables the automatic determination of both the *location* and *scale* of salient image points. This is implemented by preliminary decomposition of the image into a Gaussian pyramid, and application of the feature transformation to each of the resulting pyramid layers.

2.2 Salient point detection

The second step of discriminant saliency is salient point detection. Given a class of interest, this is implemented with the MPE rule for the classification problem that opposes that class to the null hypothesis. This rule consists of a likelihood ratio test between the two hypothesis, where the different features are weighted according to their saliency, and can be shown to produce a saliency measure of the form

$$S(\mathbf{l}) = \sum_k I(X_k; Z) R_k(\mathbf{l}), \quad R_k(\mathbf{l}) = \{\max[-Im * F_k(\mathbf{l}), 0]\}^2 + \{\max[Im * F_k(\mathbf{l}), 0]\}^2 \tag{2}$$

where $R_k(\mathbf{l})$ is the result of half-wave rectification of the output of filter F_k, associated with feature X_k, at location $\mathbf{l}$ [8]. We refer to $S(\mathbf{l})$ as the *saliency map* with respect to the class of interest. Salient points are defined as the local maxima of the saliency map, and identified by feeding the latter to a peak detection module [11]. The saliency scale of a given location is the scale (radius of the region of support) of the most salient feature at that location. The saliency map is compactly described by a table of salient points of location $\mathbf{l}_k$, scale s_k, and amplitude $S(\mathbf{l}_k)$, ordered by decreasing amplitude. Finally, this amplitude(θ_k) is normalized and the detector outputs the list of salient point parameters($\mathbf{z}_k$)

$$\theta_k = \frac{S(l_k)}{\sum_{j=1}^{N} S(l_j)}, \quad \mathbf{z}_k = (\theta_k, l_k, s_k)^T, k = 1, \ldots, N. \tag{3}$$

N is chosen so that $S(l_k) < 0.1 \max_i S(l_i), \forall k = 1, \ldots, N$, eliminating locations of trivially small response.

3 Salient point validation

The saliency detection procedure described above favors computational efficiency over detection accuracy. When applied to complex scenes, where the target visual concept is presented against a background containing substantial ammounts of clutter, it can have a relatively large false positive rate. This problem is addressed with an, equally efficient, validation procedure that rejects configurations of salient points which are geometrically inconsistent with the training examples from the target class.

3.1 Representation

Saliency validation is based on a probabilistic representation of the saliency map. To account for false-positives (salient locations which do not depict the target visual concept) the set of salient point parameters $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ is divided into two mutually exclusive subsets $\mathcal{Z} = \mathcal{Z}_+ \cup \mathcal{Z}_-$, where $\mathcal{Z}_+$ contains the points whose region of support overlaps with image region covered by the target concept and $\mathcal{Z}_-$ the remaining. A binary variable Y denotes whether the image location under consideration depicts the target or not, i.e. $Y = 1$ if the location belongs to the region of support of the target and $Y = 0$ otherwise, leading to the following generative model for the saliency map. First, a label is drawn from Y, determining wether the salient point is a true or a false positive. If $Y = 1$ (true positive) the i^{th} salient parameter vector in $\mathcal{Z}_+$ is selected with probability π_i^1. The salient location is finally sampled from a Gaussian distribution whose mean and variance are the saliency parameters $\mathbf{l}_i^1$ and $(s_i^2)^1\mathbf{I}$. If $Y = 0$ (false positive) the salient parameter vector is selected from $\mathcal{Z}_-$, and the Gaussian distribution has parameters $\mathbf{l}_i^0$ and $(s_I^2)^0\mathbf{I}$. The overall model is a Gaussian mixture,

$$P_{\mathbf{X}}(\mathbf{x}) = \sum_{i=1}^{N^1} \pi_i^1 \mathcal{G}(\mathbf{x}, \mathbf{l}_i^1, (s_i^2)^1\mathbf{I}) + \sum_{i=1}^{N^0} \pi_i^0 \mathcal{G}(\mathbf{x}, \mathbf{l}_i^0, (s_i^2)^0\mathbf{I}) \tag{4}$$

[1] All results reported in this work were obtained with the DCT

where

$$\mathcal{G}(\mathbf{x}, \mu, \mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}, \quad \pi_i^1 = \frac{\theta_i^1 (s_i^2)^1}{\sum_{i=1}^{N^1} \theta_i^1 (s_i^2)^1 + \sum_{i=1}^{N^0} \theta_i^0 (s_i^2)^0}. \tag{5}$$

and π_i^0 is defined similarly. Note that the weight of each Gaussian is a function of both saliency amplitude and scale.

3.2 Generative model for saliency configuration

The elimination of false-positive salient points assumes that the visual concepts of interest have a consistent geometric configuration. A generative model is assumed for this configuration, and its parameters are learned from training data. Given the results of discriminant saliency on an unseen test image, the likelihood of the configuration of the detected points under the learned model is measured, enabling the rejection of geometrically inconsistent configurations. While various graphical models have been proposed in computer vision for modeling the relationship between object parts [12], their learning complexity (hours if not days for relatively small training sets) is unsuitable for user-driven training. The desire for a flexible ROI detection algorithm, advises the use of simpler configuration models. In this work, we adopt a simple "blob-based" model, which reduces the overall saliency distribution to a Gaussian. To maximize discrimination, the model accounts for both the true and false-positive saliency classes

$$P_{\mathbf{X}}(\mathbf{x}) = \alpha_1 \mathcal{G}(\mathbf{x}, \mu_1, \mathbf{\Sigma}_1) + \alpha_0 \mathcal{G}(\mathbf{x}, \mu_0, \mathbf{\Sigma}_0), \tag{6}$$

where $\alpha_1 + \alpha_0 = 1$, μ_1 and μ_0 are the centers of mass of the true and false-positive components of the saliency map, and $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_0$ their covariances.

3.3 Hierarchical EM algorithm for parameter learning

Learning of model parameters is accomplished by considering the salient points extracted from each image separately. If there are K images, the training set is a collection of K salient point sets, where the k^{th} set contains N_k salient points $\theta_{ik} = (\pi_{ik}, \mathbf{l}_{ik}, s_{ik}\mathbf{I}), i = 1, \ldots, N_k$ derived from the k^{th} image. It is assumed that these points are subject to a translation of μ_k with respect to the origin of a coordinate frame common to all images. The individual image saliency mixtures are then combined into an overall image level saliency mixture

$$P_{\mathbf{X}}(\mathbf{x}) \quad = \quad \sum_{k=1}^{K} \sum_{i=1}^{N_k} \pi_{ik} \mathcal{G}(\mathbf{x}, \mathbf{l}_{ik} - \mu_k, s_{ik}^2 I). \tag{7}$$

The configuration model is assumed to be centered at the origin of the canonical coordinate frame, i.e.

$$P_{\mathbf{X}}(\mathbf{x}) = \alpha_1 \mathcal{G}(\mathbf{x}, \mathbf{0}, \mathbf{\Sigma}_1) + \alpha_0 \mathcal{G}(\mathbf{x}, \mathbf{0}, \mathbf{\Sigma}_0), \quad \text{with } \alpha_1 + \alpha_0 = 1. \tag{8}$$

Given a set of image saliency parameters $\theta_{ik}, k = 1, \ldots K, i = 1, \ldots, N_k$, the parameters of the configuration model $\{\alpha_j, \Sigma_j\}, j \in \{0, 1\}$, and the displacements $\mu_k, k = 1, \ldots, K$ are learned with a hierarchical EM algorithm, that iterates between the following steps

E-step: for $k = 1, \ldots K, i = 1, \ldots, N_k$ and $j \in \{0, 1\}$, compute

$$h_{ik}^j = \frac{[\mathcal{G}(\mathbf{l}_{ik} - \mu_k, \mathbf{0}, \mathbf{\Sigma}_j) e^{-\frac{1}{2}\text{trace}\{(\mathbf{\Sigma}_j)^{-1}(s_{ik}^2 I)\}}]^{M_{ik}} \alpha_j}{\sum_{l \in \{0,1\}} [G(\mathbf{l}_{ik} - \mu_k, 0, \mathbf{\Sigma}_l) e^{-\frac{1}{2}\text{trace}\{(\mathbf{\Sigma}_l)^{-1}(s_{ik}^2 I)\}}]^{M_{ik}} \alpha_l} \tag{9}$$

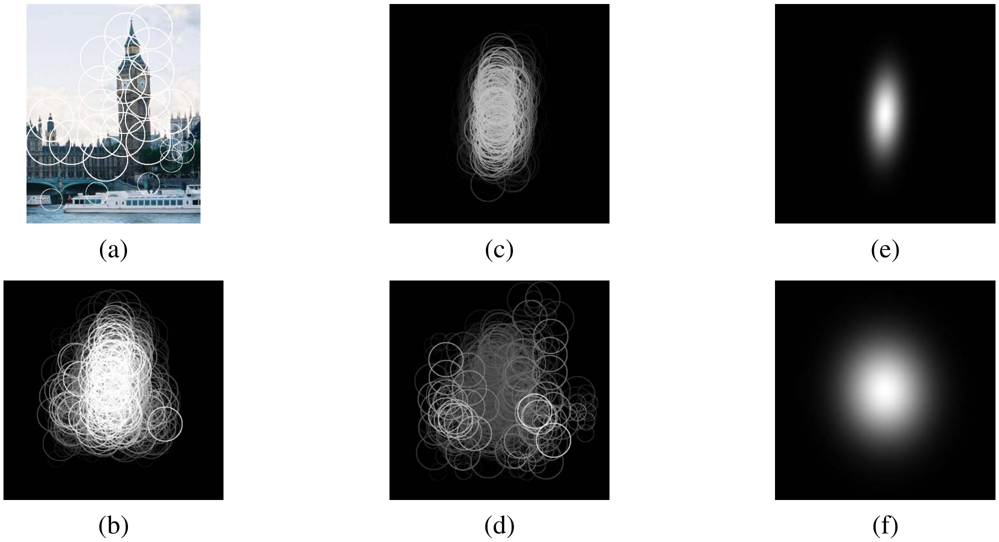
M-step: for $k = 1, \ldots K, i = 1, \ldots, N_k$ and $j \in \{0, 1\}$, set

$$(\alpha_j)^{new} \quad = \quad \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i=1}^{N_k} h_{ik}^j}{N_k} \tag{10}$$

$$(\mathbf{\Sigma}_j)^{new} \quad = \quad \frac{1}{K} \sum_{k=1}^{K} \frac{1}{\sum_{i=1}^{N_k} h_{ij}^j \pi_{ik}} \times [\sum_{i=1}^{N_k} h_{ij}^j \pi_{ik} (\mathbf{l}_{ik} - \mu_k)(\mathbf{l}_{ik} - \mu_k)^T + s_{ik}^2 I]$$

$$\mu_k^{new} \quad = \quad (\sum_{j=0}^{1} \sum_{i=1}^{N_k} \Sigma_j^{-1} h_{ik}^j \pi_{ik})^{-1} \sum_{j=0}^{1} \sum_{i=1}^{N_k} \Sigma_j^{-1} h_{ik}^j \mathbf{l}_{ik} \pi_{ik} \tag{11}$$

$$\tag{12}$$



(a) (c) (e)

(b) (d) (f)

Figure 1: (a) Saliency detection result for a single image, (b) salient points detected on 40 images, (c) points classified as true-positives, (d) points classified as false-postives, (e) true-positive component of the configuration model, and (f) false-positive component.

This algorithm is an extension of the hierarchical EM algorithm introduced in [9], and its derivation is ommitted. As usual for EM, the E-step computes the posterior probability of the salient points under the two classes, and the M-step computes sample statistics weighted by these posterior assignments. The hierarchical EM equations are also quite similar to those of the standard EM algorithm for learning mixture parameters, but account for the additional information contained in the image level covariances and weights. If the covariances are equal to the identity, and the weights π_{ik} follow a uniform distribution, the algorithm reduces to standard EM, applied to the salient locations $\mathbf{l}_i$. A final difference is that, in the parameter update step, salient points from the different images are aligned by application of the displacements μ_k. The new parameters are then computed for each image and averaged over the set of training images. The displacement vectors μ_k can also be seen as the centers of mass of the saliency map extracted from each image, before the image alignment.

3.3.1 Robust estimation

Ideally, the displacements μ_k should be computed only from true-positive salient points, since these by definition cover the target. False-positives are outliers due to background objects and have much greater variability. As illustrated by examples (b) and (d) of Figure 2, they can appear substantially far from the object. As is usual in statistics, the presence of outliers advises the adoption of a robust estimator. In the context of EM, robustness can usually be achieved by exploiting the fact that the posterior probabilities of each class are available for each point. In the specific case of the saliency problem, we limit the set of points that contribute to the estimation of the displacements to those that can be classified as true-positives with a MPE rule, i.e. those which have greater than 0.5 probability of belonging to the true-positive class. This is implemented by modifiying the M-step according to

$$\mu_k^{new} = \frac{\sum_{i=1}^{N_k} \delta[h_{ik}^1](\mathbf{l}_{ik} - \mu_k)\pi_{ik}}{\sum_{i=1}^{N_k} \delta[h_{ik}^1]\pi_{ik}} \quad \text{where} \quad \delta[h_{ik}^j] = \begin{cases} h_{ik}^j & h_{ik}^j \geq 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

3.3.2 Examples

Figure 1 illustrates the learning process, using an example image from the "Big Ben" class. Figure 1 (a) presents the output of the saliency detector, representing each salient point by a circle centered at the salient point location and with radius equal to the salient point scale. Note that while the regions of support of various salient points overlap with the target object, there are a number of false-positives. Figure 1 (b) presents all the

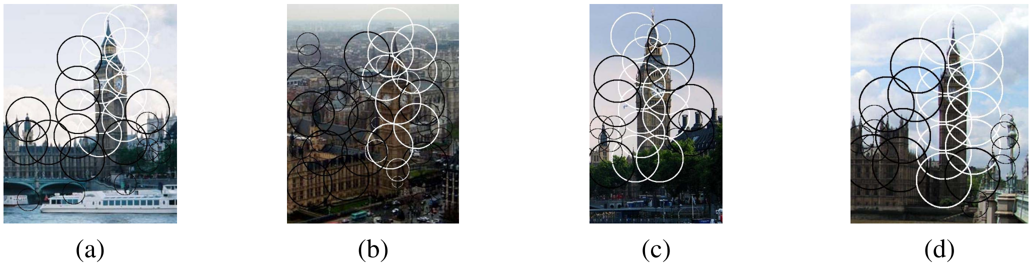


Figure 2: Classification of salient points obtained on some of the training images from the 'Big Ben' class. White salient points are assigned to the true-positive class, black to the false-positive class.

salient points extracted from 40 training image after alignment by the displacements learned with EM. The points assigned to the true and false-positive classes are then presented in (c) and (d), respectively. The magnitude reflects the posterior probability h^{+}_{ik} of assignment to the true-positive class (grayscale ranging from black for '0' to white for '1'). Finally, (e) and (f) present a visualization of the Gaussians, Σ^1 and Σ^0, that compose the configuration model. Note that the true-positive component has a shape that closely resembles that of the object of interest, while the false-positive component has broder support and, therefore, accounts for the salient points not associated with the target. Figure 2 presents some examples of the classification of salient points into true and false-positives. Points depicted in white are true-positives ($h^{+}_{ik} > h^{-}_{ik}$), while false-positives are shown in black.

4 Generation of ROI mask

The "blob-based" configuration model of Figures 1 (e)-(f) can be seen as a template for the saliency configuration of the target object. Given an image on which saliency is to be determined, henceforth refered as the *test* image, a ROI mask is generated by finding the image location at which the saliency map best matches this template. We next describe a MPE rule to determine this location.

4.1 Template matching

The salient point detection procedure of Section 2 is first applied to the test image. To achieve invariance to the scale of the target, the test image is subject to a four-level Gaussian pyramid decomposition, and the procedure repeated at the four levels. The search for the best match to the saliency template is performed at all scales, and the scale with the best match is selected. For the sake of simplicity, we assume a single scale in the discussion that follows. As in (4), the saliency of the test image is represented by a mixture of Gaussians $P_{\mathbf{X}}(\mathbf{x}) = \sum_{i=1}^{N} \pi_i \mathcal{G}(\mathbf{x}, \mathbf{l}_i, s_i^2\mathbf{I})$. The configuration model is centered at location $\mathbf{p}$ and denoted by

$$P_{\mathbf{X}}(\mathbf{x}; \mathbf{p}) = \alpha_1 \mathcal{G}(\mathbf{x}, \mathbf{p}, \mathbf{\Sigma}_1) + \alpha_0 \mathcal{G}(\mathbf{x}, \mathbf{p}, \mathbf{\Sigma}_0). \tag{14}$$

To determine whether the model matches the saliency map of the test image at this location, we rely on the MPE rule for the decision between two hypothesis:

- $\mathcal{H}_0$: the object of interest is not located at $\mathbf{p}$,
- $\mathcal{H}_1$: the object of interest is located at $\mathbf{p}$.

As usual, the posterior probabilties of the two hypothesis are derived, by Bayes rule, from the probabilities of the observed saliency map given the hypothesis. It is assumed that, under hypothesis $\mathcal{H}_0$, salient points are drawn from the false-positive component of the model and, under $\mathcal{H}_1$, they are drawn from the true-positive component. To compute the desired probabilties, we consider a sample of independent observations $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ drawn from the saliency mixture associated with the test image. The log-probability of this sample under model component y ($y \in \{0, 1\}$) of the configuration model is $\sum_{i=1}^{N} \log \mathcal{G}(\mathbf{x}_i, \mathbf{p}, \mathbf{\Sigma}_y)$ and, by the law of large numbers, is equivalent to $NE[\log \mathcal{G}(\mathbf{x}, \mathbf{p}, \mathbf{\Sigma}_y)]$, as N grows to infinity. The expectation is taken with respect to the Gaussian components, e.g. the log-probability of image saliency component i under model component y is, $\forall i \in \{1, \ldots, N\}$ and $y \in \{0, 1\}$,

$$\log p_{i,y}(\mathbf{p}) \quad = \quad N\pi_i \int \mathcal{G}(\mathbf{x}, \mathbf{l}_i, s_i I) \log \mathcal{G}(\mathbf{x}, \mathbf{p}, \mathbf{\Sigma}_y) d\mathbf{x} \tag{15}$$

$$
\begin{aligned}
&= N\pi_i \int \mathcal{G}(\mathbf{x}, \mathbf{l}_i - \mathbf{p}, s_i I) \log \mathcal{G}(\mathbf{x}, \mathbf{0}, \mathbf{\Sigma}_y) d\mathbf{x} \\
&= N\pi_i[-\log(2\pi) - \frac{1}{2}\log|\Sigma_y| - \frac{s_i}{2}\text{trace}(\Sigma_y^{-1}) - (\mathbf{l}_i - \mathbf{p})^T \Sigma_y^{-1}(\mathbf{l}_i - \mathbf{p})].
\end{aligned}
$$

The total log-probability of the saliency map under each of the hypotheses $\mathcal{H}_y$ is the sum (over i) of these log-probabilities, for the associated value of y.

When the test image is much larger than the model template, the number N of salient points can be quite large. Although most of these points are far from the template location $\mathbf{p}$, they produce non-zero probabilities $p_{i,y}$ due to the unbounded spatial support of the Gaussian. For large N, these small contributions can add up to a significant component of the total probability, making the decision-rule more error-prone. To limit the influence of these points, the saliency amplitudes π_i are weighted by a Gaussian window centered at $\mathbf{p}$ and with covariance equal to the average of the covariances in the model. That is, π_i is replaced by $\beta_i = \frac{\pi_i \mathcal{W}(\mathbf{l}_i, \mathbf{p}, \mathbf{\Sigma}_w)}{\sum_j \pi_j \mathcal{W}(\mathbf{l}_j, \mathbf{p}, \mathbf{\Sigma}_w)}$, with $\mathbf{\Sigma}_w = \frac{\Sigma^1 + \Sigma^0}{2}$, and $\mathcal{W}(.)$ a non-normalized Gaussian. Note that this has no probabilistic interpretation, it simply corresponds to a soft-windowing of the image saliency map, to downgrade the contributions of outlier salient points, located far from $\mathbf{p}$.

In addition to the probabilty of the saliency map under each hypothesis, the MPE rule requires the probabilities of the two hypotheses. These are estimated by the total amounts saliency probability mass that fall within (hypothesis $\mathcal{H}_1$) or outside ($\mathcal{H}_1$) the soft template

$$
P(\mathcal{H}_1) = \sum_i \pi_i \mathcal{W}(\mathbf{l}_i, \mathbf{p}, \mathbf{\Sigma}_w), \quad P(\mathcal{H}_0) = 1 - \sum_i \pi_i \mathcal{W}(\mathbf{l}_i, \mathbf{p}, \mathbf{\Sigma}_w).
$$

The MPE rule is a threshold on the log-posterior ratio

$$
L(\mathbf{p}) = \log[\frac{\prod_{i=1}^N p_{i,1}(\mathbf{p}) \times P(\mathcal{H}_1)}{\prod_{i=1}^N p_{i,0}(\mathbf{p}) \times P(\mathcal{H}_0)}],
$$

and the posterior probability of $\mathcal{H}_1$ holding increases monotonically with this ratio. This justifies the use of $L(\mathbf{p})$ as a cost funtion for template matching, namely the determination of the optimal template location by

$$
\begin{aligned}
\mathbf{p}^* &= \arg\max_{\mathbf{p}} L(\mathbf{p}) \qquad (16) \\
&= \arg\max_{\mathbf{p}} \text{trace}\{(\Sigma_0^{-1} - \Sigma_1^{-1}) \sum_{i=1}^N \beta_i (s_i I + (\mathbf{l}_i - \mathbf{p})(\mathbf{l}_i - \mathbf{p})^T)\} + \log \frac{P(\mathcal{H}_1)}{P(\mathcal{H}_0)} \\
&= \arg\max_{\mathbf{p}} \sum_{i=1}^N \beta_i (\mathbf{l}_i - \mathbf{p})^T (\Sigma_0^{-1} - \Sigma_1^{-1})(\mathbf{l}_i - \mathbf{p})^T + \log \frac{P(\mathcal{H}_1)}{P(\mathcal{H}_0)}.
\end{aligned}
$$

4.2 ROI mask

Given the optimal template location $\mathbf{p}^*$, the saliency mask is determined by thresholding the true-postive component of the configuration model of (14),

$$
\text{ROI} = \{\mathbf{x} | \mathcal{G}(\mathbf{x}, \mathbf{p}^*, \Sigma_1) > \alpha\} \qquad (17)
$$

Figure 3 illustrates the various steps of the generation of the ROI mask, for the test image shown in (a). In the figure, (b) depicts the saliency map relative to the class of 'street signs' (brightest pixels representing most salient regions). The template matching cost $L(\mathbf{p})$, with respect to the saliency configuration model of the same class, is shown in (c). The optimal template location $\mathbf{p}^*$ is that of the peak of this cost, and the ROI mask is shown in (d).

5 Training sets

One potentially very rich source of examples for training image classifiers is the Web. Unfortunately, because existing automatic image retrieval systems have relatively small precision, training sets assembled automatically tend to be quite noisy, and must be processed to filter out false-positive images. Because saliency is, by definition, robust to clutter, the saliency detector proposed above can be used for this purpose. In particular, we rely on the saliency detector to attribute a score to each potential training image, as follows.

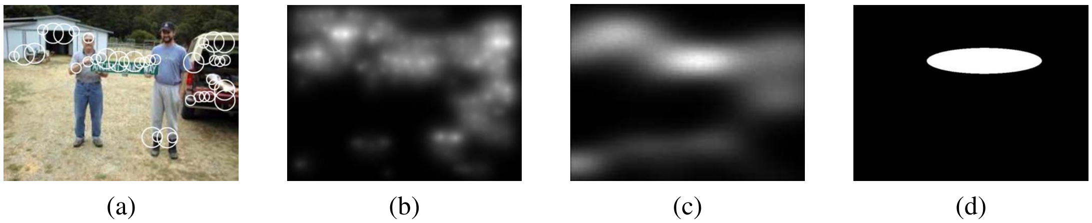


Figure 3: (a) An image and salient points detected with respect to the 'street sign' class, (b) saliency map, (c) map of the matching cost $L(\mathbf{p})$, (d) final ROI mask for the 'street sign' object.

1. a set $\mathcal{S}_1$ of images from a class of interest and a set $\mathcal{S}_0$ of random images that do not depict the class are downloaded from the web, e.g. from the *flickr* web site.
2. the set of features that optimally discriminates bewteen the two classes is determined as described in Section 2.2.
3. saliency maps are produced for all images, and a configuration model learned for the class of interest, using the hierarchical EM procedure of Section 3.3.
4. for each image, the optimal template location $\mathbf{p}^*$ and ROI mask are found with the procedure of Section 4.
5. each image receives a score proportional to the amount of saliency within the ROI

   $$\rho = \sum_{i=1}^{N} S(\mathbf{l}_i)(s_i^2)\mathcal{G}(\mathbf{l}_i, \mathbf{p}^*, \Sigma_1),$$

   where $\mathbf{l}_i, s_i$, and $S(\mathbf{l}_i)$ are defined in (3) and (2), and Σ_1 is the covariance of the true-positive component of the configuration model.
6. images are sorted by decreasing ρ.

6 Experimental results

Since the performance of the image scoring algorithm is indicative of the robustness of the ROI mask generation process, we start with an evaluation of image sorting.

6.1 Image scoring

We tested the image scoring algorithm on the object class of 'cars'. For this, we collected 1,000 matches to the query 'car' on the *flickr* web site to assemble $\mathcal{S}_1$ and 1,000 images from 200 random image categories for $\mathcal{S}_0$. All images were resized so as to have smallest dimension of 240 pixels, with the remaining dimension determined by their original aspect ratio. The downloaded 'car' images were manually labeled as belonging to sub-classes 0, 1, or 2, where 2 indicates images which are good examples for training a 'car' detector, 1 images which are not good examples although they may depict car parts, or some other 'car' related activity, and 0 indicates images which are completly unrelated to the 'car' concept. Among the 1,000 images, 246 fell in sub-class 2, 108 in 1, and 646 in 0, suggesting that only 25% of the images downloaded from the web are useful training examples.

Figure 4 (a) shows the histogram of scores obtained for both the useful (sub-class 2) and useless (union of sub-classes 0 and 1) training images. The center of mass of the score distribution is higher for the former than for the latter and, due to the shape of the distributions, the MPE detector of useful images is a simple score threshold. The vertical line shown in the figure is the optimal threshold, in the MPE sense. Figure 4 (d) presents the ROC curve obtained by varying this threshold. The performance is clearly above chance level, with higher than 90% detection rate for any false positive rate greater than 50%.

6.2 ROI mask accuracy

We next evaluated the accuracy of ROI masks produced by the proposed algorithm. For this, we considered two classes, 'faces' and 'cars', and used the CalTech face database and the UIUC car side database, for which object location groundtruth is available. Since these databases contain training and test sets, we considered two different experiments. In the first, we used the training images provided with each database to train the ROI detector. In the second, training was based on datasets downloaded from the web, using the protocol of the previous section. For each class, 1,000 images were downloaded, a preliminary ROI detector was learned, and the images were sorted with the procedure of Section 5. The images with the largest 100 scores were then selected to form a "clean" training set, which was used to train a final ROI detector.

Comparison of the ROI accuracy on the two experiments is indicative of the robustness of automatically trained ROI detection algorithms to training sets containing both incorrect images and images that, although of the class of interest, have not been pre-processed to limit either the variability of object appearence or the amount of unrelated background. ROI accuracy is measured by comparing the detected objects to the ground truth, using the true positive (TP) and false-positive (FP) rates defined as

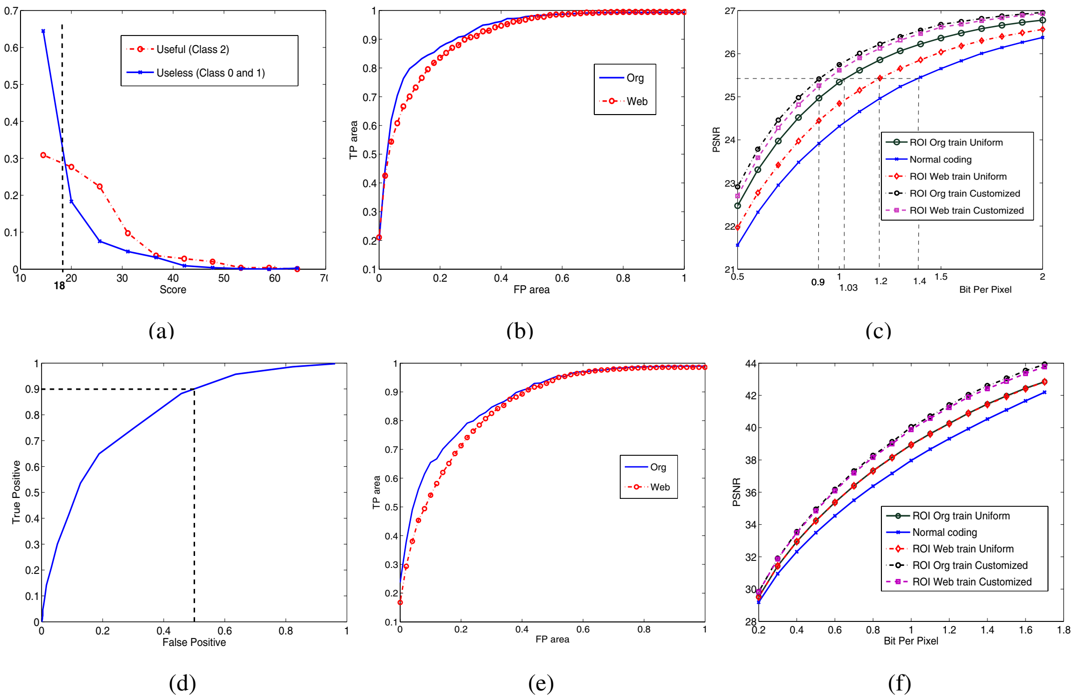
$$TP = \frac{A_{det} \cap A_{truth}}{A_{truth}} \qquad FP = \frac{A_{det} \cap A^c_{truth}}{A^c_{truth}} \tag{18}$$

where A_{det} is the area of the ROI mask, A_{truth} that of the ground truth, and $\cap$ the area of the intersection of the two masks. An ROC curve is generated by varying the threshold α used, in (17), to determine the ROI mask. Figures 4 (b) and (e) present the ROC curves obtained on the UIUC car database (b) and Caltech face database (e). It is interesting to note that, in both cases, the ROC obtained with the manually assembled dataset is only marginally superior to that resulting from the training set assembled automatically from the web. When combined with the results of Figure 4 (a,d), this indicates that the proposed ROI detection algorithm is quite insensitive to the presence of incorrect images on the training set.

6.3 Compression performance

The final evaluation addressed image compression performance. All images were compressed using JPEG2000 ROI-based coding, and the PSNR was measured within the groundtruth area for the object of interest. Various compression possibilities were tested, and, in all cases, the number of bits used to compress the entire image was held constant. The first method, "regular coding", consists of simple application of the JPEG2000 coder without any ROI information. The second method is fully automated: the value of α that, when applied *all* images, maximizes the average PSNR is found by cross-validation during training. Because this value is then used in all test images, the approach is referred to as "ROI Uniform". The third method involves some amount of manual tuning before compression: the threshold α is selected individually for each image, so as to maximize the individual image PSNR. The method is referred to as "ROI Customized", and could be useful when there is room for some amount of interativity before compression. Note that all ROI detection is automatic, except the determination of the threshold which allows the user to scale the ROI up or down. Since threshold selection only requires the manipulation of one variable, it can be easily accomplished even in devices with limited interactivity, e.g. a cell phone camera.

The PSNR curves obtained with all methods are presented in Figure 4 (c) for cars and (f) for faces. For the automated ROI detection methods, the curves are reported for training from both the manually assembled training sets provided with the databases (denoted as "Org") and the datasets automatically collected from the web (denoted as "Web"). It is important to note that the curves are relative to the average PSNR over *all* images, including those where the ROI detector failed to find the object of interest. Various conclusions are possible. First, all ROI-based methods outperform regular JPEG2000 coding. Second, individual tuning of the saliency threshold produces non-negligible gains over the fully automated solution. Finally, the compression performance is quite insensitive to whether the training sets are assembled manually, or with the proposed method. A significant difference only exists for car detection with a uniform ROI threshold. This, once again, illustrates the robustness of the ROI detection algorithm proposed above. At a more quantitative level, the savings of ROI-based coding can be as large as 35.7% of the total number of bits per image for UIUC (where the average size of the car regions is 20% of the image) and 14.29% for Caltech (where the average face covers 32% of the image). In general, achieving a given PSNR on a smaller area requires a smaller fraction of the total number of bits than those required by a larger area.



(a) (b) (c)

(d) (e) (f)

Figure 4: (a) Histograms of scores for training images in the useful and useless classes. (d) ROC curves for the detection of useful images. (b) and (e) ROC curves for object localization on the car and face datasets, respectively. (c) and (f) ROI PSNR under various coding strategies on the car and face datasets, respectively.

References

[1] Sanchez, V.; Basu, A.; Mandal, M.K., Prioritized region of interest coding in JPEG2000, Circuits and Systems for Video Technology, IEEE Transactions on Volume 14, Issue 9, Sept. 2004 Page(s):1149 - 1155

[2] Sanchez, V.; Mandal, M.; Basu, A. Robust, Wireless transmission of regions of interest in JPEG2000, ICIP, 2004.

[3] C.M. Privitera, L.W. Stark, Algorithm for defining visual Regions-of-Interest: Comparison with Eye Fixations, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 22, No. 9: 970-982, 2000

[4] L. Itti, C. Koch, E. Niebur, A Model of Saliency-Based Visual Attention for Rapid Scene Analysis, IEEE Transactions on Patterm Analysis and Machine Intelligence Vol. 20, No. 11: 1254-1259, 1998

[5] C. Koch, S. Ullman, Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry, Human Neurobiology, Vol. 4: 219-227, 1985

[6] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, CVPR 2001, Vol. 1: 511-518

[7] D. Gao, N. Vasconcelos. Integrated learning of saliency, complex features, and object detectors from cluttered scenes, CVPR 2005. Volume 2, 10.25 June 2005 Page(s):282 - 287 vol. 2

[8] D. Gao, N. Vasconcelos, Discriminant Saliency for Visual Recognition from Cluttered Scenes, NIPS, 2004

[9] N. Vasconcelos, A. Lippman, Learning Mixture Hierarchies, NIPS 11, Denver, Colorado, 1998

[10] Carron, T.; Lambert, P. Color edge detector using jointly hue, saturation and intensity, ICIP 1994

[11] C. Koch, S. Ullman, "Shift in selective visual attention: towards the underlying neural circuitry," Hum. Neyrobiol., vol. 4, pp. 219-227, 1985.

[12] S. Agarwal, A Awan, and D. Roth, Learning to Detect Objects in images via a Sparse, Part-Based Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 26, No. 11,: 1475-1490, 2004

[13] A. Yarbus, Eye movements and vision. Plenum, New York, 1967

[14] A. Treisman. Preattentive processing in vision. Computer vision, Graphics, Image Processing, 31, 156 - 177, 1985

[15] J. Wolfe, Guided search 2.0: A revised model of visual search, Psychonomic Bulletic, Review, 202 - 238, 1994

[16] J. Duncan, G. Humphreys, Visual search surface: visual search and attentional engagement, Journal of Experimental Psychology: Human Perception and Performance, 18(2), 578 - 588, 1992

[17] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning. CVPR 2003

[18] P. Felzenszwalb, D. Hutenlocher, Pictorial structures for object recognition, International Journal of Computer Vision, 61, 55 - 79, 2005