Submitted to Journal of Vision, May 2009

Static and Space-time Visual Saliency Detection by Self-Resemblance

Hae Jong Seo and Peyman Milanfar
Electrical Engineering Department
University of California, Santa Cruz
1156 High Street, Santa Cruz, CA, 95064
{rokaf,milanfar}@soe.ucsc.edu

Abstract

We present a novel unified framework for both static and space-time saliency detection. Our method is a bottom-up approach and computes so-called local regression kernels (i.e., local descriptors) from the given image (or a video), which measure the likeness of a pixel (or voxel) to its surroundings. Visual saliency is then computed using the said "self-resemblance" measure. The framework results in a saliency map where each pixel (or voxel) indicates the statistical likelihood of saliency of a feature matrix given its surrounding feature matrices. As a similarity measure, matrix cosine similarity (a generalization of cosine similarity) is employed. State of the art performance is demonstrated on commonly used human eye fixation data (static scenes [5] and dynamic scenes [16]) and some psychological patterns.

I. Introduction

Visual saliency detection has been of great research interest [5], [8], [10], [14], [17], [36], [43], [42], [18], [24], [13] in recent years. Analysis of visual attention is considered a very important component in the human vision system because of a wide range of applications such as object detection, predicting human eye fixation, video summarization [23], image quality assessment [20], [26] and more. In general, saliency is defined as what drives human perceptual attention. There are two types of computational models for saliency according to what the model is driven by: a bottom-up saliency [5], [8], [14], [17], [43], [42], [24], [13] and a top-down saliency [10],

[36], [18]. As opposed to bottom-up saliency algorithms that are fast and driven by low-level features, top-down saliency algorithms are slower and task-driven.

The problem of interest addressed in this paper is bottom-up saliency which can be described as follows: Given an image or a video, we are interested in accurately detecting salient objects or actions from the data without any background knowledge. To accomplish this task, we propose to use, as features, so-called *local steering kernels* and *space-time local steering kernels* which capture local data structure exceedingly well. Our approach is motivated by a probabilistic framework, which is based on a nonparametric estimate of the likelihood of saliency. As we describe below, this boils down to the local calculation of a "self-resemblance" map, which measures the similarity of a feature matrix at a pixel of interest to its neighboring feature matrices.

A. Previous work

Itti et al. [17] introduced a saliency model which was biologically inspired. Specifically, they proposed the use of a set of feature maps from three complementary channels as intensity, color, and orientation. The normalized feature maps from each channel were then linearly combined to generate the overall saliency map. Even though this model has been shown to be successful in predicting human fixations, it is somewhat ad-hoc in that there is no objective function to be optimized and many parameters must be tuned by hand. With the proliferation of eye-tracking data, a number of researchers have recently attempted to address the question of what attracts human visual attention by being more mathematically and statistically precise [5], [8], [9], [10], [16], [43], [13].

Bruce and Tsotsos [5] modeled bottom-up saliency as the maximum information sampled from an image. More specifically, saliency is computed as Shannon's self-information $-\log p(\mathbf{f})$, where $\mathbf{f}$ is a local visual feature vector (i.e., derived from independent component analysis (ICA) performed on a large sample of small RGB patches in the image.) The probability density function is estimated based on a Gaussian kernel density estimate in a neural circuit.

Gao et al. [8], [9], [10] proposed a unified framework for top-down and bottom-up saliency as a classification problem with the objective being the minimization of classification error. They first applied this framework to object detection [10] in which a set of features are selected such that a class of interest is best discriminated from all other classes, and saliency is defined as the

weighted sum of features that are salient for that class. In [8], they defined bottom-up saliency using the idea that pixel locations are salient if they are distinguished from their surroundings. They used difference of Gaussians (DoG) filters and Gabor filters, measuring the saliency of a point as the Kullback-Leibler (KL) divergence between the histogram of filter responses at the point and the histogram of filter responses in the surrounding region. Mahadevan and Vasconcelos [22] applied this bottom-up saliency to background subtraction in highly dynamic scenes.

Oliva and Torralba [27], [36] proposed a Bayesian framework for the task of visual search (i.e., whether a target is present or not.) They modeled bottom-up saliency as $\frac{1}{p(\mathbf{f}|\mathbf{f}_G)}$ where $\mathbf{f}_G$ represents a global feature that summarizes the appearance of the scene and approximated this conditional probability density funtion by fitting to a multivariate exponential distribution. Zhang et al. [43] also proposed saliency detection using natural statistics (SUN) based on a similar Bayesian framework to estimate the probability of a target at every location. They also claimed that their saliency measure emerges from the use of Shannon's self-information under certain assumptions. They used ICA features as similarly done in [5], but their method differs from [5] in that natural image statistics were applied to determine the density function of ICA featuers. Itti and Baldi [16] proposed so-called "Bayesian Surprise" and extended it to the video case [15]. They measured KL-divergence between a prior distribution and posterior distribution as a measure of saliency.

For saliency detection in video, Marat et al. [24] proposed a space-time saliency detection algorithm inspired by the human visual system. They fused a static saliency map and a dynamic saliency map to generate the space-time saliency map. Gao et al. [8] adopted a dynamic texture model using a Kalman filter in order to capture the motion patterns even in the case that the scene is itself dynamic. Zhang et al. [42] extended their SUN framework to a dynamic scene by introducing temporal filter (Difference of Exponential:DoE) and fitting a generalized Gaussian distribution to the estimated distribution for each filter response.

Most of the methods [8], [17], [27], [42] based on Gabor or DoG filter responses require many design parameters such as the number of filters, type of filters, choice of the nonlinearities, and a proper normalization scheme. These methods tend to emphasize textured areas as being salient regardless of their context. In order to deal with these problems, [5], [43] adopted non-linear features that model complex cells or neurons in higher levels of the visual system. Kienzle et al. [19] further proposed to learn a visual saliency model directly from human eyetracking data

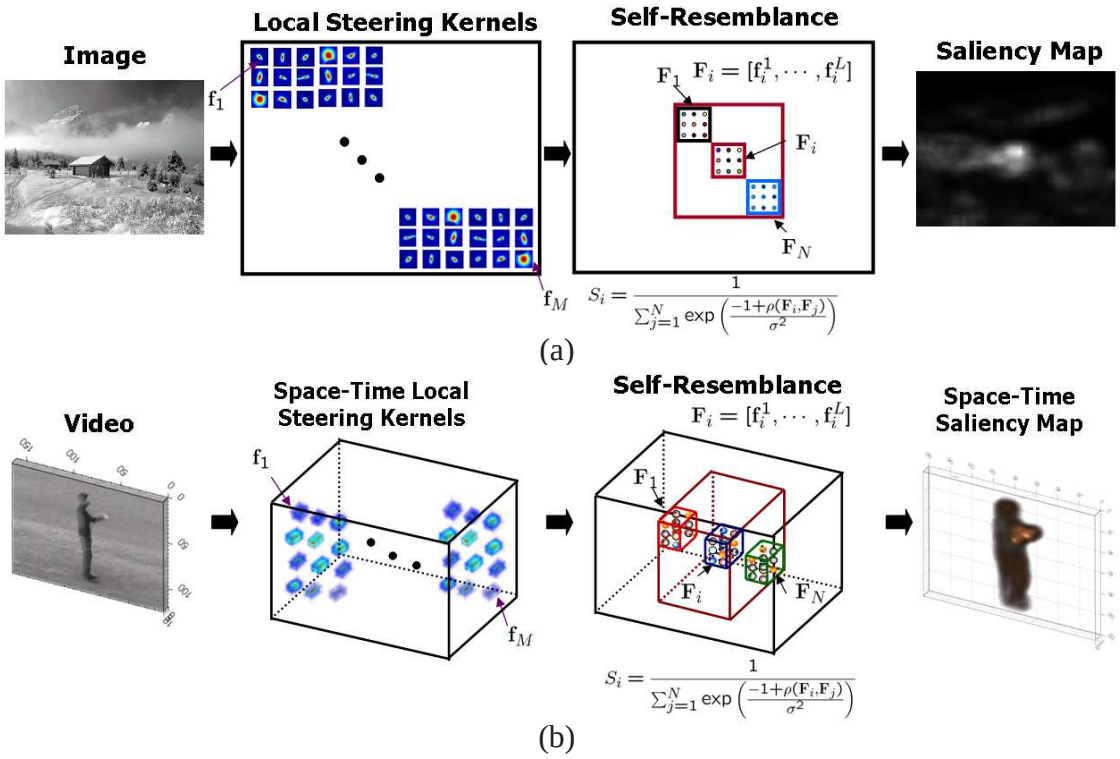


Fig. 1. Graphical overview of saliency detection system (a) static saliency map (b) space-time saliency map. Note that the number of neighboring features N in (b) is obtained from a space-time neighborhood.

using a support vector machine (SVM).

Different from traditional image statistical models, a spectral residual (SR) approach based on the Fourier transform was recently proposed by Hou and Zhang [14]. The spectral residual approach does not rely on parameters and detects saliency rapidly. In this approach, the difference between the log spectrum of an image and its smoothed version is the spectral residual of the image. However, Guo and Zhang [12] claimed that what plays an important role for saliency detection is not SR, but the image's phase spectrum. Recently, Hou and Zhang [13] proposed a dynamic visual attention model by setting up an objective function to maximize the entropy of the sampled visual features based on the incremental coding length.

B. Overview of the Proposed Approach

In this paper, our contributions to the saliency detection task are three-fold. First we propose to use local regression kernels as features which capture the underlying local structure of the

data exceedingly well, even in the presence of significant distortions. Second we propose to use a nonparametric kernel density estimation for such features, which results in a saliency map constructed from a local "self-resemblance" measure, indicating likelihood of saliency. Lastly, we provide a simple, but powerful unified framework for both static and space-time saliency detection. The original motivation behind these contributions is the earlier work on adaptive kernel regression for image and video reconstruction [34], [35] and nonparametric object detection[1] [29] and action recognition[2] [30].

As similarly done in Gao et al. [8], we measure saliency at a pixel in terms of how much it stands out from its surroundings. To formalize saliency at each pixel, we let the binary random variable y_i denote whether a pixel position $\mathbf{x}_i = [x_1, x_2]_i^T$ is salient or not as follows:

$$y_i = \begin{cases} 1, & \text{if} \quad \mathbf{x}_i \quad \text{is} \quad \text{salient}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $i = 1, \cdots, M$, and M is the total number of pixels in the image. Motivated by the approach in [43], [27], we define saliency at pixel position $\mathbf{x}_i$ as a posterior probability $Pr(y_i = 1|\mathbf{F})$ as follows:

$$S_i = Pr(y_i = 1|\mathbf{F}), \tag{2}$$

where the feature matrix, $\mathbf{F}_i = [\mathbf{f}_i^1, \cdots, \mathbf{f}_i^L]$ at pixel of interest $\mathbf{x}_i$ (what we call a center feature,) contains a set of feature vectors ($\mathbf{f}_i$) in a local neighborhood where L is the number of features in that neighborhood[3]. In turn, the larger collection of features $\mathbf{F} = [\mathbf{F}_1, \cdots, \mathbf{F}_N]$ is a matrix containing features not only from the center, but also a surrounding region (what we call a center+surround region; See Fig. 2.) N is the number of feature matrices in the center+surround region. Using Bayes' theorem, Equation (2) can be written as

$$S_i = Pr(y_i = 1|\mathbf{F}) = \frac{p(\mathbf{F}|y_i = 1)Pr(y_i = 1)}{p(\mathbf{F})}. \tag{3}$$

By assuming that 1) a-priori, every pixel is considered to be equally likely to be salient; and 2) $p(\mathbf{F})$ are uniform over features, the saliency we defined boils down to the conditional probability density $p(\mathbf{F}|y_i = 1)$.

[1]Available online from http://users.soe.ucsc.edu/~rokaf/paper/TrainingFreeGenericObjectDetection.pdf

[2]Available online from http://users.soe.ucsc.edu/~rokaf/paper/IJCV_ActionRecognition_Final_Mar27.pdf

[3]Note that if $L = 1$, we use a single feature vector. Using a feature matrix consisting of a set of feature vectors provides more discriminative power than using a single feature vector as also pointed out in [41], [3].

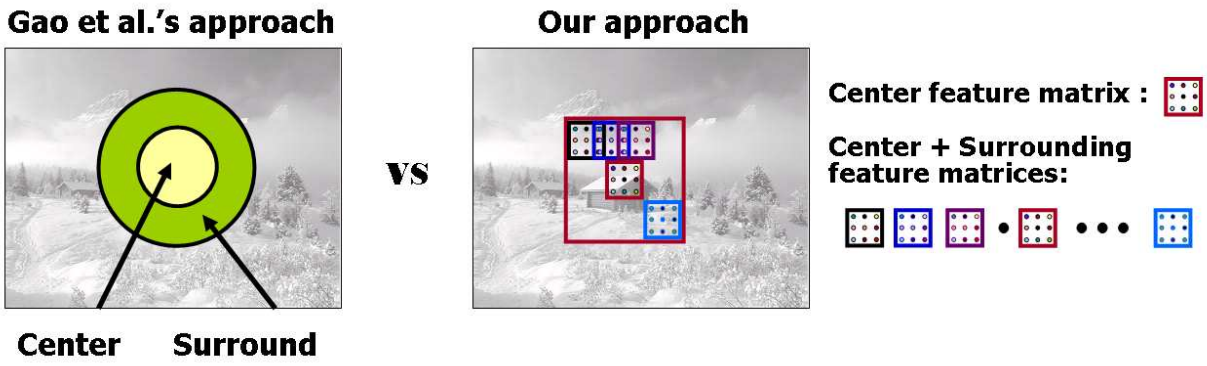


Fig. 2. Illustration of difference between Gao el al. [8]'s approach and our approach about a center-surround definition.

Since we do not know the conditional probability density $p(\mathbf{F}|y_i = 1)$, we need to estimate it. It is worth noting that Gao et al. [8] and Zhang et al. [43] fit the marginal density of local feature vectors $p(\mathbf{f})$ to a generalized Gaussian distribution. However, in this paper, we approximate the conditional density function $p(\mathbf{F}|y_i = 1)$ based on nonparametric kernel density estimation which will be explained in detail in Section II-B.

Before we begin a more detailed description, it is worthwhile to highlight some aspects of our proposed framework. While the state-of-the art methods [5], [8], [16], [43] are related to our method, their approaches fundamentally differ from ours in the following respects: 1) While they use Gabor filters, DoG filters, or ICA to derive features, we propose to use local steering kernels (LSK) which are highly nonlinear but stable in the presence of uncertainty in the data [34]. In addition, normalized local steering kernels provide a certain invariance as shown in Fig. 4; 2) As opposed to [8], [43] which model marginal densities of band-pass features as a generalized Gaussian distribution, we estimate the conditional probability density $p(\mathbf{F}|y_i = 1)$ using nonparametric kernel density estimation; 3) While Itti and Baldi [16] computed, as a measure of saliency, KL-divergence between a prior and a posterior distribution, we explicitly estimate the likelihood function directly using nonparametric kernel density estimation; 4) Our space-time saliency detection method does not require explicit motion estimation; 5) The proposed unified framework can handle both static and space-time saliency detection. Fig. 1 shows an overview of our proposed framework for saliency detection. To summarize the operation of the overall algorithm, we first compute the normalized local steering kernels (space-time local steering kernels) from the given image (video) I and vectorize them as $\mathbf{f}$'s. Then, we identify features $\mathbf{F}_i$

centered at a pixel of interest $\mathbf{x}_i$, and a set of feature matrices $\mathbf{F}_j$ in a center+surrounding region and compute the self-resemblance measure (See Equations (13) and (14).) The final saliency map is given as a density map as shown in Fig 1. A shorter version of this paper[4] was accepted for the IEEE Conference on Computer Vision and Pattern Recognition, 1st International Workshop on Visual Scene Understanding (ViSU09) [31].

In the next section, we provide further technical details about the steps outlined above. In Section III, we demonstrate the performance of the system with experimental results, and we conclude this paper in Section IV.

II. Technical Details

A. Local Regression Kernel as a Feature

1) Local Steering Kernel (2-D LSK): The key idea behind local steering kernels is to robustly obtain the local structure of images by analyzing the radiometric (pixel value) differences based on estimated gradients, and use this structure information to determine the shape and size of a canonical kernel. The local steering kernel is modeled as

$$K(\mathbf{x}_l - \mathbf{x}_i) = \frac{\sqrt{\det(\mathbf{C}_l)}}{h^2} \exp\left\{\frac{(\mathbf{x}_l - \mathbf{x}_i)^T \mathbf{C}_l (\mathbf{x}_l - \mathbf{x}_i)}{-2h^2}\right\}, \quad \mathbf{C}_l \in \mathbb{R}^{2\times 2}, \tag{4}$$

where $l \in \{1, \cdots, P\}$, P is the number of pixels in a local window; h is a global smoothing parameter, and the matrix $\mathbf{C}_l$ is a covariance matrix estimated from a collection of spatial gradient vectors within the local analysis window around a position $\mathbf{x}_l = [x_1, x_2]_l^T$ (See Fig. 3 (a).)

2) Space-Time Local Steering Kernel (3-D LSK): Now, we introduce the time axis to the data model so that $\mathbf{x}_l = [x_1, x_2, t]_l^T$: x_1 and x_2 are the spatial coordinates, t is the temporal coordinate. In this setup, the covariance matrix $\mathbf{C}_l$ can be naively estimated as $\mathbf{J}_l^T \mathbf{J}_l$ with

$$\mathbf{J}_l = \begin{bmatrix} z_{x_1}(\mathbf{x}_1), & z_{x_2}(\mathbf{x}_1), & z_t(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ z_{x_1}(\mathbf{x}_P), & z_{x_2}(\mathbf{x}_P), & z_t(\mathbf{x}_P) \end{bmatrix}$$

where $z_{x_1}(\cdot), z_{x_2}(\cdot)$, and $z_t(\cdot)$ are the first derivatives along x_1-, x_2-, and $t-$ axes, and P is the total number of samples in a *space-time* local analysis window (or cube) around a sample

[4] Available online from http://users.soe.ucsc.edu/~rokaf/paper/CVPR2009_saliency_CameraReady.pdf

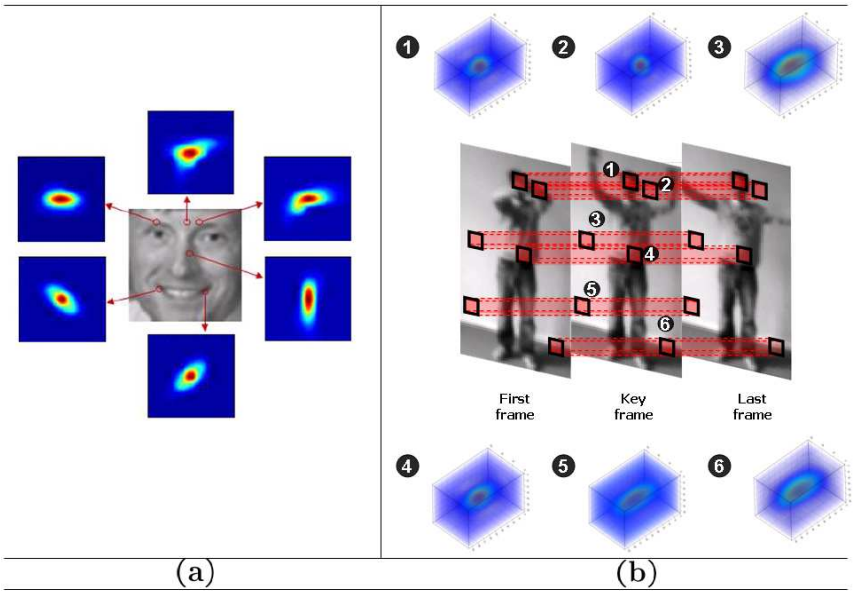


Fig. 3. (a) Examples of 2-D LSK in various regions. (b) Examples of space-time local steering kernel (3-D LSK) in various regions. Note that key frame means the frame where the center of 3-D LSK is located.

position at $\mathbf{x}_i$. For the sake of robustness, we compute a more stable estimate of $\mathbf{C}_l$ by invoking the singular value decomposition (SVD) of $\mathbf{J}_l$ with regularization as [35]:

$$\mathbf{C}_l = \gamma_l \sum_{q=1}^{3} a_q^2 \mathbf{v}_q \mathbf{v}_q^T \in \mathbb{R}^{(3\times 3)}, \tag{5}$$

with

$$a_1 = \frac{s_1 + \lambda'}{\sqrt{s_2 s_3} + \lambda'}, \quad a_2 = \frac{s_2 + \lambda'}{\sqrt{s_1 s_3} + \lambda'}, \quad a_3 = \frac{s_3 + \lambda'}{\sqrt{s_1 s_2} + \lambda'}, \quad \gamma_i = \left(\frac{s_1 s_2 s_3 + \lambda''}{P}\right)^{\alpha} \tag{6}$$

where λ' and λ'' are regularization parameters that dampen the noise effect and restrict γ_i and the denominators of a_q's from being zero. The singular values (s_1, s_2, and s_3) and the singular vectors ($\mathbf{v}_1, \mathbf{v}_2$, and $\mathbf{v}_3$) are given by the compact SVD of $\mathbf{J}_l$:

$$\mathbf{J}_l = \mathbf{U}_l \mathbf{S}_l \mathbf{V}_l^T = \mathbf{U}_l \mathrm{diag}[s_1, s_2, s_3][\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]^T, \tag{7}$$

Then, the covariance matrix $\mathbf{C}_l$ modifies the shape and size of the local kernel in a way which robustly encodes the space-time local geometric structures present in the video (See Fig. 3 (b)

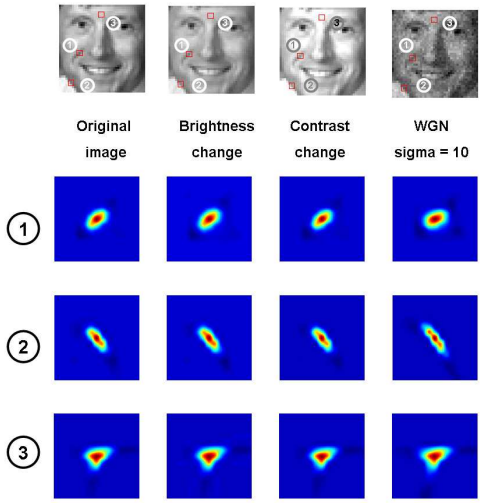


Fig. 4. Invariance and robustness of LSK weights $W(\mathbf{x}_l - \mathbf{x}_i)$ in various challenging conditions. Note that WGN means White Gaussian Noise.

for an example.) Similarily to 2D case, 3-D LSKs are formed as follows:

$$K(\mathbf{x}_l - \mathbf{x}_i) = \frac{\sqrt{\det(\mathbf{C}_l)}}{h^2} \exp\left\{ -\frac{(\mathbf{x}_l - \mathbf{x}_i)^T \mathbf{C}_l (\mathbf{x}_l - \mathbf{x}_i)}{2h^2} \right\}, \quad \mathbf{C}_l \in \mathbb{R}^{(3\times 3)}. \tag{8}$$

In the 3-D case, orientation information captured in 3-D LSK contains the motion information implicitly [35]. It is worth noting that a significant strength of using this implicit framework (as opposed to the direct use of estimated motion vectors) is the flexibility it provides in terms of smoothly and adaptively changing the parameters defined by the singular values in Equation 6. This flexibility allows the accommodation of even complex motions, so long as their magnitudes are not excessively large. For a more in depth analysis of local steering kernels, we refer the interested reader to [34], [35].

In what follows, at a position $\mathbf{x}_i$, we will essentially be using (a normalized version of) the function $K(\mathbf{x}_l - \mathbf{x}_i)$. To be more specific, the local steering kernel function $K(\mathbf{x}_l - \mathbf{x}_i)$ is calculated at every pixel location and normalized as follows

$$W(\mathbf{x}_l - \mathbf{x}_i) = \frac{K(\mathbf{x}_l - \mathbf{x}_i)}{\sum_{l=1}^{P} K(\mathbf{x}_l - \mathbf{x}_i)}, \quad i = 1, \cdots, M. \tag{9}$$

From a human perception standpoint [36], it has been shown that local image features are salient when they are distinguishable from the background. Computationally, measuring saliency requires, as we have seen, the estimation of local feature distributions in an image. For this purpose, a generalized Gaussian distribution is often employed as in [8], [36], [43].

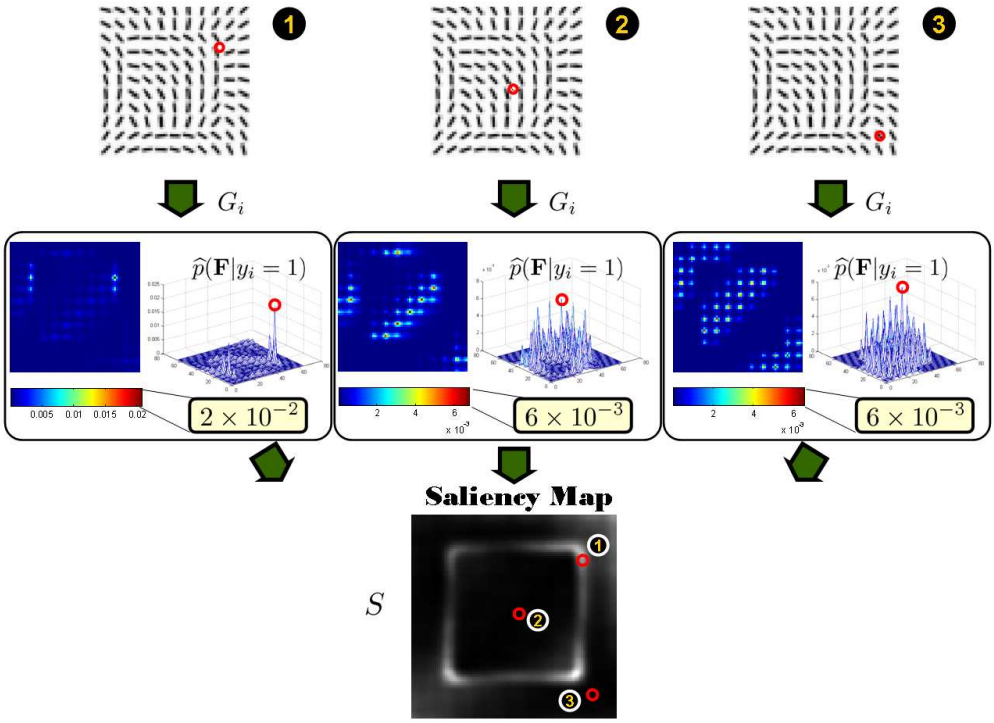


Fig. 5. Example of saliency computation in psychological pattern. Note that center+surrounding regions to compute Self-Resemblance is as large as the entire image in this case. i.e., $N = M$

However, LSK features follow a power-law distribution (a long-tail distribution) [29]. In other words, the LSK features are scattered out in a high dimensional feature space, and thus there basically exists no dense cluster in this feature space. Instead of using a generalized Gaussian model for this data, we employed a locally adaptive kernel density estimation method which we explain in the next section.

B. Saliency by Self-Resemblance

As we alluded to in Section I-B, saliency at a pixel $\mathbf{x}_i$ is measured using the conditional density of the feature matrix at that position: $S_i = p(\mathbf{F}|y_i = 1)$. Hence, the task at hand is to estimate $p(\mathbf{F}|y_i = 1)$ over $i = 1, \cdots, M$. In general, the Parzen density estimator is a simple and generally accurate non-parametric density estimation method [33]. However, in higher dimensions and with an expected long-tail distribution, the Parzen density estimator with an isotropic kernel is not the most appropriate tool [2], [4], [39]. As explained earlier, the LSK features tend to generically come from long-tailed distributions, and as such, there are generally

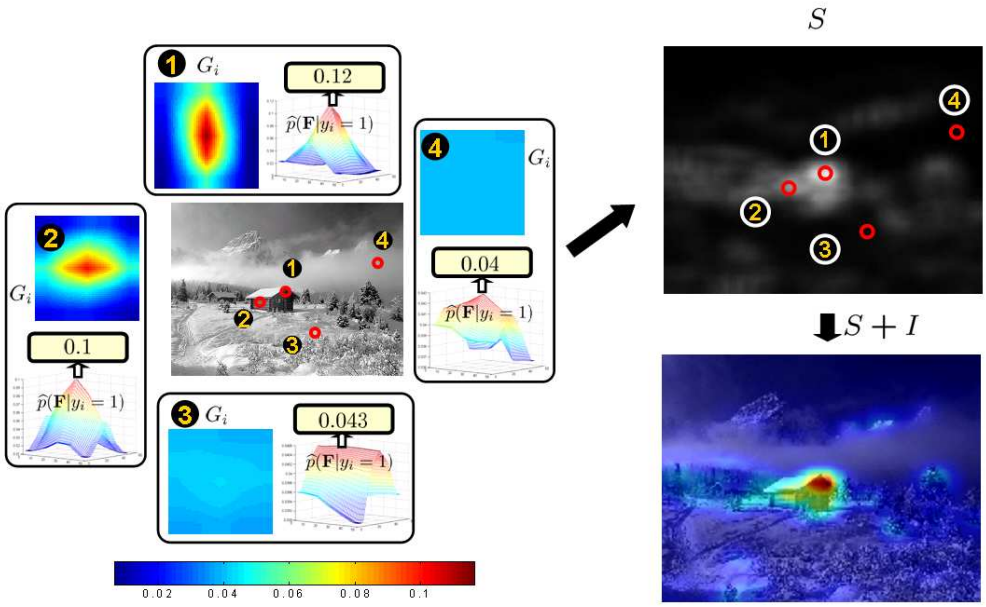


Fig. 6. Example of saliency computation in natural gray-scale image. Note that center+surrounding regions to compute self-resemblance is a local neighborhood in this case. i.e., $N << M$. Note that red values in saliency map represent higher saliency, while blue values mean lower saliency.

no tight clusters in the feature space. When we estimate a probability density at a particular feature point, for instance $\mathbf{F}_i = [\mathbf{f}_i^1, \cdots, \mathbf{f}_i^L]$ (where L is the number of vectorized LSKs ($\mathbf{f}$'s) employed in the feature matrix), the isotropic kernel centered on that feature point will spread its density mass equally along all the feature space directions, thus giving too much emphasis to irrelevant regions of space and too little along the manifold. Earlier studies [2], [4], [39] also pointed out this problem. This motivates us to use *a locally data-adaptive kernel density estimator*. We define the conditional probability density $p(\mathbf{F}|y_i = 1)$ at $\mathbf{x}_i$ as a center value of a normalized adaptive kernel (weight function) $G(\cdot)$ computed in the center+surround region as follows:

$$S_i = \widehat{p}(\mathbf{F}|y_i = 1) = \frac{G_i(\overline{\mathbf{F}}_i - \overline{\mathbf{F}}_i)}{\sum_{j=1}^{N} G_i(\overline{\mathbf{F}}_i - \overline{\mathbf{F}}_j)}, \tag{10}$$

where $G_i(\overline{\mathbf{F}}_i - \overline{\mathbf{F}}_j) = \exp\left(\frac{-||\overline{\mathbf{F}}_i - \overline{\mathbf{F}}_j||_F^2}{2\sigma^2}\right)$, $||\cdot||_F$ is the Frobenious norm, $\overline{\mathbf{F}}_i = \left[\frac{\mathbf{f}_i^1}{\|\mathbf{F}_i\|_F}, \cdots, \frac{\mathbf{f}_i^L}{\|\mathbf{F}_i\|_F}\right]$ and $\overline{\mathbf{F}}_j = \left[\frac{\mathbf{f}_j^1}{\|\mathbf{F}_j\|_F}, \cdots, \frac{\mathbf{f}_j^L}{\|\mathbf{F}_j\|_F}\right]$, and σ is a parameter controlling the fall-off of weights.

Inspired by earlier works such as [6], [7], [21], [29] that have shown the effectiveness of correlation-based similarity, the kernel function G_i in Equation (10) can be rewritten using the

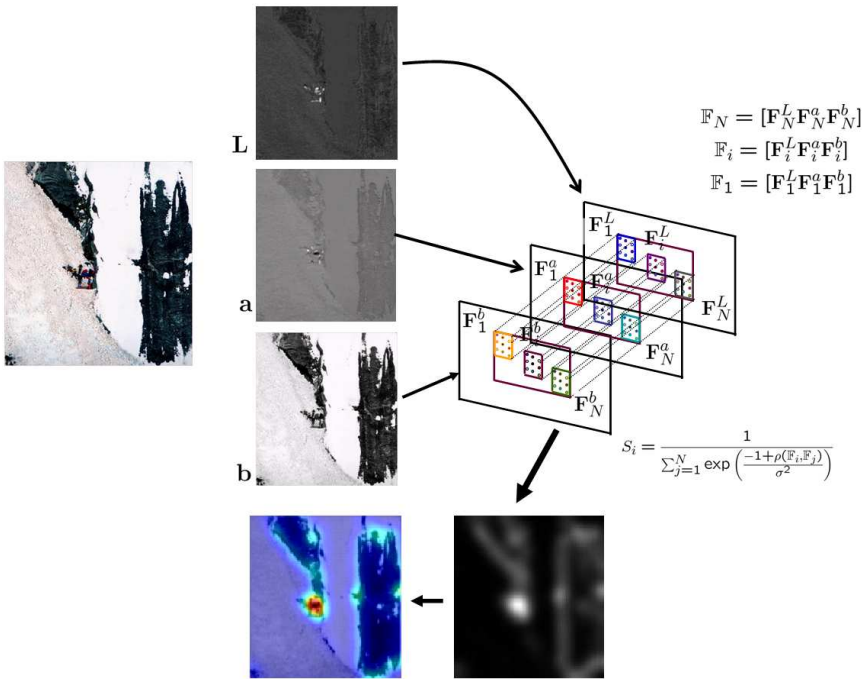


Fig. 7. As a example of saliency detection in a color image (in this case, CIE L*a*b*), we show how saliency is computed using matrix cosine similarity.

concept of matrix cosine similarity [29] as follows:

$$G_i(\overline{\mathbf{F}}_i - \overline{\mathbf{F}}_j) = \exp\left(\frac{-1+\rho(\mathbf{F}_i,\mathbf{F}_j)}{\sigma^2}\right), \quad j = 1, \cdots, N, \tag{11}$$

where $\rho(\mathbf{F}_i, \mathbf{F}_j)$ is the "Matrix Cosine Similarity (MCS)" between two feature matrices $\mathbf{F}_i, \mathbf{F}_j$ and is defined as the "Frobenius inner product" between two normalized matrices $\rho(\mathbf{F}_i,\mathbf{F}_j) =< \overline{\mathbf{F}}_i, \overline{\mathbf{F}}_j >_F = \text{trace}\left(\frac{\mathbf{F}_i^T \mathbf{F}_j}{\|\mathbf{F}_i\|_F \|\mathbf{F}_j\|_F}\right) \in [-1, 1]$. This matrix cosine similarity can be rewritten as a weighted sum of the vector cosine similarities [6], [7], [21] $\rho(\mathbf{f}_i, \mathbf{f}_j)$ between each pair of corresponding feature vectors (i.e., columns) in $\mathbf{F}_i, \mathbf{F}_j$ as follows:

$$\rho_i = \sum_{\ell=1}^{L} \frac{\mathbf{f}_i^{\ell T} \mathbf{f}_j^{\ell}}{\|\mathbf{F}_i\|_F \|\mathbf{F}_j\|_F} = \sum_{\ell=1}^{L} \rho(\mathbf{f}_i^{\ell}, \mathbf{f}_j^{\ell}) \frac{\|\mathbf{f}_i^{\ell}\| \|\mathbf{f}_j^{\ell}\|}{\|\mathbf{F}_i\|_F \|\mathbf{F}_j\|_F}. \tag{12}$$

The weights are represented as the product of $\frac{\|\mathbf{f}_i^{\ell}\|}{\|\mathbf{F}_i\|_F}$ and $\frac{\|\mathbf{f}_j^{\ell}\|}{\|\mathbf{F}_j\|_F}$ which indicate the relative importance of each feature in the feature sets $\mathbf{F}_i, \mathbf{F}_j$. This measure[5] not only generalizes the cosine

[5]This measure can be efficiently computed by column-stacking the matrices $\mathbf{F}_i, \mathbf{F}_j$ and simply computing the cosine similarity between two long column vectors.

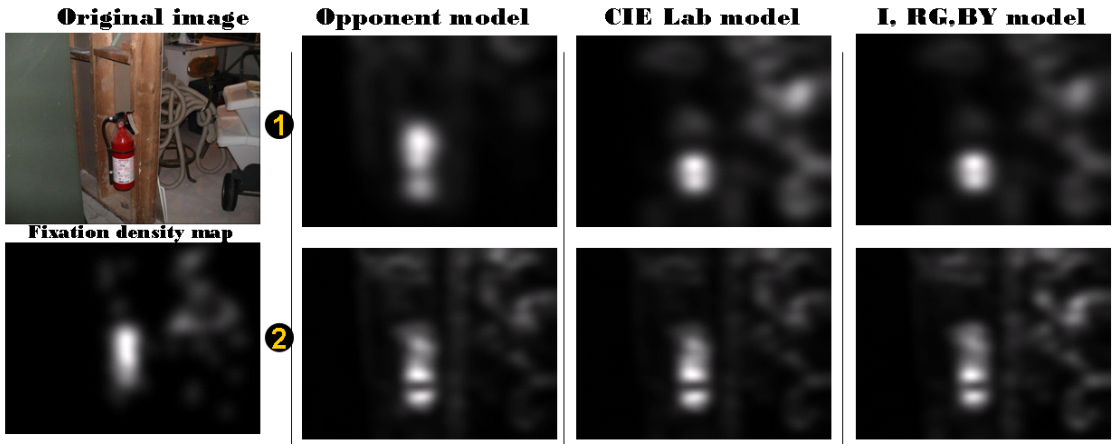


Fig. 8. Comparisons between (1) Simple normalized summation and (2) The use of matrix cosine similarity without any fusion in three different color spaces. Simple normalized summation method tends to be dominated by a particular chrominance information. It is clearly shown that using matrix cosine similarity provides consistent results than the simple normalized summation fusion method.

similarity, but also overcomes the disadvantages of the conventional Euclidean distance which is sensitive to outliers.

Fig. 5 describes what kernel functions G_i look like in various regions of a psychological pattern image[6]. As shown in Fig. 5, each kernel function G_i has a unique peak value at $\mathbf{x}_i$ which represents a likelihood of the pixel $\mathbf{x}_i$ being salient given feature matrices in the center+surrounding region. Therefore, saliency at $\mathbf{x}_i$ ($S_i = \widehat{p}(\mathbf{F}|y_i = 1)$) is the center value of (the normalized version) of the weight function G_i which contains contributions from all the surrounding feature matrices. Specifically, S_i is computed by inserting Equation (11) into Equation (10) as follows:

$$S_i = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1+\rho(\mathbf{F}_i,\mathbf{F}_j)}{\sigma^2}\right)}. \tag{13}$$

As a consequence, $\widehat{p}(\mathbf{F}|y_i = 1)$ reveals how $\mathbf{F}_i$ is salient given all the features $\mathbf{F}_j$'s in a neighborhood. Fig. 6 illustrates how these values computed from a natural image provide a reliable saliency measure.

[6]The image came from the website, http://www.svcl.ucsd.edu/projects/discsalbu/

C. Handling color images

Up to now, we only dealt with saliency detection in a grayscale image. If we have color input data, we need an approach to integrate saliency information from all color channels. To avoid some drawbacks of earlier methods [17], [25], we do not combine saliency maps from each color channel linearly and directly. Instead we utilize the idea of matrix cosine similarity. More specifically, we first identify feature matrices from each color channel c_1, c_2, c_3 as $\mathbf{F}_i^{c_1}, \mathbf{F}_i^{c_2}, \mathbf{F}_i^{c_3}$ as shown in Fig. 7. By collecting them as a larger matrix $\mathbb{F}_i = [\mathbf{F}_i^{c_1}, \mathbf{F}_i^{c_2}, \mathbf{F}_i^{c_3}]$, we can apply matrix cosine similarity between $\mathbb{F}_i$ and $\mathbb{F}_j$. Then, the saliency map from color channels can be analogously defined as follows:

$$S_i = \widehat{p}(\mathbb{F}|y_i = 1) = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1+\rho(\mathbb{F}_i,\mathbb{F}_j)}{\sigma^2}\right)}. \tag{14}$$

In order to verify that this idea allows us to achieve a consistent result and leads us to a better performance than using fusion methods, we have compared three different color spaces [7] ; namely opponent color channels [38], CIE L*a*b* [29], [32] channels, and I R-G B-Y channels [43]

Fig. 8 compares saliency maps using simple normalized summation of saliency maps from different channels as compared to using matrix cosine similarity. It is clearly seen that using matrix cosine similarity provides consistent results regardless of color spaces and helps to avoid some drawbacks of fusion-based methods. To summarize, the overall pseudo-code for the algorithm is given in **Algorithm** 1.

III. Experimental Results

In this section, we demonstrate the performance of the proposed method with comprehensive experiments in terms of 1) interest region detection; 2) prediction of human fixation data; and 3) performance on psychological patterns. Comparison is made with other state-of-the-art methods both quantitatively and qualitatively.

[7] Opponent color space has proven to be superior to RGB, HSV, normalized RGB, and more in the task of object and scene recognition [38]. Shechman and Irani [32] and Seo and Milanfar [29] showed that CIE L*a*b* performs well in the task of object detection.

Algorithm 1 Visual Saliency Detection Algorithm

I : input image or video, P : size of local steering kernel (LSK) or 3-D LSK window, h : a global smoothing parameter for LSK, L : number of LSK or 3-D LSK used in the feature matrix, N : size of a center+surrounding region for computing self-resemblance, σ : a parameter controlling fall-off of weights for computing self-resemblance.

$\mathbf{Stage1}$: $\mathbf{Compute\ Features}$

if I is an image **then**

Compute the normalized LSK W_i and vectorize it to $\mathbf{f}_i$, where $i = 1, \cdots, M$.

else

Compute the normalized 3-D LSK W_i and vectorize it to $\mathbf{f}_i$, where $i = 1, \cdots, M$.

end if

$\mathbf{Stage2}$: $\mathbf{Compute\ Self\text{-}Resemblance}$

for $i = 1, \cdots, M$ **do**

if I is a grayscale image (or video) **then**

Identify feature matrices $\mathbf{F}_i$,$\mathbf{F}_j$ in a local neighborhood.

$$S_i = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1+\rho(\mathbf{F}_i,\mathbf{F}_j)}{\sigma^2}\right)}$$

else

Identify feature matrices $\mathbb{F}_i = [\mathbf{F}_i^{c1}, \mathbf{F}_i^{c3}, \mathbf{F}_i^{c3}]$ and $\mathbb{F}_j = [\mathbf{F}_j^{c1}, \mathbf{F}_j^{c3}, \mathbf{F}_j^{c3}]$ in a local neighborhood from three color channels.

$$S_i = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1+\rho(\mathbb{F}_i,\mathbb{F}_j)}{\sigma^2}\right)}$$

end if

end for

$\mathbf{Output}$: Saliency map $S_i, \quad i = 1, \cdots, M$

A. Interest region detection

1) Detecting proto-objects in images: In order to efficiently compute the saliency map, we downsample an image I to an appropriate coarse scale (64×64). We then compute LSK of size 3×3 as features and generate feature matrices $\mathbf{F}_i$ in a 5×5 local neighborhood. The number of LSK used in the feature matrix $\mathbf{F}_i$ is set to 9. For all the experiments, the smoothing parameter h for computing LSK was set to 0.008 and the fall-off parameter σ for computing self-resemblance was set to 0.07. We obtained an overall saliency map by using CIE L*a*b* color space throughout all the experiments. A typical run time takes about 1 second at scale (64×64) on an Intel Pentium 4, 2.66 GHz core 2 PC with 2 GB RAM.

From the point of view of object detection, saliency maps can explicitly represent proto-objects. We use the idea of non-parametric significance testing to detect proto-objects. Namely, we compute an empirical PDF from all the saliency values and set a threshold so as to achieve, for instance, a 95 $\%$ significance level in deciding whether the given saliency values are in the extreme (right) tails of the empirical PDF. The approach is based on the assumption that in the

Fig. 9. Some examples of proto-objects detection in face images [1].

image, a salient object is a relatively rare object and thus results in values which are in the tails of the distribution of saliency values. After making a binary object map by thresholding the saliency map, a morphological filter is applied. More specifically, we dilate the binary object map with a disk shape of size 5×5. Proto-objects are extracted from corresponding locations of the original image. Multiple objects can be extracted sequentially. Fig. 9 shows that the proposed method works well in detecting proto-objects in the images which contain a group of people in a complicated cluttered background. Fig. 10 also illustrates that our method accurately detects only salient objects in natural scenes [14].

2) Detecting actions in videos: The goal of action recognition is to classify a given action query into one of several pre-specified categories. Here, a query video may include a complex background which deteriorates recognition accuracy. In order to deal with this problem, it is necessary to have a procedure which automatically segments from the query video a small cube that only contains a valid action. Space-time saliency can provide such a mechanism. Seo and Milanfar [30] developed an automatic action cropping method by utilizing the idea of non-parametric significance testing on absolute difference images. Since their method is based on the absolute difference image, a sudden illumination change between frames can affect the performance and a choice of the anchor frame is problematic. However, the proposed space-time saliency detection method can avoid these problems. In order to compute the space-time saliency map, we only use the illumination channel because color information does not play a vital role in

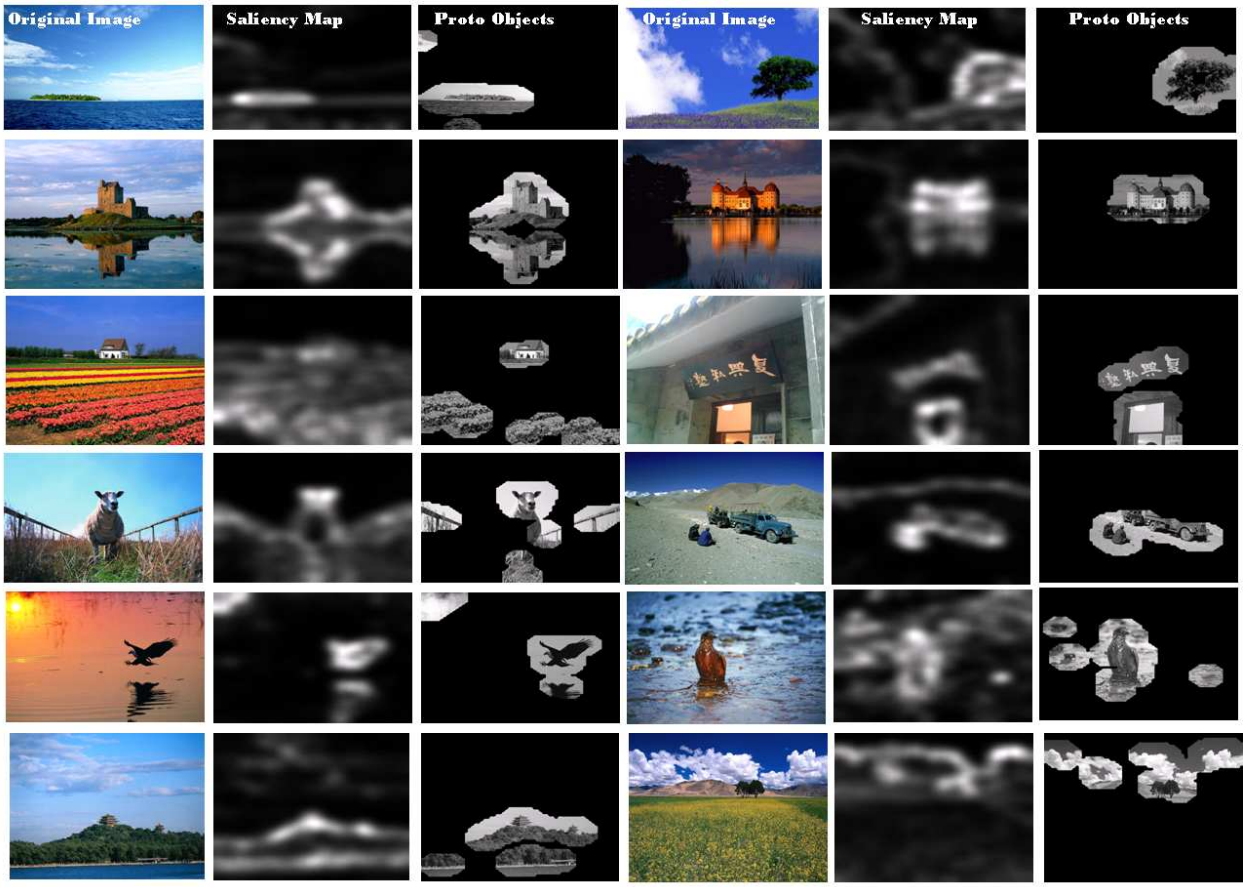
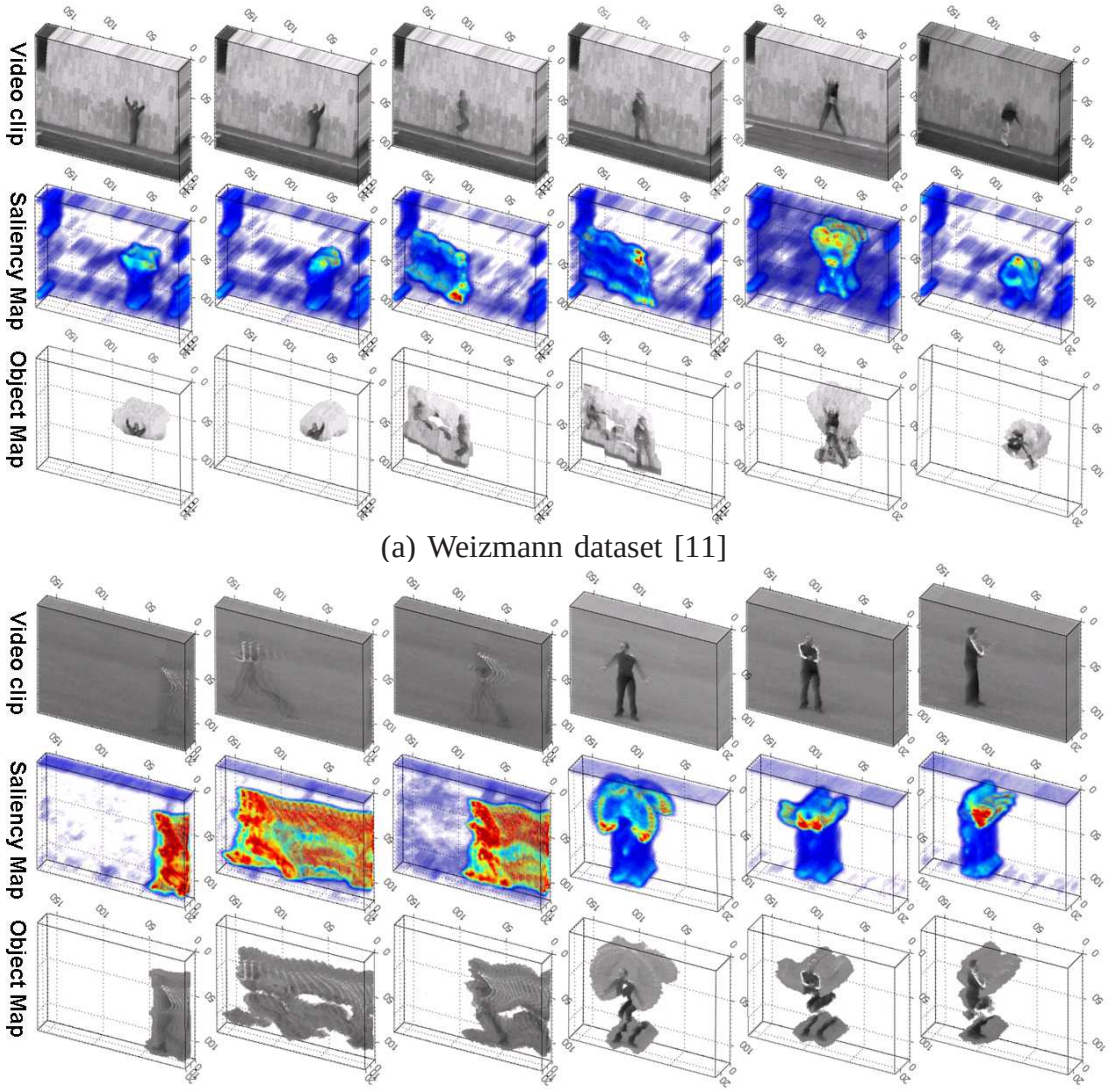


Fig. 10. Some examples of proto-objects detection in natural scene images [14]

detecting motion saliency. We downsample each frame of input video I to a coarse spatial scale (64×64) in order to reduce the time-complexity[8]. We then compute 3-D LSK of size $3 \times 3 \times 3$ as features and generate feature matrices $\mathbf{F}_i$ in a $(3 \times 3 \times 7)$ local space-time neighborhood. The number of 3-D LSK used in the feature matrix $\mathbf{F}_i$ is set to 1 for time efficiency. The procedure for detecting space-time proto-objects and the rest of parameters remain the same as in the 2-D case. A typical run of space-time saliency detection takes about 52 seconds on 50 frames of a video at spatial scale (64×64) on an Intel Pentium 4, 2.66 GHz core 2 PC with 2 GB RAM.

Fig. 11 shows that the proposed space-time saliency detection method successfully detects only salient human actions in both the Weizmann dataset [11] and the KTH dataset [28]. Our method is also robust to the presence of fast camera zoom in and out as shown in Fig. 12 where

[8]We do not downsample the video in the time domain.



(a) Weizmann dataset [11]

(b) KTH dataset [28]

Fig. 11. Some examples of detecting salient human actions in the video (a) the Weizmann dataset [11] and (b) the KTH dataset [28]

a man is performing a boxing action while a camera zoom is activated.

B. Predicting human visual fixation data

1) Static images: In this section, we used an image database and its corresponding fixation data collected by Bruce and Tsotsos [5] as a benchmark for quantitative performance analysis and comparison. This dataset contains eye fixation records from 20 subjects for a total of 120 images of size 681 $\times$ 511. The parameter settings are the same as explained in Section III-A.1. Some

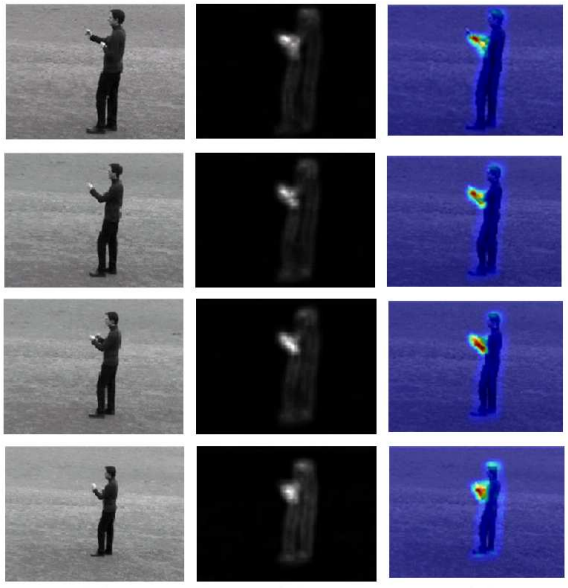


Fig. 12. Space-time saliency detection even in the presence of fast camera zoom-in. Note that a man is performing a boxing action while a camera zoom is activated

visual results of our model are compared with state-of-the-art methods in Fig. 13. As opposed to Bruce's method [5] which is quite sensitive to textured regions, and SUN [43] which is somewhat better in this respect, the proposed method is much less sensitive to background texture. To compare the methods quantitatively, we also computed the area under receiver operating characteristic (ROC) curve, and KL-divergence by following the experimental protocol of [43]. In [43], Zhang et al. pointed out that the dataset collected by Bruce [5] is center-biased and the methods by Itti et al. [17], Bruce et al. [5] and Gao et al. [8] are all corrupted by edge effects which resulted in relatively higher performance than they should have (See Fig. 14.). We compare our model against Itti et al.[9] [17], Bruce and Tsotsos[10] [5], Gao et al. [8], and SUN[11] [43]. For the evaluation of the algorithm, we used the same procedure as in [43]. More specifically, the shuffling of the saliency maps is repeated 100 times. Each time, KL-divergence is computed between the histograms of unshuffled saliency and shuffled saliency on human fixations. When calculating the area under the ROC curve, we also used 100 random permutations. The mean and

[9]Downloadable from http://ilab.usc.edu/toolkit/home.shtml

[10]Downloadable from http://web.me.com/john.tsotsos/VisualAttention/ST_and_Saliency.html

[11]Downloadable from http://www.roboticinsect.net/index.htm

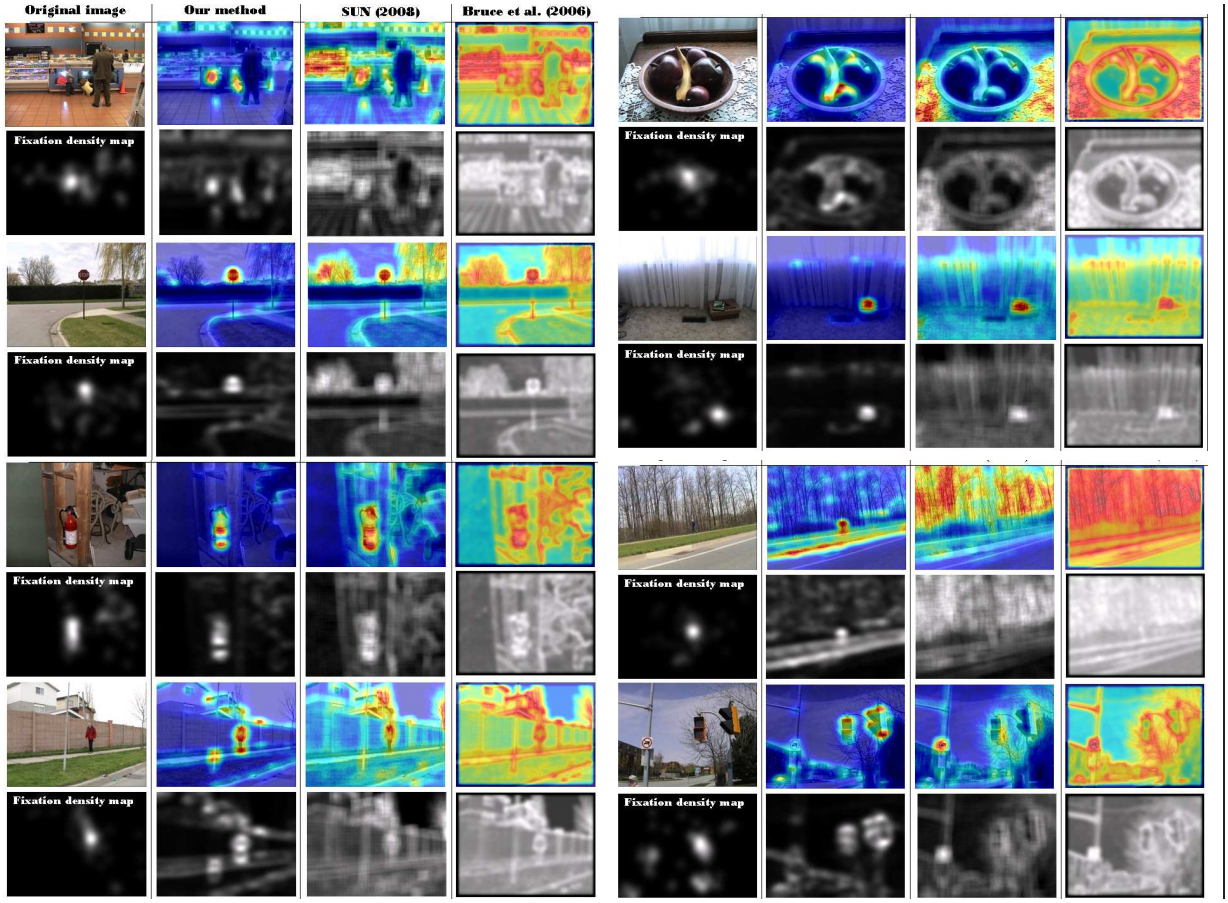


Fig. 13. Examples of saliency maps with comparison to the state-of-the-art methods. Visually, our method outperforms other state-of-the-art methods.

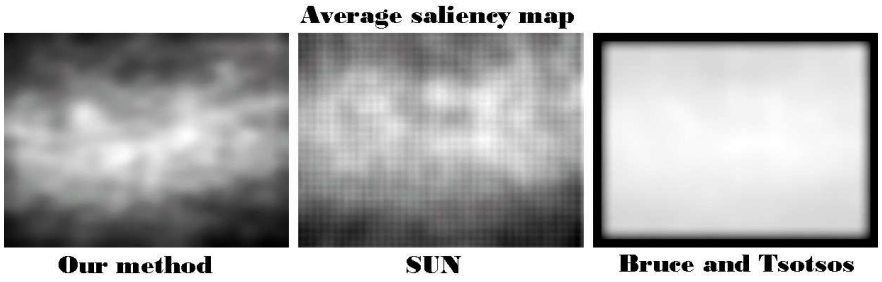


Fig. 14. Comparison of average saliency maps on human fixaton data by Bruce and Tsotsos [5]. Averages were taken across the saliency maps for a total of 120 color images. Note that Bruce et al.'s method [5] exhibits zero values at the image borders while SUN [43] and our method do not have edge effects

the standard errors are reported in Table I. Our model outperforms all the other state-of-the-art methods in terms of both KL-divergence and ROC area.

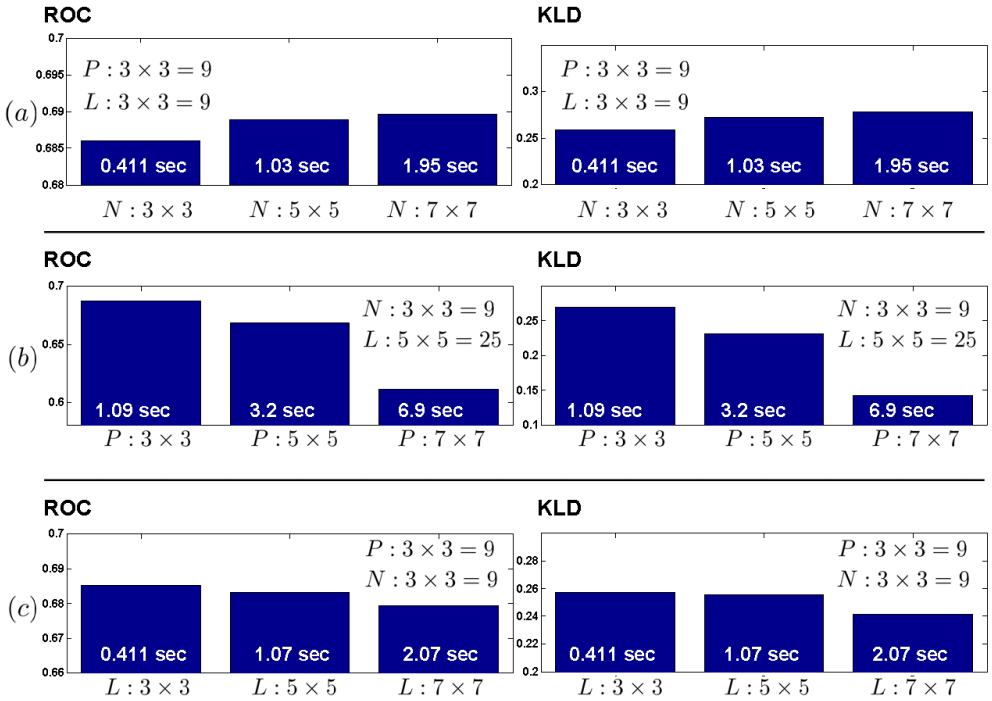


Fig. 15. Performance comparison on human fixation data by Bruce and Tsotsos [5] with respect to the choice of 1) N: size of cneter+surrouding resion for computing self-resemblance 2) P: size of LSK; and 3) L: number of LSk used in the feature matrix. Run time on one image is shown on top of each bar.

TABLE I

PERFORMANCE IN PREDICTING HUMAN EYE FIXATIONS WHEN VIEWING COLOR IMAGES. SE MEANS STANDARD ERRORS.

Model	KL (SE)	ROC (SE)
Itti *et al.* [17]	0.1130 (0.0011)	0.6146 (0.0008)
Bruce and Tsotsos [5]	0.2029 (0.0017)	0.6727 (0.0008)
Gao *et al.* [8]	0.1535 (0.0016)	0.6395 (0.0007)
Zhang *et al.* [43]	0.2097 (0.0016)	0.6570 (0.0008)
Our method	**0.2779** (0.002)	**0.6896** (0.0007)

We further examined how the performance of the proposed method is affected by the choice of parameters such as 1) N: size of center+surrouding region for computing self-resemblance 2) P: size of LSK; and 3) L: number of LSK used in the feature matrix. As shown in Fig. 15, it turns out that as we increase N, the overall performance is improved while increasing P and L rather deteriorates the performance. Overall, the best performance was achieved with the choice of $P = 3 \times 3 = 9, L = 3 \times 3 = 9$, and $N = 7 \times 7 = 49$ at the expense of increased runtime.

2) Response to Psychological Pattern: We also tested our method on psychological patterns. Psychological patterns are widely used in attention experiments not only to explore the mechanism of visual search, but also to test effectiveness of saliency maps [37], [40]. As shown in Fig. 16, whereas SUN [43] and Bruce's method [5] failed to capture perceptual differences in most cases, Gao's method [8] and Spectral Residual [14] tend to capture perceptual organization rather better. Overall, however, the proposed saliency algorithm outperforms other methods in all cases including closure pattern (Fig. 16 (a)) and texture segregation (Fig. 16 (b)) which seem to be very difficult even for humans to distinguish.

3) Dynamic scenes: In this section, we quantitatively evaluate our space-time saliency algorithm on the human fixation video data from Itti et al. [15]. This dataset consists of a total of 520 human eye-tracking data traces recorded from 8 distinct subjects watching 50 different videos (TV programs, outdoors, test stimuli, and video games: about 25 minutes of total playtime). Each video has a resolution of size 640×480. Eye movement data was collected using an ISCAN RK-464 eye-tracker. For evaluation, two hundred (four subjects $\times$ fifty video clips) eye movement traces were used (See [15] for more details.) As similarly done earlier, we computed the area under receiver operating characteristic (ROC) curve, and the KL-divergence by following the experimental protocol of [42]. We compare our model against Bayesian Surprise [17] and SUNDAy [42]. Note that human eye movement data collected by Itti et al. [15] is also center-biased and Bayesian Surprise [15] is corrupted by edge effects which resulted in relatively higher performance than it should have. For the evaluation of the algorithm, we compare the results of the proposed models from one frame to those of a randomly chosen frame from other videos. In other words, shuffling of the saliency maps is done across videos. For each video, KL-divergence is computed between the histograms of unshuffled saliency and shuffled saliency on human fixations. When calculating the area under the ROC curve, we also used the same shuffling procedure. The mean ROC area and the mean KL-divergence are reported in Table II. Our model outperforms Bayseian Surprise and SUNDAy in terms of both KL-divergence and ROC area. Some visual results of our model are shown in Fig. 17.

Our model is simple, but very fast and powerful. In terms of time complexity, a typical run

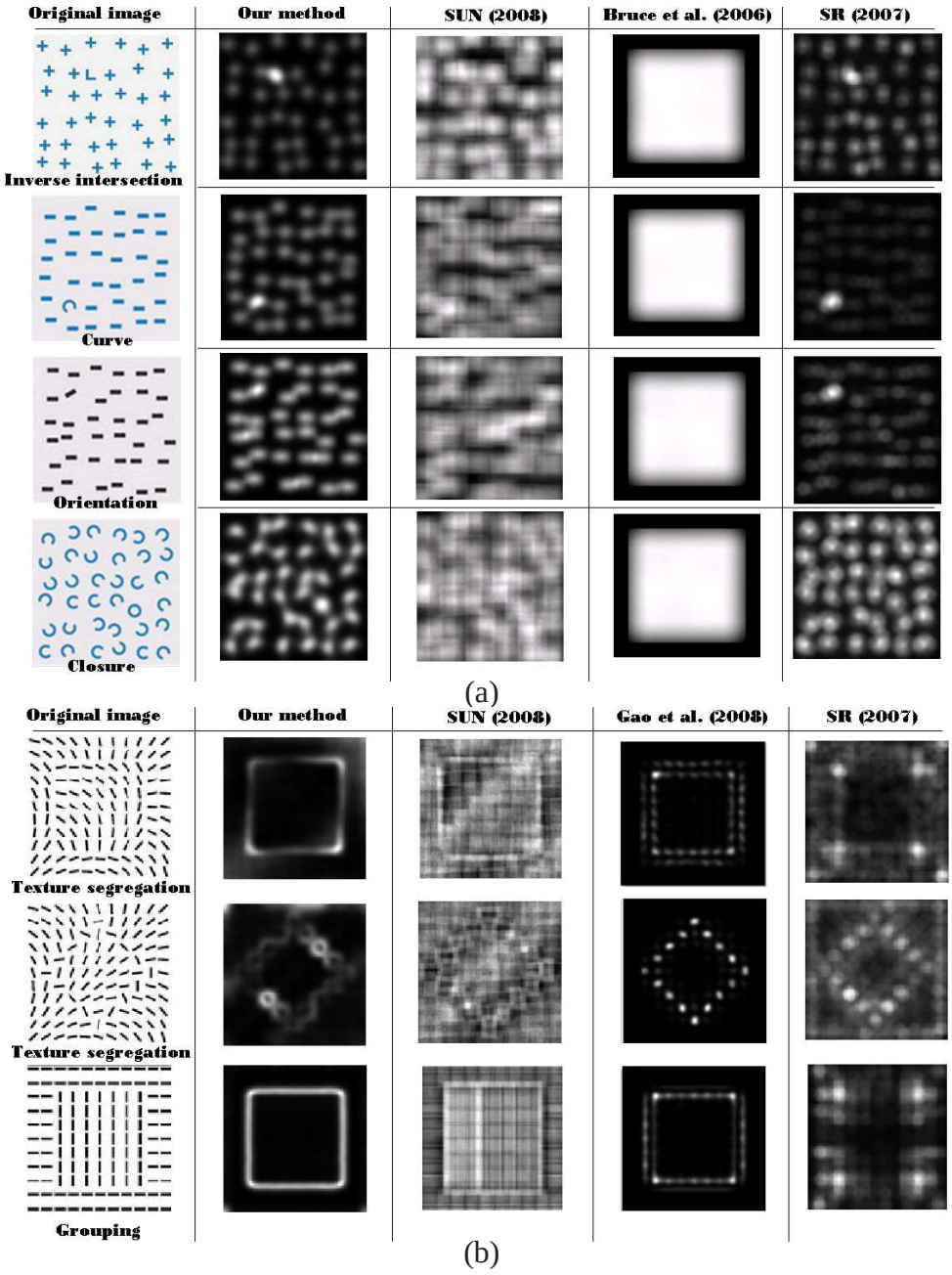


Fig. 16. Examples of Saliency map on psychological patterns. (a) images are from [14] (b) images are from [8].

time takes about 8 minutes[12] on a video of about 500 frames while Bayesian Surprise requires hours because there are 432,000 distributions that must be updated with each frame.

[12]Zhang et al. [42] reported that their method runs in Matlab on a video of about 500 frames in minutes on a Pentium 4, 3.8 GHz dual core PC with 1 GB RAM.

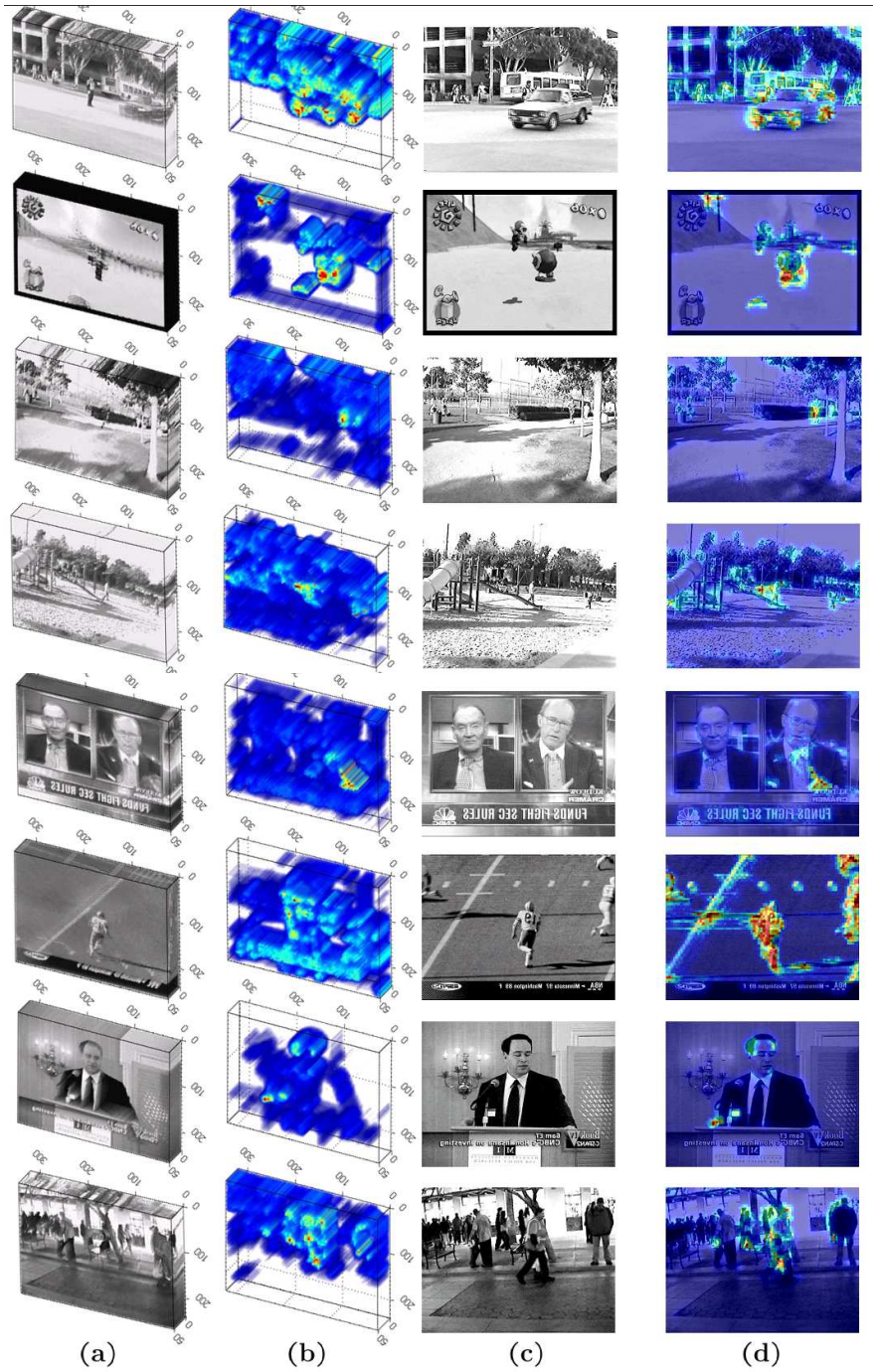


Fig. 17. (Some results on the video dataset [15] a) video clips (b) space-time saliency map (c) a frame from (a) (d) a frame superimposed with corresponding saliency map from (b)

TABLE II

PERFORMANCE IN PREDICTING HUMAN EYE FIXATIONS WHEN VIEWING VIDEOS [15].

Model	KL	ROC
Bayesian Surprise [15]	0.034	0.581
SUNDAy [42]	0.041	0.582
Our method	**0.262**	**0.589**

IV. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a unified framework for both static and space-time saliency detection algorithm by employing 2-D and 3-D *local steering kernels*; and by using a nonparametric kernel density estimation based on "Matrix Cosine Similarity" (MCS). The proposed method can automatically detect salient objects in the given image and salient moving objects in videos. The proposed method is practically appealing because it is nonparametric, fast, and robust to uncertainty in the data. Experiments on challenging sets of real-world human fixation data (both images and videos) demonstrated that the proposed saliency detection method achieves a high degree of accuracy and improves upon state-of-the-art methods. Due to its robustness to noise and other systemic perturbations, we also expect the present framework to be quite effective in other applications such as image quality assessment, background subtraction in dynamic scene, and video summarization.

ACKNOWLEDGMENT

The authors would like to thank Neil Bruce and John K. Tsotsos for kindly sharing their human fixation data; Laurent Itti for sharing his eye movement data, and Lingyun Zhang for sharing her Matlab codes and helpful discussion. This work was supported by AFOSR Grant FA 9550-07-01-0365.

REFERENCES

[1] http://www.facedetection.com/facedetection/datasets.htm.

[2] Y. Bengio, H. Larochelle, and P. Vincent. Non-local manifold parzen windows. *In Advances in Neural Information Processing Systems (NIPS)*, 18:115–122, 2005.

[3] M. Bregonzio, S. Gong, and T. Xiang. Recognizing action as clouds of space-time interest points. *To appear in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[4] T. Brox, B. Rosenhahn, and H.-P. S. D. Cremers. Nonparametric density estimation with adaptive anisotropic kernels for human motion tracking. *2nd. Workshop on Human Motion, Springer-Verlag Berlin Heidelberg (LNCS)*, 4814:152–165, 2007.

[5] N. Bruce and J. Tsotsos. Saliency based on information maximization. *In Advances in Neural Information Processing Systems*, 18:155–162, 2006.

[6] Y. Fu and T. S. Huang. Image classification using correlation tensor analysis. *IEEE Transactions on Image Processing*, 17(2):226–234, 2008.

[7] Y. Fu, S. Yan, and T. S. Huang. Correlation metric for generalized feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2229–2235, 2008.

[8] D. Gao, V. Mahadevan, and N. Vasconcelos. On the plausibility of the discriminant center-surround hypothesis for visual saliency. *Journal of Vision*, 8(7):13,1–18, 2008.

[9] D. Gao and N. Vasconcelos. Discriminant saliency for visual recognition from cluttered scenes. *In Advances in Neural Information Processing Systems*, 17:481–488, 2004.

[10] D. Gao and N. Vasconcelos. Integrated learning of saliency, complex features, and object detectors from cluttered scenes. *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:282–287, 2005.

[11] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:2247–2253, December 2007.

[12] C. Guo, Q. Ma, and L. Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[13] X. Hou and L. Zhang. Dynamic visual attention: Searching for coding length increments. *In Advances in Neural Information Processing Systems*, 21:681–688, 2008.

[14] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[15] L. Itti and P. Baldi. A principled approach to detecting surprising events in video. *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:631–637, 2005.

[16] L. Itti and P. Baldi. Bayesian surprise attracts human attention. *In Advances in Neural Information Processing Systems*, 18:1–8, 2006.

[17] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transcations on Pattern Analysis and Machine Intelligence (PAMI)*, 20:1254–1259, 1998.

[18] C. Kanan, M. Tong, L. Zhang, and G. Cottrell. SUN: Top-down saliency using natural statistics. *Accepted for Visual Cognition*, 2009.

[19] W. Kienzle, F. Wichmann, B. Scholkopf, and M. Franz. A nonparametric approach to bottom-up visual saliency. *In Advances in Neural Information Processing Systems*, 19:689–696, 2007.

[20] Q. Ma and L. Zhang. Saliency-based image quality assessment criterion. *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues (LNCS)*, 5226:1124–1133, 2008.

[21] Y. Ma, S. Lao, E. Takikawa, and M. Kawade. Discriminant analysis in correlation similarity measure space. *International Conference on Machine Learning*, 227:577–584, 2007.

[22] V. Mahadevan and N. Vasconcelos. Background subtraction in highly dynamic scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2008.

[23] S. Marat, M. Guironnet, and D. Pellerin. Video summarization using a visual attentional model. *EUSIPCO, EURASIP*, pages 1784–1788, 2007.

[24] S. Marat, T. Phuoc, L. Granjon, N. Guyader, D. Pellerin, and A. Guerin-Dugue. Modelling spatio-temporal saliency to

predict gaze direction for short videos. *International Journal of Computer Vision (IJCV)*, 82(3):231–243, 2009.

[25] O. Meur, P. L. Callet, and D. Barba. Predicting visual fixations on video based on low-level visual features. *Vision Research*, 47:2483–2498, 2007.

[26] A. Niassi, O. LeMeur, P. Lecallet, and D. barba. Does where you gaze on an image affect your perception of quality? applying visual attention to image quality metric. *IEEE International Conference on Image Processing (ICIP)*, 2:169–172, 2007.

[27] A. Oliva, A. Torralba, M. Castelhano, and J. Henderson. Top-down control of visual attention in object detection. *In Proceedings of International Conference on Image Processing*, pages 253–256, 2003.

[28] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. *IEEE Conference on Pattern Recognition (ICPR)*, 3:32–36, June 2004.

[29] H. J. Seo and P. Milanfar. Training-free, generic object detection using locally adaptive regression kernels. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 2008.

[30] H. J. Seo and P. Milanfar. Generic human action recognition from a single example. *Submitted to International Journal of Computer Vision*, March 2009.

[31] H. J. Seo and P. Milanfar. Nonparametric bottom-up saliency detection by self-resemblance. *Accepted for IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1st International Workshop on Visual Scene Understanding (ViSU09)*, Apr 2009.

[32] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. *In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.

[33] B. Silverman. *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability 26, New York: Chapman & Hall, 1986.

[34] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16(2):349–366, February 2007.

[35] H. Takeda, P. Milanfar, M. Protter, and M. Elad. Super-resolution without explicit subpixel motion estimation. *Accepted for IEEE Transactions on Image Processing*, May 2008.

[36] A. Torralba, A. Oliva, M. Castelhano, and J. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features on object search. *Psychological Review*, 113:766–786, 2006.

[37] A. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[38] K. vande Sande, T. Gevers, and C. Snoek. Evaluation of color descriptors for object and scene recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[39] P. Vincent and Y. Bengio. Manifold parzen windows. *In Advances in Neural Information Processing Systems (NIPS)*, 15:825–832, 2003.

[40] J. Wolfe. Guided search 2.0: A revised model of guided search. *Psychonomic bulletic and Rivew*, 1:202–238, 1994.

[41] B. Wu and R. Nevatia. Simultaneous object detection and segmentation by boosting local shape feature based classifier. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.

[42] L. Zhang, M. Tong, and G. Cottrell. SUNDAy: Saliency using natural statistics for dynamic analysis of scenes. *Submitted to International Conference on Computer Vision*, 2009.

[43] L. Zhang, M. Tong, T. Marks, H. Shan, and G. Cottrell. SUN: A bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7):32,1–20, 2008.