Segmenting Salient Objects from Images and Videos

Esa Rahtu[1], Juho Kannala[1], Mikko Salo[2], and Janne Heikkilä[1]

[1]Machine Vision Group, University of Oulu, Finland
[2]Department of Mathematics and Statistics, University of Helsinki, Finland

Abstract. In this paper we introduce a new salient object segmentation method, which is based on combining a saliency measure with a conditional random field (CRF) model. The proposed saliency measure is formulated using a statistical framework and local feature contrast in illumination, color, and motion information. The resulting saliency map is then used in a CRF model to define an energy minimization based segmentation approach, which aims to recover well-defined salient objects. The method is efficiently implemented by using the integral histogram approach and graph cut solvers. Compared to previous approaches the introduced method is among the few which are applicable to both still images and videos including motion cues. The experiments show that our approach outperforms the current state-of-the-art methods in both qualitative and quantitative terms.

Keywords: Saliency measure, background subtraction, segmentation

1 Introduction

Biological vision systems are remarkably effective in finding relevant targets from a scene [1]. Identifying these prominent, or salient, areas in the visual field enables one to allocate the limited perceptual resources in an efficient way. Compared to biological systems, computer vision methods are far behind in the ability of saliency detection. However, reliable saliency detection methods would be useful in many applications like adaptive compression and scaling [2, 3], unsupervised image segmentation [4, 5], and object recognition [6, 7].

Perhaps the most common approach to reduce scene clutter is to detect moving objects against a static background [8–10]. These methods have been very successful in many applications, but they have severe limitations in the case of dynamic scenes or moving cameras. These circumstances have been addressed by introducing adaptive background models and methods to eliminate camera movements [11, 12], but both of these are difficult problems and technically demanding. Moreover, the methods in this class are applicable only to video sequences, but not to still images where motion cues are not available.

A different approach is provided by supervised object detection techniques, which are aimed at finding particular categories like persons, tables, cars, etc. [13–15]. These methods have resulted in high performance, but the limitation is

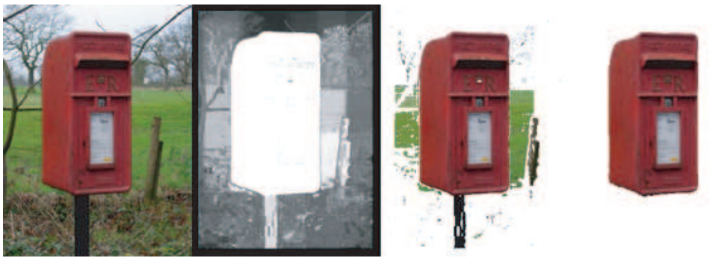

Fig. 1. Example result achieved using the proposed approach. From left to right: original image, saliency map, segmentation by thresholding, and segmentation by using the CRF model.

that the objects of interest must reside in the predefined categories from which the training samples must be available. Furthermore, the training process is rather extensive and the performance is dictated by the training data.

An alternative method is offered by general purpose saliency detectors. These methods are inspired by the ability of human visual system to quickly focus on general salient targets without preceding training. Such techniques are suitable in situations where possible targets and imaging conditions are not known in advance. Perhaps the first biologically plausible saliency detector was presented in [16], where the key idea was based on contrast measurements using difference of Gaussians filtering.

Since [16] several saliency detectors have been introduced. They are similarly focusing on estimating local feature contrast between image regions and their surroundings. Most methods implement this by local filtering or sliding window techniques [18–22]. Other methods apply Fourier transform [23, 24], mutual information formulation [25], or band-pass filtering [26].

The main limitation with many general saliency detection methods is their low resolution, e.g. 64×64 with the approaches in [23, 24] and small fraction of the image dimension with [16, 17]. An exception to this is provided by sliding window based methods [20–22] and the band-pass filtering approach [26], where the output map has the same resolution as the input image. Another drawback is that only few methods [22, 24] are capable of incorporating motion cues in the saliency map. Finally, large computational demands and variable parameters are limiting the usage of several methods [16, 18, 19, 22].

In the previous experiments the sliding window and band-pass filtering approaches have resulted in the best performance [26]. Based on this observation we present a new saliency segmentation method, which is a composition of a sliding window based saliency measure and a conditional random field (CRF) segmentation model. The introduced saliency measure is based on a rigorous statistical formulation enabling feature level information fusion and analysis of the robustness properties.

In contrast to previous methods our approach is directly applicable to both still images and videos including also motion cues in the saliency measure and the

CRF model. The method which is the most similar to our saliency measure is the approach in [21], but it differs in the formulation of saliency measure, information fusion approach, and application of motion cues in estimation. Experiments with the saliency segmentation test framework [26] show considerable improvements in terms of both precision and recall. Current state-of-the-art methods are outperformed slightly even by using a simple thresholding of the saliency map, and with a clear margin when the proposed CRF model is used.
Contributions We present a salient object segmentation method for images and video sequences. The contributions of our paper include:

1. A rigorous statistical formulation of a saliency measure, which is based on local feature contrast, and analysis of its properties under noisy data.
2. Feature level information fusion in the construction of saliency maps and inclusion of motion cues by using optical flow.
3. CRF model for segmenting objects in images and videos based on information in saliency maps.

2 Saliency Measure

In this section we describe the proposed saliency measure. The measure is based on applying a sliding window to the image, and on comparing in each window the contrast between the distribution of certain features in an inner window to the distribution in the collar of the window. The basic setup for this saliency measure was introduced in [21], but here we will modify it by taking into account the properties 1 and 2 listed in the contributions above.

2.1 Definition of saliency measure

Consider an image in $\mathbb{R}^2$ and a map F which maps every point x to a certain feature $F(x)$ (which could be the intensity, the value in different color channels, or information obtained from motion). The feature space is divided into disjoint bins, with $Q_{F(x)}$ denoting the bin which contains $F(x)$.

We consider a rectangular window W divided into two disjoint parts, a rectangular inner window K (the kernel) and the border B (see Figure 2), and apply the hypothesis that points in K are salient and points in B are part of the background. A similar hypothesis has also been used in [21, 22]. Let Z be a random variable with values in W, describing the distribution of pixels in W. Under the stated hypothesis, the saliency measure of a point $x \in K$ is defined to be the conditional probability

$$S_0(x) = P(Z \in K | F(Z) \in Q_{F(x)}). \tag{1}$$

The saliency measure of x is always a number between 0 and 1. It follows from the definition that a pixel x is salient (that is, $S_0(x)$ is close to 1) if the feature at x is similar to the features at points of the inner window (and different from points in the border).

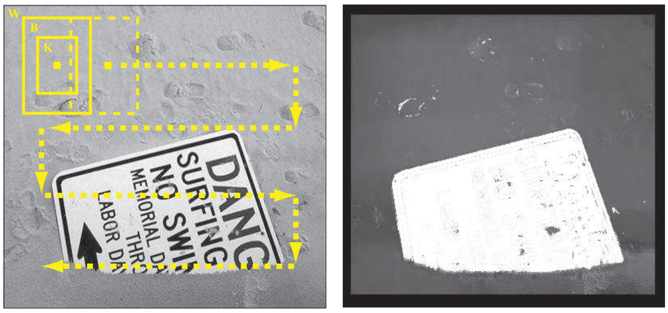

Fig. 2. Illustration of saliency map computation.

The computation of $S_0(x)$ can be achieved through the Bayes formula $P(A|B) = P(B|A)P(A)/P(B)$. Using the abbreviations H_0, H_1, and $F(x)$ for the events $Z \in K$, $Z \in B$, and $F(Z) \in Q_{F(x)}$, respectively, gives that

$$S_0(x) = \frac{P(F(x)|H_0)P(H_0)}{P(F(x)|H_0)P(H_0) + P(F(x)|H_1)P(H_1)}. \tag{2}$$

The computation of this measure is greatly simplified if we assume that Z has a probability density function p which is constant on K and on B. In fact, given p_0 with $0 < p_0 < 1$, we take $p(x) = p_0/|K|$ for $x \in K$ and $p(x) = (1 - p_0)/|B|$ for $x \in B$. With this choice, the conditional probabilities in the last expression for $S_0(x)$ become normalized histograms. For instance, for the set K we write

$$h_K(x) = P(F(x)|H_0) = \frac{1}{P(H_0)} \int_{K \cap F^{-1}(Q_{F(x)})} p(w)\, dw. \tag{3}$$

Since p is constant on K, the discretized version of the last quantity is obtained by just counting the number of points z in K for which $F(z)$ is in $Q_{F(x)}$, and by dividing by the number of points in K. Defining similarly $h_B(x) = P(F(x)|H_1)$, the saliency measure may be written as

$$S_0(x) = \frac{h_K(x)p_0}{h_K(x)p_0 + h_B(x)(1 - p_0)}. \tag{4}$$

Clearly $S_0(x)$ is always a number between 0 and 1.

2.2 Regularized saliency measure

Note that a small change in the function F may change the bin of $F(x)$, possibly resulting in a large change in the value of $h_K(x)$. Therefore the measure $S_0(x)$ is not stable with respect to noise. To increase robustness we introduce a regularized saliency measure. For computational purposes it is most convenient to regularize the normalized histograms directly.

Assume that the bins in feature space are indexed by integers j, and let $j(x)$ be such that $F(x)$ lies in the bin $Q_{j(x)}$. Let also $h_A(j) = h_A(x)$ for $j = j(x)$. If $\alpha > 0$ let $g_\alpha(x) = c_\alpha e^{-\frac{x^2}{2\alpha}}$ be the Gaussian function with variance α, normalized so that $\sum_{j=-\infty}^{\infty} g_\alpha(j) = 1$. Define the regularized histogram $h_{K,\alpha}(j) = \sum_{j=-\infty}^{\infty} g_\alpha(j-k)h_K(k)$. With a similar definition for $h_{B,\alpha}$, the regularized saliency measure is defined by

$$S_\alpha(x) = \frac{h_{K,\alpha}(j(x))p_0}{h_{K,\alpha}(j(x))p_0 + h_{B,\alpha}(j(x))(1-p_0)}. \tag{5}$$

It can be shown that for $\alpha > 0$ and under certain assumptions, the continuous analog of the measure $S_\alpha(x)$ is stable with respect to small changes in the function F (more details in the on-line Appendix[1]). This indicates that the regularized measure is indeed quite robust. Another benefit of the regularization is that by suitable choices for α it is possible to emphasize and de-emphasize different features that are used for the function F. Having a larger α for a certain feature will decrease the weight of that feature in the resulting saliency map.

2.3 Implementation

For the feature function F we will use the CIELab color values of an image, and also motion information in the case of video sequences. For still images, if $L(x)$, $a(x)$, and $b(x)$ are the CIELab values at a point x, the feature map is $F(x) = (L(x), a(x), b(x))$. In the case of frames in a video sequence we combine the CIELab information for each frame with the magnitude of the optical flow $Y(x)$. The feature map is then $F(x) = (L(x), a(x), b(x), Y(x))$. All the values are quantized, and the bins are the elements in the finite feature space.

To simplify the computations, we make the assumption that the random variables $L(Z)$, $a(Z)$, $b(Z)$, $Y(Z)$ are independent in any subwindow. This is reasonable since the CIELab color space is constructed so that the intensity value L is independent of the a and b coordinates, and also since in our experiments using a joint distribution for a and b did not yield improved results compared to the case where independence was assumed. It is also fair to assume that the optical flow $Y(Z)$ is independent of $L(Z)$, $a(Z)$, $b(Z)$.

Using the independence, we have $P(F(x)|H_0) = \boldsymbol{h}_K(x)$ where $\boldsymbol{h}_K(x)$ is the product of normalized histograms $h_K^t(t(x))$ (here t is one of L, a, b, Y) and $h_K^t(t_0)$ is equal to the number of points z in K such that $t(z) = t_0$ divided by the number of points in K, etc. We define regularized histograms

$$h_{K,\alpha}^t(j) = \mathcal{N}(\sum_k g_\alpha(j-k)h_K^t(k)), \qquad t \text{ is one of } L, a, b, Y. \tag{6}$$

Here $\mathcal{N}(f(j)) = \frac{1}{\sum_k f(k)} f(j)$ is the normalization operator. The final saliency measure is given by

$$S_\alpha(x) = \frac{\boldsymbol{h}_{K,\alpha}(x)p_0}{\boldsymbol{h}_{K,\alpha}(x)p_0 + \boldsymbol{h}_{B,\alpha}(x)(1-p_0)}. \tag{7}$$

[1] http://www.ee.oulu.fi/mvg/page/saliency

Here $\boldsymbol{h}_{K,\alpha}(x)$ is equal to $h^L_{K,\alpha}(L(x))h^a_{K,\alpha}(a(x))h^b_{K,\alpha}(b(x))$ for still images and to $h^L_{K,\alpha}(L(x))h^a_{K,\alpha}(a(x))h^b_{K,\alpha}(b(x))h^Y_{K,\alpha}(Y(x))$ for frames in a video sequence, and $\boldsymbol{h}_{B,\alpha}(x)$ is defined in a similar manner.

The saliency map for the entire image is achieved by sliding the window W with different scales over the image, constructing the proposed feature histograms for each window, smoothing the histograms, and then computing the measure for each pixel in K at each window position and scale. The final saliency value is then taken as the maximum over all windows containing a particular pixel. Figure 2 shows an illustration of the process.

In practice it is enough to evaluate the measure only in a small subset of all possible window positions and scales. In our experiments we used a regular grid with step size equal to 1 percent of the largest image dimension. We applied four scales with row and column sizes equal to $\{25, 10; 30, 30; 50, 50; 70, 40\}$ percents of the largest image dimension, respectively. An illustrative Matlab implementation of our measure is available on-line[2].

3 Salient Object Segmentation

In this section, we propose a bilayer segmentation method that estimates the salient and non-salient pixels of an image or a video by minimizing an energy function, which is derived from a conditional random field model that incorporates the pixelwise saliency measure of the previous section. The motivation for using a CRF model is the fact that usually the goal of saliency detection is to achieve an object-level segmentation rather than pixel-level segmentation. That is, the user is more interested in objects which contain salient pixels than the salient pixels themselves. Therefore, instead of considering pixels independently and segmenting the saliency maps by simple thresholding, it is reasonable to formulate the binary labeling problem in terms of a CRF based energy function, whose exact global minimum can be computed via graph cuts [27, 28]. In the following, we describe the energy functions used in our experiments. The formulations are inspired by several previous works which apply graph cuts for binary segmentation problems, e.g. [29, 30].

3.1 Segmentation Energy for Still Images

First, given an image with N pixels, we use the saliency measure S_α to compute a saliency map $s = (s_1, \ldots, s_N)$, which is an array of saliency values. Further, we represent the image as an array $c = (c_1, \ldots, c_N)$, where each $c_n = (L_n, a_n, b_n)$ is a Lab color vector for a single pixel. Our task is to find a binary labeling $\sigma = (\sigma_1, \ldots, \sigma_N)$ so that $\sigma_n \in \{0, 1\}$ indicates whether the pixel n belongs to a salient object or not.

[2] `http://www.ee.oulu.fi/mvg/page/saliency`

The optimal labeling is computed by minimizing the energy function

$$E_I(\sigma, c, s) = \sum_{n=1}^{N} \big(w_S U^S(\sigma_n, s_n) + w_C U^C(\sigma_n, c_n) \big) + \sum_{(n,m)\in\mathcal{E}} V(\sigma_n, \sigma_m, c_n, c_m), \tag{8}$$

which consists of two unary terms, U^S and U^C, and a pairwise term V, which depends on the labels of neighboring pixels.[3] The weight factors w_S and w_C are scalar parameters. The purpose of U^S is to penalize labelings which assign pixels with low s_n to the salient layer, whereas U^C encourages such labelings where the salient layer includes pixels which have similar colors as pixels for which s_n is high. The pairwise term V favors spatial continuity of labels. Overall, the energy function (8) has the standard form [28], which is used in many segmentation approaches [29] and can be statistically justified by using the well-known CRF formulation [30]. The precise definitions for U^S, U^C, and V are described below.

The unary saliency term U^S is defined by

$$U^S(\sigma_n, s_n) = \delta_{\sigma_n,1}(1 - f(s_n)) + \delta_{\sigma_n,0} f(s_n), \tag{9}$$

where $\delta_{\cdot,\cdot}$ is the Kronecker delta and f is defined by either

$$f(s_n) = \max(0, \operatorname{sign}(s_n - \tau)) \qquad \text{or} \qquad f(s_n) = (s_n)^{\kappa}\,. \tag{10}$$

In probabilistic terms, one may think that U^S is an approximation to

$$-\log P(\mathcal{S}_n = s_n | \sigma_n) = -\delta_{\sigma_n,1} \log p_1(s_n) - \delta_{\sigma_n,0} \log p_0(s_n), \tag{11}$$

where p_1 and p_0 are the conditional density functions of s_n given that pixel n is salient or non-salient, respectively. Hence, loosely speaking, the ratio $f(s_n) : (1 - f(s_n))$ can be seen as a one-parameter model for the ratio of negative log-likelihoods, $(-\log p_0(s_n)) : (-\log p_1(s_n))$.

The unary color term U^C is defined by

$$U^C(\sigma_n, c_n) = -\log P(\mathcal{C}_n = c_n | \sigma_n) = -\delta_{\sigma_n,1} \log p_1^c(c_n) - \delta_{\sigma_n,0} \log p_0^c(c_n), \tag{12}$$

where the conditional density functions p_1^c and p_0^c are the color distributions of salient and non-salient pixels, respectively. Given image c, we compute p_1^c and p_0^c as a product of two histograms, that is, $p_1^c(c_n) = h_1^L(L_n) h_1^{ab}(a_n, b_n)$ and $p_0^c(c_n) = h_0^L(L_n) h_0^{ab}(a_n, b_n)$. The histograms h_1^L and h_0^L are computed as weighted histograms of pixels' intensity values, where the weights for pixel n are $f(s_n)$ and $(1 - f(s_n))$, respectively. The color histograms h_1^{ab} and h_0^{ab} are computed in a similar manner using $f(s_n)$ and $(1 - f(s_n))$ as weights.

The pairwise prior V is

$$V(\sigma_n, \sigma_m, c_n, c_m) = \gamma \delta_{\sigma_n,\sigma_m} e^{-||c_n - c_m||^2_{\Lambda}} + \eta \delta_{\sigma_n,\sigma_m}, \tag{13}$$

where γ and η are scalar parameters and $||\cdot||_{\Lambda}$ is a Mahalanobis distance with diagonal matrix Λ. Both terms in (13) penalize neighboring pairs of pixels that

[3] Set $\mathcal{E}$ contains pairs (n, m) for which $n < m$ and pixels n and m are 4-connected.

have different labels. However, the first term adds lower cost for such segmentation boundaries that co-occur with contours of high image contrast [30].

Given image c and saliency map s, we estimate σ by minimizing (8) via graph cuts. The pixels labeled by 1 belong to salient objects and the rest is background. Further, given test images with ground truth saliency maps where the pixels of salient objects are manually labeled, we compare the two choices for f in (10) by computing the corresponding ROC curves. That is, by changing the value of parameter τ (or κ) from 0 to ∞ the labeling gradually changes from one map to zero map, and we may draw a ROC curve by counting the number of correctly and incorrectly labeled pixels at each parameter setting. Section 4 reports the results obtained with a publicly available dataset of 1000 images. For the experiments, we determined the values of w_S, w_C, γ, and η by the approach in [31]. All other parameters except τ and κ were set to manually predefined values and kept constant during the experiments.

3.2 Segmentation Energy for Videos

Our CRF segmentation model for videos incorporates motion information indirectly via the saliency measure S_α, as described in Section 2, but also directly via an additional unary term, which is introduced below. In detail, the energy function for videos is an augmented version of (8), i.e.

$$E_V(\sigma^t, \sigma^{t-1}, \sigma^{t-2}, c^t, c^{t-1}, s) = E_I(\sigma^t, c^t, s) + \sum_{n=1}^{N} U^T(\sigma_n^t, \sigma_n^{t-1}, \sigma_n^{t-2}, c_n^t, c_n^{t-1}) \tag{14}$$

where σ^t is the segmentation of the current frame, σ^{t-1} and σ^{t-2} are the segmentations of the two previous frames, c^t is the current frame, c^{t-1} is the previous frame, and U^T is an additional unary term which improves temporal coherence.

The term U^T has the following form,

$$U^T(\sigma_n^t, \sigma_n^{t-1}, \sigma_n^{t-2}, c_n^t, c_n^{t-1}) = \mu\, \delta_{\sigma_n^t, \sigma_n^{t-1}}\, e^{-||c_n^t - c_n^{t-1}||_\Gamma^2} - \nu \log p_T(\sigma_n^t | \sigma_n^{t-1}, \sigma_n^{t-2}), \tag{15}$$

where μ and ν are scalar parameters, $||\cdot||_\Gamma$ is a Mahalanobis distance with diagonal matrix Γ, and p_T is the prior probability density function of σ_n^t conditioned on σ_n^{t-1} and σ_n^{t-2}. Thus, since $p_T(\sigma_n^t = 0 | \sigma_n^{t-1}, \sigma_n^{t-2}) = 1 - p_T(\sigma_n^t = 1 | \sigma_n^{t-1}, \sigma_n^{t-2})$, p_T is defined by four parameters which determine $p_T(\sigma_n^t = 1 | \sigma_n^{t-1}, \sigma_n^{t-2})$ corresponding to the following four cases: $(\sigma_n^{t-1}, \sigma_n^{t-2}) = \{(0,0), (0,1), (1,0), (1,1)\}$. The first term in (15) is an additional data-dependent cost for pixels which change their label between frames $(t-1)$ and t. This extra cost is smaller for those pixels whose color changes a lot between the frames.

Given a video sequence, we compute the segmentation σ^t for frames $t > 2$ by minimizing (14) via graph cuts. For grayscale videos we use the grayscale version of (14). In the experiments, the values of the common parameters were the same for the grayscale and color versions. Further, we used the first choice for f in (10) and all parameter values were kept constant in the experiments.

4 Experiments

In this section, we assess the proposed approach in saliency segmentation experiments. The performance is compared with the state-of-the-art methods using the programs given by the authors [10, 23, 21, 20, 26] or our own implementation with default parameters [24]. The experiments are divided into two parts, where the first one considers still images and the second one video sequences.

4.1 Segmenting salient objects from images

First, we run the publicly available saliency segmentation test, introduced in [26]. The proposed method is compared to the band-pass approach in [26], which was reported to achieve clearly the best performance among the several tested methods [26] (note the erratum[4]). In addition we also include the approaches from [21] and [24], since they were not evaluated in [26].

The experiment contains 1000 color images with pixel-wise ground truth segmentations provided by human observers. First a saliency map is computed for each test image and then a segmentation is generated by simply thresholding the map by assigning the pixels above the given threshold as salient (white foreground) and below the threshold as non-salient (black background). A precision and recall rate is then computed using definitions:

$$precision = |SF \cap GF|/|SF|, \quad recall = |SF \cap GF|/|GF|, \tag{16}$$

where SF denotes the segmented foreground pixels, GF denotes the ground truth foreground pixels, and $|\cdot|$ refers to number of elements in a set. By sliding the threshold from minimum to maximum saliency value, we achieved the precision-recall curves illustrated in Figure 3 (magenta, cyan, orange, and green).

The results show that the proposed saliency measure achieves the highest performance up to a recall rate 0.9. Furthermore also the method from [21] seems to outperform the state-of-the-art results in [26]. Notice that the precision-recall curves of the proposed method and the method in [21] do not have values for small recalls because several pixels reach the maximum saliency value and they change labels simultaneously when the threshold is lowered below one. At maximum recall all methods converge to 0.2 precision, which corresponds to a situation where all pixels are labeled as foreground.

We continue the experiment by adding the CRF segmentation model from Section 3 on top of our saliency measure. First, we perform the same experiment as above, but refine the thresholded saliency maps using the CRF model (i.e. the first choice is used for f in (10)). The resulting precision-recall-curve in Figure 3 (blue) illustrates a clear gain compared to thresholded saliency map in both precision and recall. Finally, we replace the thresholded saliency maps in the CRF by the soft assignment approach of Section 3 (i.e. the second choice for f in (10)). Now, instead of sliding threshold τ we change the exponent κ, and

[4] `http://ivrg.epfl.ch/supplementary_material/RK_CVPR09/index.html`

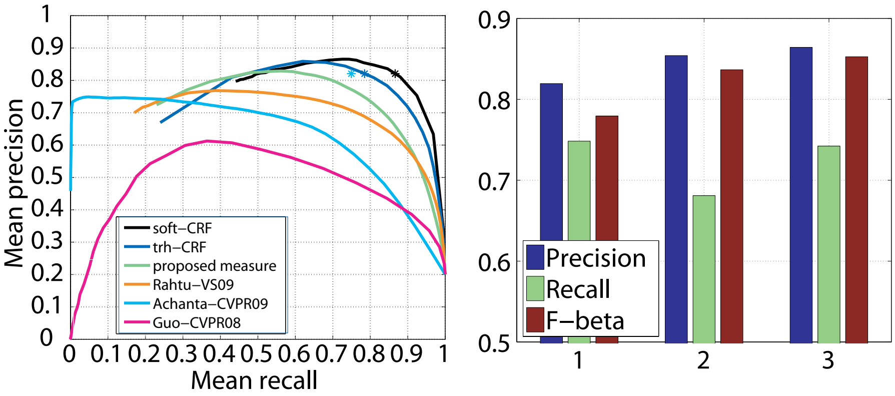

Fig. 3. Left: Mean precision-recall curves using comparison methods and the proposed approach. Right: Mean precision, recall, and F-measure values for comparison method [26] (1), our method with thresholding (2), and our method with soft assignments (3). Notice that $\beta = 0.3$ (used according to [26]) strongly emphasizes precision.

achieve the corresponding precision-recall-curve in Figure 3 (black), which shows further improvement in performance.

In [26] the best results were achieved by combining the band-pass saliency map with adaptive thresholding and the mean-shift segmentation algorithm. The achieved precision, recall, and F-measure values were 0.82, 0.75, and 0.78, respectively. The F-measure was computed from precision and recall by $F_\beta = (1 + \beta^2)\,(precision \cdot recall) \,/\, \left(\beta^2 \cdot precision + recall\right)$, where $\beta = 0.3$ was used [26]. This corresponds to a point marked using by a cyan star in Figure 3. This result remains lower than our results with both naïve thresholding and soft mapping with CRF, which provide the same precision with recalls 0.79 and 0.87, respectively. These points are also marked in Figure 3 with correspondingly colored stars. The maximum F-measure value we achieve is 0.85, which represents 9 percent improvement over [26]. The comparison of F-measures is shown in Figure 3. A few results of the proposed saliency segmentation method are shown in Figure 4 for subjective evaluation.

4.2 Segmenting salient objects from video sequences

Another set of experiments was performed using videos. The saliency maps were computed as described in Section 2 by using both the CIELab color values (only L in the case of gray-scale videos) and the magnitude of optical flow as features. The optical flow was computed using a publicly available[5] implementation [32], which can provide real-time performance. The final salient segments were computed using either direct thresholding or the CRF method of Section 3.

[5] http://gpu4vision.org/

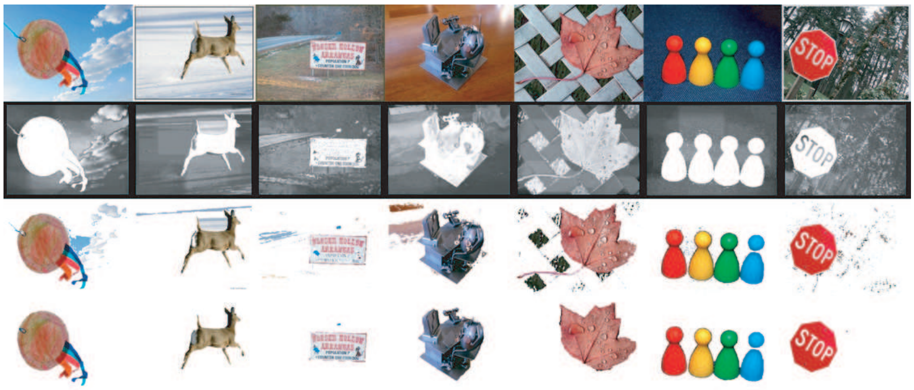

Fig. 4. Examples of saliency maps and segmentations. Top row shows the original image, second row shows the saliency maps, third row shows the segmentations using threshold 0.7, and bottom row shows the segmentations using the CRF model.

The results are compared with methods in [24, 21, 20, 10] from which the last mentioned is a general background subtraction method. All comparison methods used default parameters given by the authors. Further, in order to achieve best possible performance with comparison methods, we also included all the postprocessing techniques presented in the original papers. As test videos, we used the publicly available image sequences originally used in [21] and [22]. The two sequences from [21] illustrate moving and stationary objects in the case of a fixed and a mobile camera. Sequences from [22] show highly dynamic backgrounds with targets of various size. The original results of [22] are available on-line and are directly comparable to our results. Their experiments also include several traditional background subtraction approaches.

Figure 5 illustrates characteristic frames from tested sequences. The results include original frames, saliency maps, and final segmentations. Full videos are also available on-line[6]. The results illustrate the problems of traditional background subtraction methods, which work well with stationary cameras and constantly moving objects. However serious problems appear if the camera is moving and targets may stop every once in a while. The poor resolution of [24] is visible in the inaccurate segmentations and several missed objects. The method in [21] works better, but the result is rather noisy and the segmentations are not accurate. The missing motion information is also visible in the results with [21].

The proposed approach achieves the most stable results, where also the effect of motion cues is clearly visible. The returned segments mostly correspond to natural objects. Sometimes the method may return salient segments which a human observer would classify as part of the background (e.g. grass between the roads). However, like with all saliency detection methods this is difficult to avoid if these objects are distinct from the background in terms of visual contrast.

[6] http://www.ee.oulu.fi/mvg/page/saliency

5 Conclusions

In this paper, we presented a new combination of a saliency measure and a CRF based segmentation model. The measure was formulated using a probabilistic framework, where different features were fused together in joint distributions. The sensitivity of the proposed measure was shown to be controlled by a smoothing parameter, which can also be used to set the relative weights of the features.

The resulting saliency map was turned into a segmentation of natural and well-defined objects using the CRF model. The segmentations were constantly improved and stabilized especially in the case of video sequences, where the smoothness over frames was emphasized by the applied model. In addition we proposed a technique to include optical flow motion cues into the saliency estimation, which greatly improved the recall rate with videos.

The experiments with a publicly available dataset showed that our approach yields clearly higher performance than the state-of-the-art in terms of both recall and precision. The new method produces both more dicriminative saliency maps and more accurate segmentations. The precision was improved especially at high recalls, where previous results were rather poor. The experiments with video sequences showed also consistent improvement over the tested methods.

The features used in our approach included Lab color values and optical flow, which are both obtainable in real-time. The saliency measure itself was evaluated using sliding windows and integral histograms. The processing takes about 8 seconds per image with our current Matlab implementation, but we believe that this can be reduced to close to real time. The CRF energy minimization by graph cuts took 1/20 seconds per image. In future, we aim to achieve a real time implementation by using total-variation techniques instead of graph cuts.

Acknowledgments. The work has been supported by the Academy of Finland.

References

1. Yarbus, A.: Eye movements and vision. Plenum, New York, US (1967)
2. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. ACM Transactions of Graphics **26** (2007)
3. Marchesotti, L., Cifarelli, C., Csurka, G.: A framework for visual saliency detection with applications to image thumbnailing. In: ICCV. (2009)
4. Han, J., Ngan, K.N., Li, M., Zhang, H.: Unsupervised extraction of visual attention objects in color images. IEEE Trans. Circuits Syst. Video Techn. **16** (2006) 141–145
5. Ko, B., Nam, J.: Object-of-interest image segmentation based on human attention and semantic region clustering. J. Opt. Soc. Am. **23** (2006) 2462–2470
6. Rutishauser, U., Walther, D., Koch, C., Perona, P.: Is bottom-up attention useful for object recognition. In: CVPR. (2004)
7. Yang, L., Zheng, N., Yang, J., Chen, M., Cheng, H.: A biased sampling strategy for object categorization. In: ICCV. (2009)

8. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. IEEE TPAMI **27** (2005) 1778–1792
9. Monnet, A., Mittal, A., Paragios, N., Ramesh, V.: Background modeling and subtraction of dynamic scenes. In: CVPR. Volume 2. (2003) 1305–1312
10. Heikkilä, M., Pietikäinen, M.: A texture-based method for modeling the background and detecting moving objects. IEEE TPAMI **28** (2006) 657–662
11. Ren, Y., Chua, C., Ho, Y.: Motion detection with nonstationary background. Machine Vision and Applications **13** (2003) 332–343
12. Sheikh, Y., Javed, O., Kanade, T.: Background Subtraction for Freely Moving Cameras. In: ICCV. (2009)
13. Lampert, C., Blaschko, M., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. IEEE TPAMI **31** (2009) 2129 –2142
14. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV. (2009)
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. Volume 1. (2005)
16. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE TPAMI **20** (1998) 1254–1259
17. Itti, L., Koch, C.: A saliency-based search mechanism for overt and covert shifts of visual attention. Vision Research **40** (2000) 1489–1506
18. Ma, Y., Zhang, H.: Contrast-based image attention analysis by using fuzzy growing. In: ACM Intl. Conf. on Multimedia. (2003) 59–68
19. Hu, Y., Xie, X., Ma, W., Chia, L., Rajan, D.: Salient region detection using weighted feature maps based on the human visual attention model. In: Advances in Multimedia Information Processing. (2004) 993–1000
20. Achanta, R., Estrada, F.J., Wils, P., Süsstrunk, S.: Salient region detection and segmentation. In: ICVS. (2008) 66–75
21. Rahtu, E., Heikkilä, J.: A simple and efficient saliency detector for background subtraction. In: IEEE Intl. Workshop on Visual Surveillance. (2009) 1137–1144
22. Mahadevan, V., Vasconcelos, N.: Spatiotemporal saliency in highly dynamic scenes. IEEE TPAMI **32** (2010) 171–177
23. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: CVPR. (2007)
24. Guo, C., Ma, Q., Zhang, L.: Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: CVPR. (2008)
25. Gao, D., Vasconcelos, N.: Bottom-up saliency is a discriminant process. In: ICCV. (2007)
26. Achanta, R., Hemami, S.S., Estrada, F.J., Süsstrunk, S.: Frequency-tuned salient region detection. In: CVPR. (2009)
27. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE TPAMI **23** (2001) 1222–1239
28. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE TPAMI **26** (2004) 147–159
29. Rother, C., Kolmogorov, V., Blake, A.: ”Grabcut”: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23** (2004) 309–314
30. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: CVPR. Volume 1. (2006) 53–60
31. Szummer, M., Kohli, P., Hoiem, D.: Learning CRFs using graph cuts. In: ECCV. (2008)
32. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: BMVC. (2009)

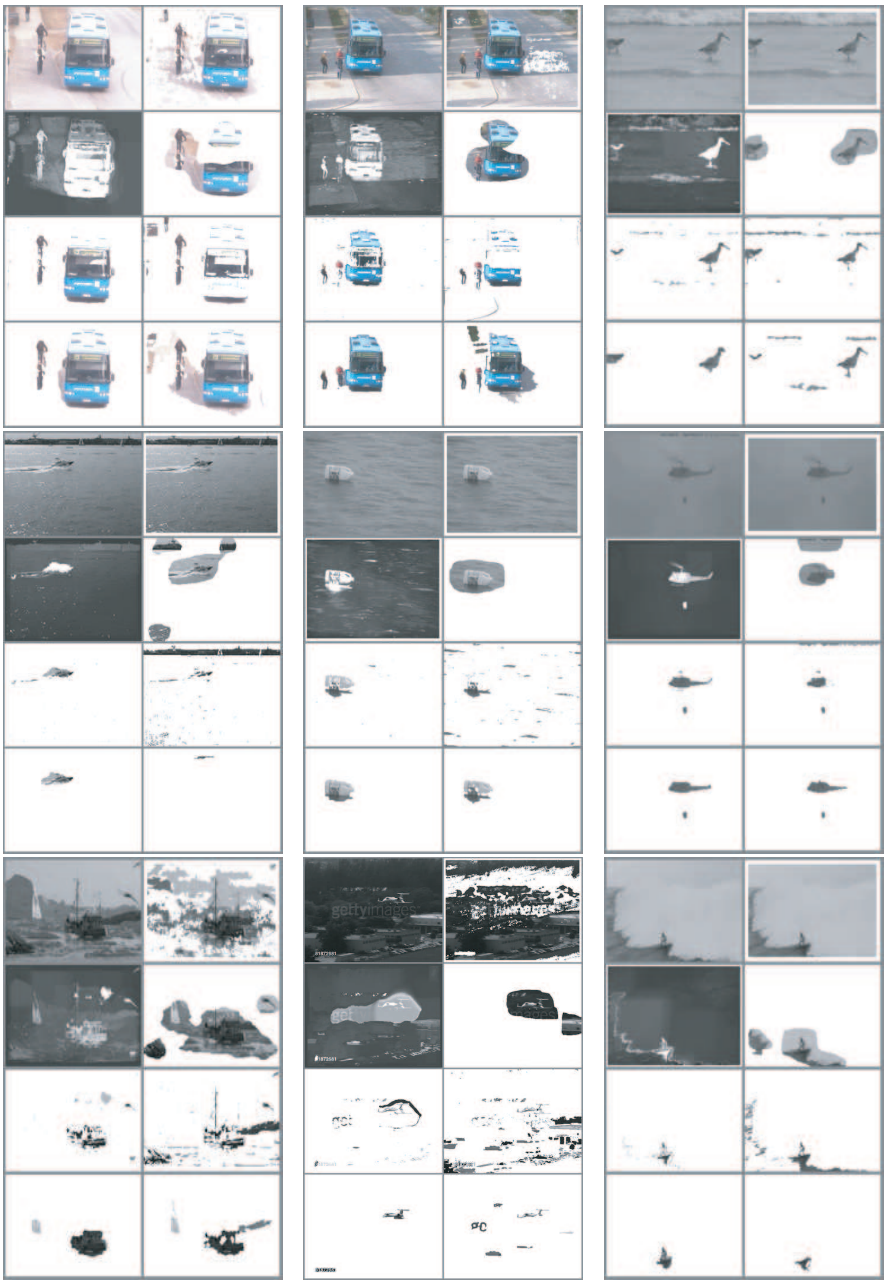

Fig. 5. Results of saliency detection from videos. Each group of eight images correspond to one test sequence and they are organized as follows: Left column from top to bottom consists of the original frame, the proposed saliency map, segmentation by thresholding proposed saliency map, and segmentation using the proposed CRF model. Right column from top to bottom consists of segmentations using the saliency maps and full post processing of comparison methods [10], [24], [20], and [21], respectively.