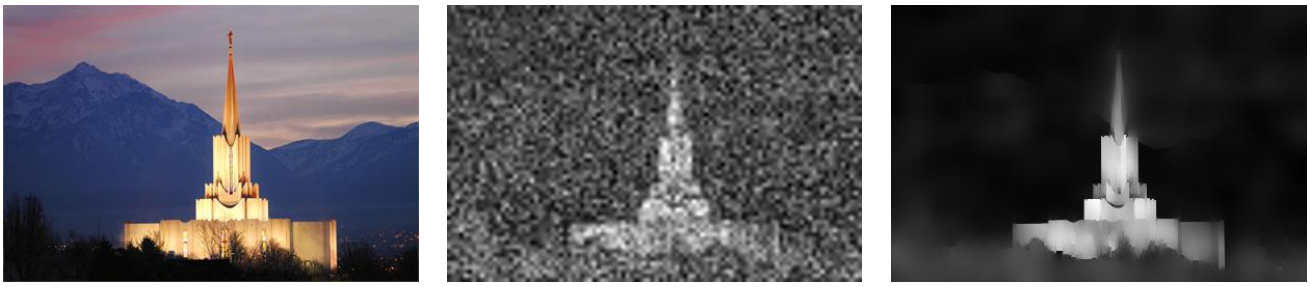
Fast Saliency Detection Using Sparse Random Color Samples and Joint Upsampling

Maiko Min Ian Lie[1], Gustavo Benvenutti Borba[2], Hugo Vieira Neto[1], Humberto Remigio Gamba[1]
[1]Graduate Program in Electrical and Computer Engineering
[2]Graduate Program in Biomedical Engineering
Federal University of Technology – Paraná
Curitiba, Brazil
minian.lie@gmail.com, {gustavobborba, hvieir, humberto}@utfpr.edu.br



(a) Input (b) Approximate saliency map (c) Upsampled saliency map

Fig. 1. From the input image (a), an approximate saliency map (b) is computed using the color distance from each pixel to a sample of random pixels from the same image. The sample size is kept small and is taken from a downsampled version of the input image in order to reduce runtime – the resulting saliency map, however, is noisy. This noisy approximation can be denoised and upsampled into a full-resolution saliency map (c) through joint upsampling using the original resolution input as guide image. Upsampling of a lower resolution approximation is considerably faster than the direct computation of a full-resolution saliency map.

***Abstract*—The human visual system employs a mechanism of visual attention, which selects only part of the incoming information for further processing. Through this mechanism, the brain avoids overloading its limited cognitive capacities. In computer vision, this task is usually accomplished through saliency detection, which outputs the regions of an image that are distinctive with respect to its surroundings. This ability is desirable in many technological applications, such as image compression, video quality assessment and content-based image retrieval. In this paper, a saliency detection method based on color distance with sparse random samples and joint upsampling is presented. This approach computes full-resolution saliency maps with short runtime by leveraging both edge-preserving smoothing and joint upsampling capabilities of the Fast Global Smoother. The proposed method is assessed through precision-recall curves, F-measure and average runtime on the MSRA1K dataset. Results show that the method is competitive with state-of-the-art algorithms in both saliency detection accuracy and runtime.**

***Keywords*-Saliency detection; Fast Global Smoother; Joint upsampling; Visual attention.**

I. Introduction

At any given moment, the human visual system ignores, or at least attenuates, most of the information it receives [1]. This is done mostly through a mechanism of *visual attention*, which selects only part of the incoming information for further processing [2]. Through this mechanism, the brain avoids overloading its cognitive capacities and is able to respond rapidly to stimuli, even in the presence of massive quantities of visual information. Due to this capacity of reducing the amount of incoming visual information to a manageable rate [3] – a highly desirable characteristic in many technological applications – several computational models of visual attention have been proposed in the last decade [4]. These have been applied in areas such as image compression [5], video quality assessment [6] and content-based image retrieval [7].

Most computational models of visual attention are based on the concept of a *saliency map* – a grayscale image that maps each location to an intensity value which is proportional to its conspicuity (i.e. how much this location differs from its surroundings) [8]. This is done mostly using a bottom-up approach, that is, based on low-level features of the image [9], in a process called *saliency detection*. While top-down approaches (i.e. those which involve high level aspects like experience and expectations) exist, they are not as explored due to the complexity of the mental processes involved [3].

Recent approaches have achieved very accurate saliency detection, according to a survey on several popular datasets by Borji and colleagues [4]. Most of these methods, however, are too slow to be used in real-time applications. For example,

among the top performing saliency detectors, according to this benchmark, the *High-dimensional Color Transform* [10] and the *Dense and Sparse Reconstruction* [11] approaches take on average 4.12 s and 10.2 s, respectively, to process a 400×300 image on a Xeon E5645 2.4 GHz CPU with 8 GB RAM [4]. These runtimes are inadequate for many real-time applications, such as adaptive video compression [5] and active robot vision [12] – specially considering that bottom-up saliency detection is meant to reproduce a fast, reflexive mechanism, which has been reported to take less than 150 ms in the human visual system [13].

This paper presents a bottom-up saliency detection method, which computes an approximate saliency map using the color distances of each pixel of the image to a random sample of the other pixels. By reducing the size of the random sample, this approximation can be computed in reduced periods of time at the expense of increasing noise in the saliency map. The noisy saliency map, however, can be corrected without significant increase in runtime through a fast method of edge-preserving smoothing based on weighted least squares (see Fig. 1) [14] – moreover, using the full-resolution input as guide image, it can be computed in a downsampled version of the input and then upsampled into a full-resolution denoised saliency map, using only a fraction of the time it would take to compute a full-resolution saliency map directly. The proposed method is compared to other state-of-the-art saliency detectors and assessed using the MSRA1K dataset [15] through precision-recall curves, F-measure and average runtime. The results show that the method is competitive with state-of-the-art algorithms in terms of saliency detection accuracy, while achieving a relatively short runtime.

II. Related Work

Computational models of visual attention first came to prominence with the model of bottom-up attention by Itti and colleagues [16], one of the earliest to incorporate aspects from psychological theories, most notably Treisman and Gelade's *Feature-Integration Theory of Attention* (FIT) [17], which claims that the process of visual attention involves integrating low-level features of the image (e.g. orientation, color, intensity). Since then, this approach has been incorporated by most models, due to its effective results and biological plausibility.

Following the work of Itti and colleagues [16], many different models have been proposed – a recent benchmark listed more than 30 published since 2008 [4]. While many of them compute saliency maps through combinations of feature maps, most incorporate only color features (including the top performing methods [4]), suggesting that color is the most informative low-level characteristic in terms of saliency.

A definition of saliency in terms of global color distances was given by Zhai and Shah [18]:

$$S_g(x,y) = \sum_{\forall (x_i,y_i) \in I} ||I(x,y) - I(x_i,y_i)||, \qquad (1)$$

in which each pixel of the image I has its saliency S_g computed as the sum of its color distance to *all* the other pixels of the image. The computation of this function has a complexity of $O(N^2)$ for an image with N pixels, which turns out to be impractical for real-time applications given the large number of pixels even for medium sized images (e.g. $400 \times 300 = 120,000$). Considering this, Zhai and Shah compute the saliency of each color instead, which is done in linear time with respect to the number of pixels since the number of colors does not change with image size. Attributing a saliency value to each pixel based on this color saliency has a complexity of $O(N)$. This, however, is still computationally prohibitive considering the large number of possible colors (i.e. $256^3 = 16,777,216$ colors for an 8-bit RGB image), which can be larger than the number of pixels and consequently dominate runtime. To address this, Zhai and Shah use luminance information only. Cheng and colleagues [19] enabled the use of color in this approach by substantially improving the performance through quantization of the color space (12 values per channel) and by ignoring rare colors.

Random sampling, which is at the core of the method proposed here, has been applied to the saliency detection problem before. It was first explored by Stentiford [20] and later by Vikram [21]. Vikram proposed sampling pairs of random sets of pixels and comparing their color distances. These pairs of sets were resampled and compared repeatedly, with the stop condition of reaching a certain number of iterations, which was determined empirically. Another random sampling approach [22], computes the saliency of each pixel as the sum of the absolute differences between its intensity and the mean intensity of random sub-windows containing it, where the number of sub-windows was determined empirically as $0.02 \times w \times h$ for an image with $w \times h$ pixels in size. These sub-windows have random sizes and may overlap, incurring redundant computation.

III. Proposed Method

The proposed saliency detection method leverages a principle from random sampling algorithms, which states that is possible to estimate features of the entire population in a computationally inexpensive way from a small sample [23]. As most methods in the literature [24], it is assumed that the salient object distinguishes itself from the background and distractors with respect to color features.

Our method is composed of two stages: (i) computation of a rough saliency map approximation through comparisons of each pixel of a downsampled version of the input image with a small sample of random pixels from the same image, (ii) post-processing the result of (i) using a fast edge-preserving smoothing method [14] for denoising and joint upsampling. The following subsections describe these steps in more detail.

A. Color distance to random samples

First, the image is converted from RGB to the CIELAB color space, in order to take advantage of its perceptual uniformity (i.e. the Euclidean distance is approximately linear with respect to human visual perception) [25]. Then, the

saliency S of each pixel of the image I is estimated as the sum of its color distances to a random sample I_R of the image, which is different for each pixel:

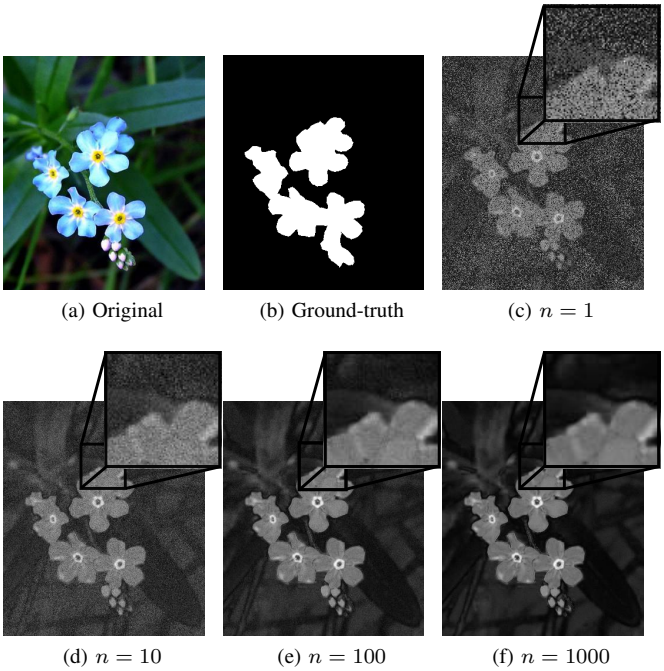
$$S(x,y) = \sum_{\forall (x_r,y_r) \in I_R} ||I(x,y) - I(x_r,y_r)||. \quad (2)$$

Eq. 2 is very similar to Eq. 1, differing only in the set of pixels to which each pixel is compared. While in Eq. 1 this set comprises the entire image I, a much smaller set I_R is adopted, randomly sampled from I. By keeping the size of this sample constant, the proposed approach can be computed with $O(N)$ complexity, for an image I with N pixels, instead of $O(N^2)$, as in Eq. 1.

Unlike the random sampling approach by Vikram [21], the proposed method compares each of all image pixels to a random sample instead of repeatedly comparing two random samples. Using this approach, there is a clear stop condition – the saliency estimation of the last pixel of I, without needing to determine the number of random samples and comparisons empirically.

Fig. 2 shows an image from the MSRA1K [15] dataset, its saliency ground-truth and examples of saliency maps computed using color distances with random samples for different sizes (n) for the set I_R. The ground-truth describes the "ideal" output of a saliency detector for this image, that is, the regions that are considered salient based on human visual inspection [15].

It is possible to observe that for small values of n the saliency map is noisy but, as n increases, it sharpens. However,



(a) Original (b) Ground-truth (c) $n = 1$

(d) $n = 10$ (e) $n = 100$ (f) $n = 1000$

Fig. 2. Examples of saliency maps computed using color differences with random samples for different sample sizes n. The smaller the sample, the noisier the resulting saliency map.

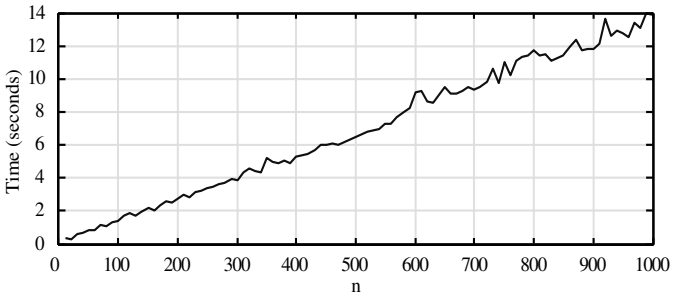


Fig. 3. Runtime for the computation of a saliency map of 400×300 pixels for different values of n, using an Intel Core i7-860 2.80 Ghz CPU. As expected, the runtime increases linearly, but already exceeds 1 s for n approximately equal to 100.

as can be seen in Fig. 3, the runtime also increases considerably. For n approximately equal to 100, runtime already exceeds 1 s, suggesting that, while comparison with random samples describes saliency reasonably well, adopting higher values of n to smooth the saliency map is inadequate for fast saliency detection. Considering this, $n = 3$ is adopted (chosen empirically) to get a fast saliency map approximation and minimize the resulting noise using an edge-preserving smoothing filter.

B. Joint Upsampling

Edge-preserving smoothing removes details of an image by filtering its high-frequency components (e.g. noise, texture), much like Gaussian filtering, but preserving edges, as they might describe information about shape. This task is usually accomplished by filtering not only in *space* but in *range* – that is, the value of a filtered pixel is influenced not only by the spatial distances to the pixels in its neighborhood, as is the case of the Gaussian filter, but also by the range distances (e.g. intensity) to these pixels. Considering this, the approximate saliency map computed previously is subject to the *Fast Global Smoother* (FGS) [14], which is an edge-preserving smoothing filter with $O(N)$ complexity.

The FGS operates based on an optimization framework, which is accelerated by an approximation as a sequence of 1D subsystems that minimizes the following energy function [14], which is computed for each row/column:

$$J(u) = \sum_n \left((u_n - f_n)^2 + \lambda \sum_{i \in \mathcal{N}(n)} w_{n,i}(g)(u_n - u_i)^2 \right), \quad (3)$$

with input image f, guide image g and output image u. These image rows/columns are defined along $n \in [0, L]$, where L is the width or height of the input image if Eq. 3 is being applied to a row or a column, respectively. $\mathcal{N}$ is the set of the two neighbors of n (i.e. $n-1$ and $n+1$). The coefficient λ is the *smoothness parameter* – the larger its value, the smoother the output. The function $w_{n,i}(g)$ determines the similarity between the pixels n and i in the image g and is defined as:

$$w_{n,i}(g) = exp\left(\frac{-||g_n - g_i||}{\sigma_c} \right), \quad (4)$$

where σ_c denotes the *range parameter*. The approximate saliency map was filtered using 3 iterations and $\sigma_c = 0.03$, as suggested in [14], while $\lambda = 10^2$ was determined empirically.

Although simple edge-preserving smoothing can be useful for denoising, the proposed method leverages a specific characteristic of the model adopted by FGS – the decoupling between the sources of domain information and range information. By taking domain information (f in Eq. 3) from the noisy saliency map and range information (g in Eq. 3) from the full-resolution input image, it is not only possible to denoise the saliency map but also upsample it. Thus, the runtime of the method can be improved not only by reducing the size of the set of random pixels in the computation of the approximate saliency map but also by doing it in a downsampled version of the input. The details lost in this process are predominantly the object contours and texture information. The former is corrected by joint upsampling with the full-resolution guide image, while the latter actually improves detection accuracy, as salient region detection aims at detecting homogeneous regions, not the details inside of it [24]. Examples of saliency map approximations in the downsampled image, as well as the result of their upsampling are presented in Fig. 4.

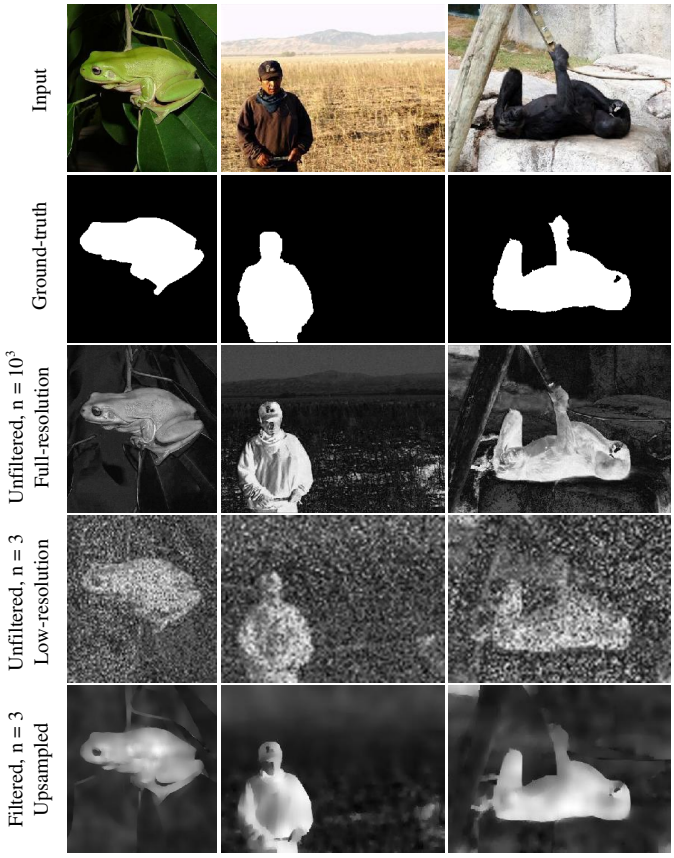


Fig. 4. Comparison of saliency maps computed using: large n in full-resolution, small n in lower resolution and upsampling of the latter. The ground-truth and input are also presented for comparison. Upsampling results in more homogeneous regions and can be computed much faster than just increasing n.

Upsampling a solution computed in a downsampled input by using information from the full-resolution input was explored before in the context of the *Bilateral Filter* [26], under the name of *Joint Bilateral Upsampling* [27]. Kopf and colleagues [27] showed that it can be used in a series of applications, such as tone mapping, colorization, and depth from stereo. In fact, the joint upsampling application in the proposed method can be considered a case of "colorization", where the salient regions of the downscaled approximate saliency map are grayscale scribbles used to fill the regions of the full-resolution input image. Even though joint upsampling can be realized by the *Bilateral Filter* and others [28], the *Fast Global Smoother* was adopted due to its fast execution and ease of parameterization.

C. Summary of the method

The method just presented is summarized in Fig. 5. First the input image is converted from RGB to the CIELAB color space. Then the image is downsampled and the color distances to sparse color samples are computed through the application of Eq. 2. The output is a downsized noisy saliency map, that is then filtered by FGS using the original sized input as guide image, which has the effect of denoising and upsampling it. This ensures that even though the originally computed saliency map is noisy and downsampled, the method still manages to output high resolution homogeneous saliency maps. The guidance is conducted using a color image because it can preserve better the edges that are not distinguishable in grayscale [28]. The upsampled saliency map is subject to gamma correction with $\gamma = 2$ to suppress eventually remaining noise in the background.

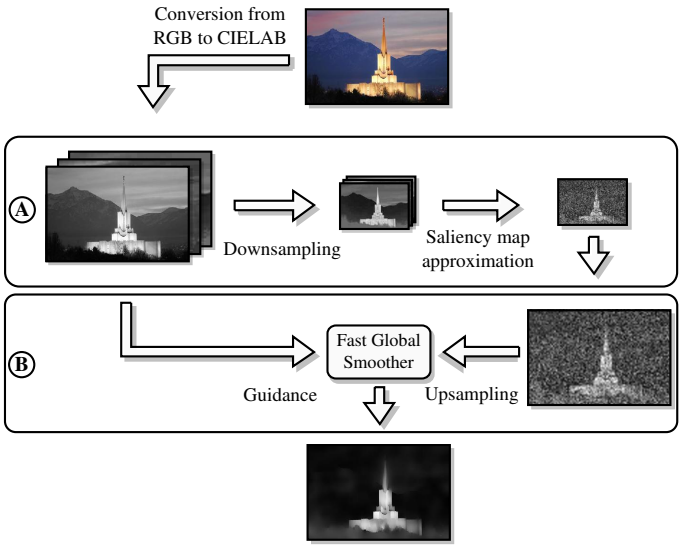


Fig. 5. Summary of the proposed method. (A) Conversion from the RGB to the CIELAB colorspace and saliency map approximation using color distance with random sparse samples – the sample size is kept small and computation is done with a downsampled version of the input to reduce runtime. (B) Joint upsampling of the approximate saliency map using the Fast Global Smoother.

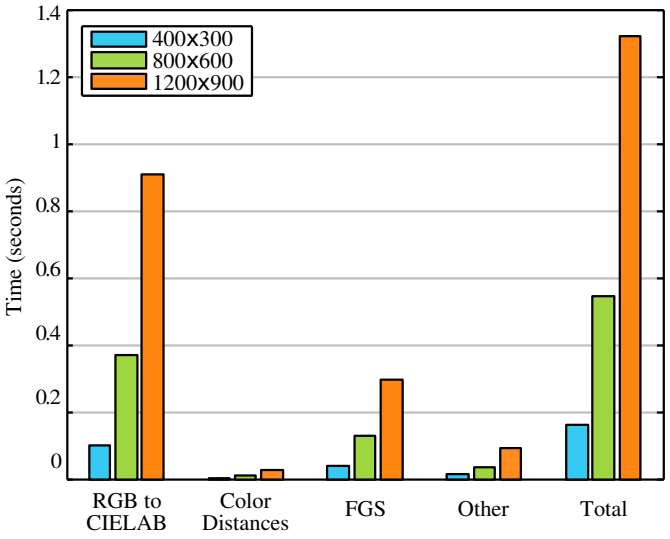


Fig. 6. Runtime of the most important steps on the proposed method for input of sizes: 400×300, 800×600 and 1200×900. Computation of color distances to random sparse samples is very fast and its denoising/upsampling using FGS can be done in real-time even for large images. The bottleneck, however, is the colorspace conversion from RGB to CIELAB.

An analysis of the runtime of the proposed method is given in Fig. 6, which presents the individual runtimes of its main steps: RGB to CIELAB colorspace conversion, color distance computation with sparse random samples, FGS, other operations (i.e. range normalization) and total runtime. As image dimensions double, the number of pixels increases exponentially, as does the runtime of each step in the proposed method, due to their linear complexity. Color distance computation is very fast even for a 1200×900 image, confirming the hypothesis that correcting an approximate saliency map is promising for fast detection. FGS is also fast, suggesting that it is a reasonable choice for real-time applications even for large images. Surprisingly, the bottleneck is the colorspace conversion from RGB to CIELAB, currently implemented using the MATLAB functions *makecform* and *applycform*, and has longer runtime than the rest of the steps combined. However, as visual saliency is inherently related to human perception, a perceptually uniform colorspace is desirable (as shown by recent work [24]) even though it involves costly nonlinear operations. Optimization of these operations is possible [29], but out of the scope of this paper and subject of future work.

IV. Experiments and Discussion

A. Experimental setup

The proposed method was assessed using precision-recall curves, F-measure and average runtime using the MSRA1K dataset [15]. The MSRA1K dataset contains 1000 images, and their respective ground-truths, of many unambiguous (in terms of what is salient) indoor and outdoor scenarios with animals, flowers, and other objects. It is currently the most popular dataset used for saliency detection assessment in the literature [24]. Our method is compared to other four state-of-the-art saliency detectors: Frequency-tuned (FT) [15], Spectral Residual (SR) [30], Random Center Surround (RCS) [22] and Absorbing Markov Chain (AMC) [31]. The motivation for this choice of algorithms for comparison were: number of citations (FT and SR have both more than 1000 citations each), similarity to the proposed approach (RCS is also based on random color distances) and performance (AMC is the fastest among the most accurate algorithms in the most extensive benchmark currently available in the literature [4]). The experiments were run on an Intel Core i7-860 2.80 Ghz CPU with 4 GB RAM.

B. Metrics

The assessment is based on the precision, recall, F-measure and runtime metrics. Precision and recall are standard metrics in saliency detection assessment [24] and are adopted mainly due to the superior visual distinctiveness of their curves relative to, for example, ROC curves on highly-skewed data [32]. They are defined as:

$$Precision = \frac{TP}{TP + FN}, \qquad Recall = \frac{TP}{TP + FP}, \quad (5)$$

where TP (true positives) are salient pixels correctly detected as such, FN (false negatives) are salient pixels detected as background and FP (false positives) are background pixels detected as salient. F-measure is a metric used to summarize precision and recall information in a single value, defined as:

$$F_\beta = (1 + \beta^2)\frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall}, \quad (6)$$

where, in saliency detection assessment, it is usual to adopt $\beta^2 = 0.3$ to give precision more weight than recall, as many authors consider the former more important [11], [15], [19].

C. Results and discussion

Fig. 7 shows the precision-recall curves for the assessed methods when applied to the MSRA1K dataset. Each curve is computed by thresholding the output of a saliency detector through the entire range of possible thresholds ([0, 255] for an 8-bit saliency map) and computing precision and recall for each case. By taking the average curve of all the images in the dataset for all the compared methods, the curves in Fig. 7 are obtained.

Table I summarizes the average runtimes of the compared methods on the MSRA1K dataset, as well as their F-measures. Unlike in the curves presented in Fig 7, instead of averaging precision and recall for each possible threshold, only one adaptive threshold was used: twice the average saliency of the image, as recommended in [15]. This was done in order to keep the results comparable to the majority of previous work, which also adopts this approach.

TABLE I
Average runtime and F-measure per image – MSRA1K dataset.

Method	SR	FT	RCS	AMC	Proposed
Runtime (s)	0.0090	0.0582	0.7333	0.1826	0.1117
F-measure	0.4819	0.7070	0.6607	0.9059	0.7363

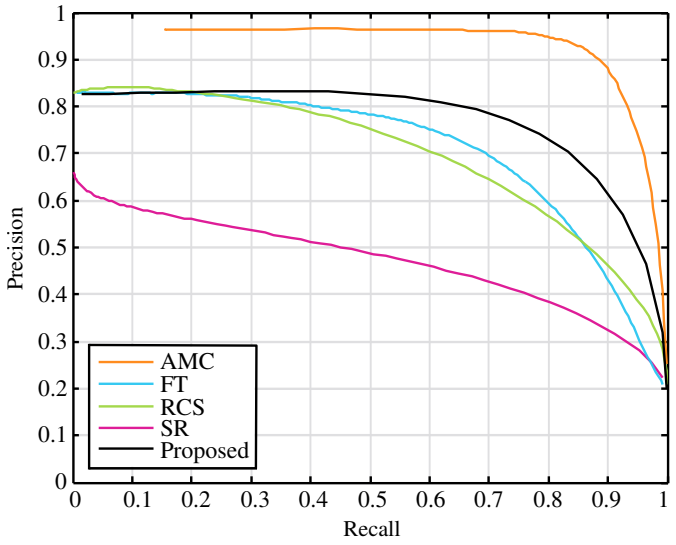


Fig. 7. Precision-recall curves for the compared methods on the MSRA1K dataset. By computing saliency maps with more homogeneous regions, the recall of the proposed method is superior to most of the compared methods.

The proposed method is capable of accurately detecting salient regions while being reasonably fast, as can be noticed in Fig. 7 and Table I. In fact, it has the second highest F-measure, even though one of its main advantages relative to the other methods (i.e. higher recall due to more homogeneous regions) was weighted down by adopting $\beta^2 = 0.3$. It also presents one of the best accuracy-runtime trade-offs among the assessed methods. AMC has the most accurate detection – however, it is also one of the slowest. AMC is slowed largely due to its graph-based model using superpixels as nodes [31]. While oversegmentation into superpixels facilitates homogeneous regions with well-defined boundaries to emerge, it results in substantial computational cost. The oversegmentation algorithm used by AMC is SLIC (*Simple Linear Iterative Clustering*) [33], which is is one of the fastest in the literature – even so, of the 0.1826 s average runtime taken by AMC, 0.1188 s are dedicated to SLIC. That is, its oversegmentation algorithm alone takes, on average, more time to compute than the entire proposed method.

FT and RCS have similar detection accuracy – however, the former is superior to the latter both in accuracy and, more significantly, runtime. FT is fast mainly because it makes only one comparison per pixel – it estimates the saliency of a pixel as its color distance to the average color of the image – and does not involve any filtering besides slightly blurring the input image (Gaussian filter using a 5×5 window). The main drawback of this approach is that it emphasizes details inside the salient region, one of the main reasons for a worse accuracy when compared to the proposed method, even though it is computed faster. RCS has the worst runtime among the compared methods, which can be largely attributed to the number of sub-windows generated in its computation: $n \times w \times h$ for an image with $w \times h$ pixels in size. For a medium sized image with 400×300 pixels, it generates 2400 sub-windows, each with a random size. Even though it is also based on random sampling, the proposed method outperforms RCS on both detection accuracy and runtime.

SR is the fastest method among the compared, by far. Its speed is not only due to its simplicity (i.e. it is computed basically as a difference in the frequency domain), but also because the input is downsampled to 64 pixels in width or height before being processed. These characteristics, apart from making the algorithm fast, make it very inaccurate for detecting salient regions, as can be observed by its considerably low accuracy in Fig. 7 and Table I, the worst among the compared methods. As such, it is usually considered for fixation prediction, instead of salient region detection [4].

Fig. 8 shows some saliency maps computed by the compared methods for qualitative assessment. As can be observed, the proposed method results in homogeneous salient regions with well-defined edges. It does not emphasize the center of the image (like AMC and RCS), edges (like SR) or details

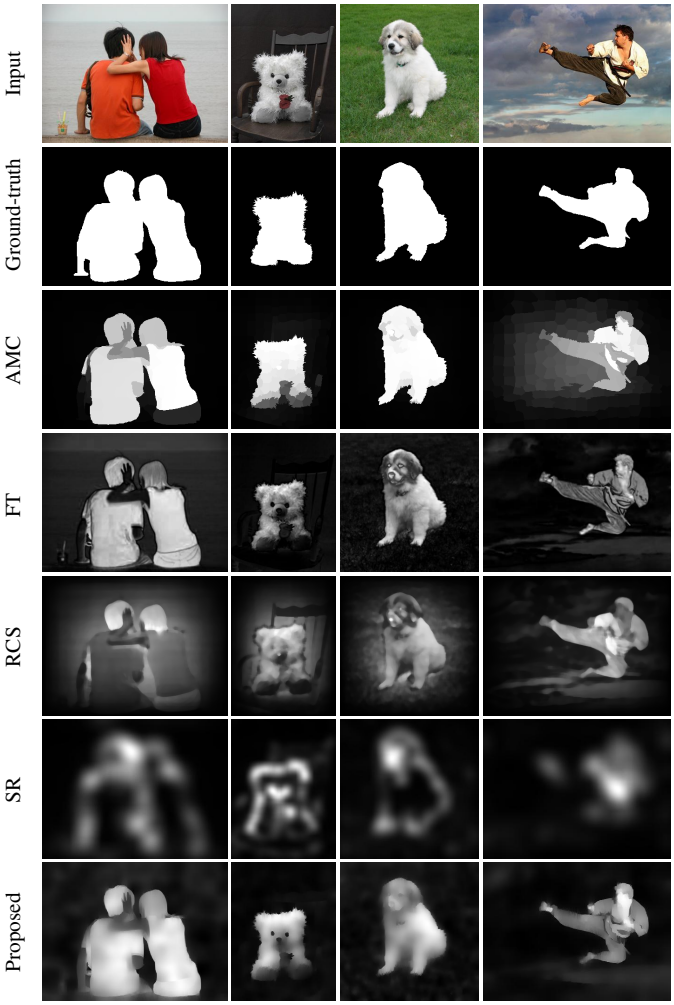


Fig. 8. Comparison of saliency maps from the assessed methods on some images from the MSRA1K dataset. The proposed method has well-defined edges as well as homogeneous salient regions and, unlike AMC and RCS, does not emphasize the center of the image.

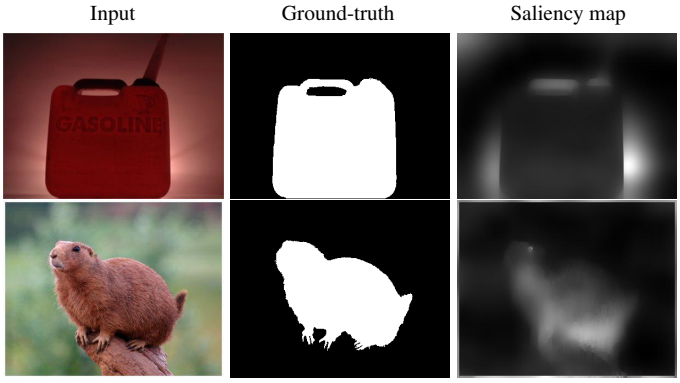


Fig. 9. Cases in which the proposed method fails. Top: the color of the salient region does not stand out from the background. Bottom: the salient region is bounded by fuzzy edges.

inside the salient region (like FT, RCS and SR). However, there are limitations, shown in Fig. 9. When the color of the salient region does not particularly stand out from the background, the method fails, which is to be expected as it operates entirely on that premise. Another situation where the proposed method fails is when the salient region has fuzzy edges, in which case, it is simply smoothed.

V. Conclusion

A saliency detection method based on color distances using random samples and joint upsampling was presented. By computing color distances to small random samples in a downsampled image, but using the original image in the CIELAB colorspace as guide image, the method is able to achieve competitive detection accuracy with state-of-the-art methods while still maintaining fast computation.

While earlier methods based on random sampling had disadvantages such as empirical stop conditions and redundant computations, the method presented here does not present such limitations. The results presented in this paper show that the proposed approach is promising, suggesting further investigation on modeling saliency in a framework of reconstruction of noisy saliency map approximations, as they can be computed very fast. Noise estimation algorithms have been used to make the parameters of the *Bilateral Filter* adaptive [34], suggesting possible improvements to the *Fast Global Smoothing* filter in the proposed saliency detection method. In future work we also intend to explore sampling from a Gaussian distribution instead of an uniform distribution, in order to approximate the center-surround characteristics of the human visual system [1]. Future work also includes investigating the impact of computing the saliency map on perceptually uniform colorspaces with conversion formulas simpler than CIELAB, since the conversion from RGB to CIELAB is the most expensive step in the proposed method.

Acknowledgment

The authors would like to acknowledge the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES) for the financial support of this work.

References

[1] J. M. Wolfe, "Guided Search 2.0 A Revised Model of Visual Search," *Psychonomic Bulletin & Review*, vol. 1, no. 2, pp. 202–238, 1994.

[2] R. Desimone and J. Duncan, "Neural Mechanisms of Selective Visual Attention," *Annual Review of Neuroscience*, vol. 18, no. 1, pp. 193–222, 1995.

[3] S. Frintrop, *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search (Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[4] A. Borji, M. M. Cheng, H. Jiang, and J. Li, "Salient Object Detection: A Benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015.

[5] N. Ouerhani, J. Bracamonte, H. Hügli, M. Ansorge, and F. Pellandini, "Adaptive Color Image Compression Based on Visual Attention," in *Proceedings of the 11th International Conference on Image Analysis and Processing*. IEEE, 2001, pp. 416–421.

[6] D. Ćulibrk, M. Mirković, V. Zlokolica, M. Pokrić, V. Crnojević, and D. Kukolj, "Salient Motion Features for Video Quality Assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 948–958, 2011.

[7] Marques, Oge and Mayron, Liam M. and Borba, Gustavo B. and Gamba, Humberto R., "Using Visual Attention to Extract Regions of Interest in the Context of Image Retrieval," in *Proceedings of the 44th Annual Southeast Regional Conference*. New York, NY, USA: ACM, 2006, pp. 638–643.

[8] Koch, Christof and Ullman, Shimon, "Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry," in *Matters of Intelligence: Conceptual Structures in Cognitive Neuroscience*, L. M. Vaina, Ed. Dordrecht: Springer Netherlands, 1987, pp. 115–141.

[9] S. Yantis, "Goal-Directed Stimulus-Driven Determinants of Attentional Control," in *Control of Cognitive Processes*, S. Monsell, J. Driver, S. Monsell, and J. Driver, Eds. MIT Press, 2000.

[10] J. Kim, D. Han, Y. W. Tai, and J. Kim, "Salient Region Detection via High-Dimensional Color Transform," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, June 2014, pp. 883–890.

[11] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency Detection via Dense and Sparse Reconstruction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2013)*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 2976–2983.

[12] A. F. Roos and H. Vieira Neto, "Towards Saliency-based Gaze Control in a Binocular Robot Head," in *Proceedings of the 6th UNICAMP Symposium on Signal Processing*, Campinas, Brazil, 2015.

[13] J. Theeuwes, "Top-down and Bottom-up Control of Visual Selection," *Acta Psychologica*, vol. 135, no. 2, pp. 77 – 99, 2010.

[14] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do, "Fast Global Image Smoothing Based on Weighted Least Squares," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5638–5653, 2014.

[15] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned Salient Region Detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009, pp. 1597 – 1604.

[16] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[17] A. M. Treisman and G. Gelade, "A Feature-integration Theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97 – 136, 1980.

[18] Y. Zhai and M. Shah, "Visual Attention Detection in Video Sequences Using Spatiotemporal Cues," in *Proceedings of the 14th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2006, pp. 815–824.

[19] M. M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. M. Hu, "Global Contrast Based Salient Region Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, March 2015.

[20] F. Stentiford, "An Estimator for Visual Attention through Competitive Novelty with Application to Image Compression," in *Picture Coding Symposium*, Seoul, Korea, 2001, pp. 24–27.

[21] T. N. Vikram, "Random center-surround approaches for modeling visual saliency," Ph.D. dissertation, Bielefeld University, 2013.

[22] T. N. Vikram, M. Tscherepanow, and B. Wrede, "A Saliency Map Based on Sampling an Image Into Random Rectangular Regions of Interest," *Pattern Recognition*, vol. 45, no. 9, pp. 3114 – 3124, 2012.

[23] R. Motwani and P. Raghavan, "Randomized algorithms," *ACM Computing Surveys*, vol. 28, no. 1, pp. 33–37, 1996.
[24] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient Object Detection: A Survey," *arXiv preprint arXiv:1411.5878*, 2014.
[25] E. Reinhard, E. A. Khan, Ahmet, O. Akyüz, and G. M. Johnson, *Color Imaging: Fundamentals and Applications*. Wellesley, MA: AK Peters, 2008.
[26] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *Proceedings of the Sixth International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–846.
[27] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint Bilateral Upsampling," *ACM Transactions on Graphics*, vol. 26, no. 3, Jul. 2007.
[28] K. He, J. Sun, and X. Tang, "Guided Image Filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
[29] C. Connolly and T. Fleiss, "A Study of Efficiency and Accuracy in the Transformation from RGB to CIELAB Color Space," *Transactions on Image Processing*, vol. 6, no. 7, pp. 1046–1048, Jul. 1997.
[30] X. Hou and L. Zhang, "Saliency Detection: A Spectral Residual Approach," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, 2007, pp. 1–8.
[31] B. Jiang, L. Zhang, H. Lu, C. Yang, and M.-H. Yang, "Saliency Detection via Absorbing Markov Chain," in *IEEE International Conference on Computer Vision (ICCV 2013)*. IEEE, 2013, pp. 1665–1672.
[32] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: ACM, 2006, pp. 233–240.
[33] R. Achanta and A. Shaji and K. Smith and A. Lucchi and P. Fua and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov 2012.
[34] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang, "Noise Estimation from a Single Image," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 901–908.