Salient Edges: A Multi Scale Approach

Michal Holtzman-Gazit[1], Lihi Zelnik-Manor[2], and Irad Yavneh[1]

[1] Computer Science Department, Technion I.I.T, Technion City, Haifa 32000, Israel,
{mikih,irad}@cs.technion.ac.il
[2] Electrical Engineering Department, Technion I.I.T, Technion City, Haifa 32000, Israel,
lihi@ee.technion.ac.il

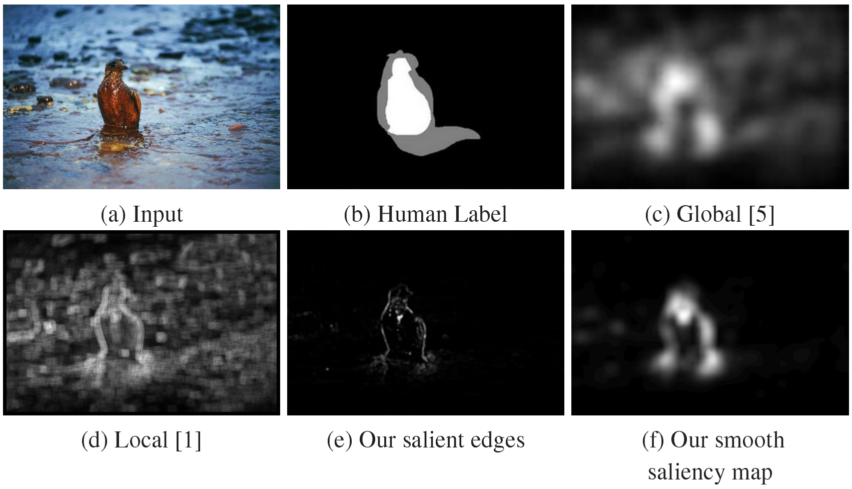
Abstract. Finding the salient features of an image is required by many applications such as image re-targeting, automatic cropping, object tracking, video encoding, and selective sharpening. In this paper we present a novel method for detection of salient objects' edges which combines local and regional considerations. Our method uses multiple levels of detail, and does not favor one level over another as done in other multi-scale methods. The proposed local-regional multi-level approach detects edges of salient objects and can handle highly textured images, while maintaining a low computational cost. We show empirically that these are useful for improving image abstraction results. We further provide qualitative results together with quantitative evaluation which shows that the proposed method outperforms previous work on saliency detection.

Key words: Saliency, Multi-scale, Edge detection, Scene analysis

1 Introduction

Images contain large amounts of visual information, and the human visual system usually focuses on certain parts of the image which are considered to be more important. Many applications rely on recovering these salient regions of the image, for example, image re-targeting, automatic cropping, video encoding and more. Other application such as selective sharping, video abstraction, and object recognition, depend on the recovery of boundaries of salient objects, in order to emphasize or track only these edges. In this paper we present a novel approach for the detection of salient edges.

In recent years, several methods were developed for visual saliency estimation, which can be divided into two main approaches, bottom-up saliency and top-down saliency. Bottom-up saliency models [1–5] first extract low level visual features and then compute saliency models using these features. The methods of [1],[2],[3], were inspired by the human visual system, to extract low level features and aggregate them into salient regions. Since they do not incorporate global considerations explicitly they often detect non-interesting background texture as salient (see Fig. 1(d) and Fig. 2(b)(c)). In [4] coarse global information was merged hence their final saliency maps lack fine details, and often miss parts of the salient objects. The method of [5], detects global saliency by identifying image regions which imply unusual frequencies in the Fourier domain. This method does not rely on parameters and detects saliency extremely fast. However, since it does not consider local saliency the results are coarse and many fine details are missed (see Fig.1(c)).


(a) Input

(b) Human Label

(c) Global [5]

(d) Local [1]

(e) Our salient edges

(f) Our smooth saliency map

Fig. 1. Global methods are too coarse while local methods detect background texture as salient. Our multi-level approach which unifies local and regional saliency detects only the **edges** of the salient objects. The human label was taken from [5], where the image was labeled by four subjects. In the labeled map, the white region represents salient region; the black region represents the non-salient region, and the gray region was selected by some labelers but rejected by others.

Top-down models are task driven and involve high level information. Saliency models combining top-down and bottom-up were used in [6–8]. The global saliency measure of [6] detects the single most salient rectangular region in the image. When the objects are not rectangular, this coarse analysis leads to inaccurate detections. In [7] a robust method combined local, global and high level considerations explicitly. This combination was shown in [7] to be highly successful, however, their approach aims at recovering part of the background as salient and not only the dominant objects. Judd et al. [8] successfully trained a bottom-up top-down model based on low, mid, and high-level features. However, these top-down bottom-up methods are highly computational. In addition, none of the above methods extracts salient edges. Our approach integrates multi-level local saliency with multi-level regional saliency and provides an accurate map of only the edges of salient objects.

In this paper we present a multi-level method for extracting edges of salient objects. Our method combines low level local and global considerations, and unlike other multi-scale methods, we do not favor one level over another. We combine information from all resolution levels of the image, in order to form a robust and fast method for generating a map of salient edges. To compute salient edges at multiple levels we adopt the Beltrami operator [10] which provides multiple resolutions while maintaining edge information. We further incorporate a multi-level regional analysis which detects regionally salient objects. Merging the explicit regional component, with the multi-scale local detection, yields detailed detection of the globally salient edges. For applications which require a saliency map rather than salient edges, we locally smooth the result around the detected

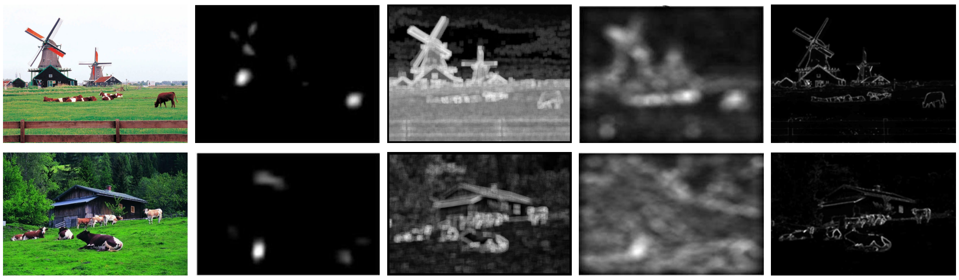

(a) Input (b) Saliency of [2] (c) Saliency of [1] (d) Saliency of [9] (e) Salient Edges

Fig. 2. Saliency results of different algorithms on images obtained from [9].

edges. An example result of our method compared with other algorithms is given in Fig. 1,2. Fig.1 (e)(f), shows our saliency map which demonstrates only the bird's edges, while the details are maintained. In Fig.2 (e) we detect all the outlines of the salient objects while all other methods show a blurry map. Some show background regions as salient (Fig.2 (c)(d)), while others miss some salient regions (Fig. 2(b)) and none show salient edges.

We propose a local measure of saliency based on a multi-level extension of [11]. There, it was shown that the inverse transform of the phase of the input image, provides a high quality saliency map which is fast to compute and does not require any parameter tuning. In [11] the input image is re-scaled a-priori to a pre-determined fixed resolution. Since every resolution has different edge information, some unimportant structures are detected as salient while other salient structures are often missed by this approach. A hierarchical multi-resolution extension of [11] was presented in [9]. There, three resolutions are used in a coarse to fine manner. A salient region found at the coarsest level, is then explored by the finer levels in order to sub-divide it into separate objects. However, regions not detected by the coarsest level (such as small objects), cannot be found at the finer resolutions. Results for this method are shown in Fig. 2(d), where the final saliency map is blurry and detects background as salient. Instead, we propose here an approach which combines information from multiple levels, without giving preference to any particular scale, in order to recover a map of salient edges.

We further propose a multi-level regional saliency measure which identifies globally salient regions. The proposed measure can handle objects of any shape and size. As in the local measure, we make sure that our algorithm maintains a low computational cost. Hence, the complexity of our overall algorithm remains linear in the number of pixels.

The rest of the paper is organized as follows: In section 2 we provide a detailed review of the related previous work, describe the two stages of our algorithm and how they are combined into a single saliency map. In section 3 we demonstrate how to use the edges recovered by our saliency map together with a video abstraction application, for enhancing the abstraction results. In section 4 we demonstrate our visual results and present a quantitative comparison with other algorithms using the databases of [1] and [5]. Finally, we conclude in section 5.

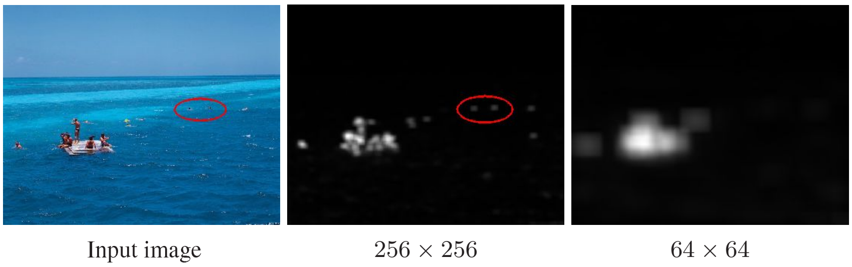


Fig. 3. Results of [11] using different resolutions of the image proposed in [9]. The original resolution of the image is 1000×800. The people marked by the red ellipse are not detected as salient in the coarse scale.

2 The algorithm

2.1 Main concepts

We follow three main principles. The first, is that a good saliency algorithm needs to be based on all resolution levels of the image. A single level of the image does not contain all the data necessary for saliency detection. Second, we believe that no level is more important than others. Unlike other multi-scale algorithms, we treat all image levels equally, and find a solution which accounts for all of them. By using multiple levels we implicitly incorporate local and global saliency. Finally, the third principle states that one needs to consider both local and global saliency explicitly. Hence, we further propose a multi-scale regional analysis which captures global saliency explicitly.

Our algorithm, therefore, unites two parts, local and regional. The local stage extends [11] where only a single resolution level of the image was processed. Using a single scale, regions which locally have high-contrast changes may appear as salient even though they are in fact part of a larger texture. When inspecting different levels of detail, different objects appear as salient in the different levels. Fig. 3 shows results of the saliency algorithm of [11] applied to different resolutions of the same image. The different resolutions result in different saliency maps. In [9] a multi-scale extension of [11] is proposed. However, in their approach an object not detected in the coarsest resolution would be lost in the coarse-to-fine approach proposed in [9]. For example in Fig. 3, some of the people in the water appear as salient in the higher resolution but not in the lower resolution. We propose a different extension of [11] where we search for saliency in all the image levels, taking them all into account in a unified solution. Furthermore, rather than using a smoothing Gaussian kernel to obtain multiple image resolutions, we use the Beltrami operator which maintains the image details.

The second part of our algorithm consists of a regional analysis. Locally, a point can be regarded as salient since it is part of an edge in all levels of the image. Regionally, this point may actually be a part of a larger texture. In order to avoid detecting textured background as salient, we add a regional stage which inspects each region in the image in comparison with its neighborhood. Again, since the characteristic length scale of the texture may vary, we explore regions of various sizes. Then, we generate a saliency map which is composed of the analysis of all region sizes together. Finally, we define saliency in locations which are salient both locally and regionally, generating a map of salient edges that combines the information produced by the local and regional algorithm. Together, these steps provide an efficient and reliable saliency algorithm.

2.2 Reviewing Phase Based Saliency of [11]

We start by reviewing the algorithm proposed in [11] for phase-based saliency computation at a single scale. We modify the notation of [11] slightly since we currently handle only single images rather than video frames.

Let r, g, and b be the red green and blue channels of the input image, respectively. Three channels, $RG\ BY\ I$, are defined to be used in the Quaternion Fourier transform.

$$RG = 1.5\,(r - g) \tag{1}$$

$$BY = b - r + b - g + |r - g|/2 \tag{2}$$

$$I = (r + g + b)\,/3 \tag{3}$$

RG is a red green channel, and BY is a blue yellow channel, designated to match the color pairs which exist in the visual cortex of the human brain. I is the intensity channel. These channels are represented as a quaternion image as follows:

$$q = RG\mu_1 + BY\mu_2 + I\mu_3, \tag{4}$$

where $\mu_i \forall i = 1, 2, 3$ satisfies $\mu_i^2 = -1$, $\mu_1 \bot \mu_2$, $\mu_3 \bot \mu_2$, $\mu_3 \bot \mu_1$, $\mu_3 = \mu_1\mu_2$.

A Quaternion Fourier transform [12] is applied to q. The magnitude of the Fourier transform is set to 1, and an inverse Quaternion Fourier transform is computed:

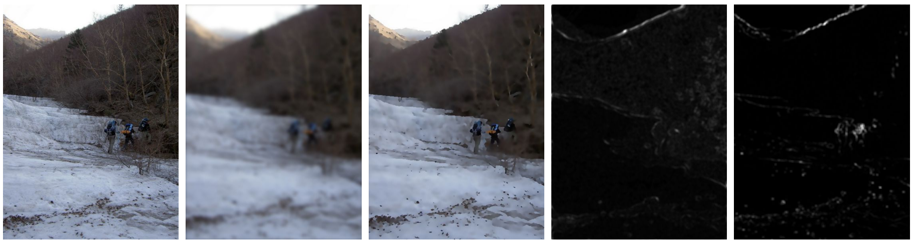
$$q' = \mathfrak{F}^{-1}\left\{\frac{\mathfrak{F}\{q\}}{\|\mathfrak{F}\{q\}\|}\right\}. \tag{5}$$

According to [11], the saliency map is expressed as follows:

$$Sl_{single-level} = g * \|q'\|^2 \tag{6}$$

where g is a 2D Gaussian filter (we use 7 pixels wide with $\sigma = 3$). Since $\|q'\|$ is actually an image with high values at the edges of objects and low values everywhere else, the purpose of applying g is to smooth these edges, thus filling objects in the saliency map. In [11] the resolution of all input images is reduced to 64×64 prior to computing the saliency map. These maps are later re-scaled to the original size of the input image.

The benefit of this method is that it is a fast and simple way to get a saliency map of an image, but it also has some flaws. Reducing the size of every image to 64×64 regardless of the original image size, removes some textures from the image - thus helping the saliency map accentuate relatively large objects in the image. However, it may lose important data as well. Small objects, which are sometimes unique, might also be removed by this down-scaling. As can be seen in Fig. 3, when using the high resolution all the details of the raft and the people in the water are detected, while at the lowest resolution some of the swimmers are removed from the saliency map and the entire map is blurred and inaccurate.



(a) Input (b) After Gaussian filtering (c) After Beltrami filtering (d) Saliency of (b) (e) Saliency of (c)

Fig. 4. The hikers are not detected as salient in (d) since Gaussian smoothing blurs edges. They are detected in (e) since anisotropic filtering preserves the edges.

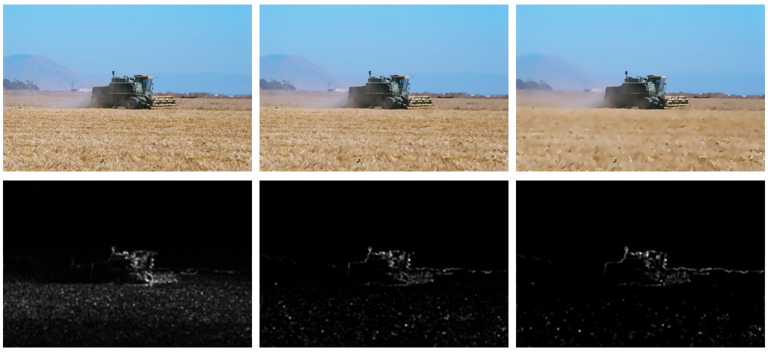


Fig. 5. Top: different detail levels of an image. Bottom: saliency results on the different image levels show that each captures different features.

2.3 Multi Level Phase Based Saliency

The common approach to multi-scale uses a Gaussian pyramid. The Gaussian smoothing process smears edges, hence, it is not well suited for saliency detection. This is illustrated in Fig. 4(b) which shows how Gaussian smoothing blurs the non-interesting texture in the trees as well as the salient people hiking in the snow. Instead we use the anisotropic diffusion proposed in [10] (see Fig. 4(c)). To avoid artifacts we do not down-sample the image. Instead we obtain multiple levels of detail by applying the Beltrami flow of [10] with different smoothing levels. Each level has a different smoothness level, therefore less texture is shown as the smoothing proceeds. Objects in high contrast with their background maintain their edges, while their inner parts are smoothed. In addition, textures such as fur and grass are smoothed more and more as we apply more iterations of the Beltrami flow. This process is an edge-preserving smoothing which tends to eliminate textures whilst retaining sharp edges (see Fig.4(c)). Fig. 4 further compares the saliency results of Eq.(6) when applied to the Gaussian smoothed image of Fig. 4(b) and the anisotropic diffused image of Fig.4(c). The people in the image are nearly lost in Fig. 4(d) since Gaussian smoothing was applied. Conversely, the people are nicely captured as salient in Fig. 4(e) when using edge preserving smoothing.

Fig. 5 (top) shows different levels of detail of an image. These levels were generated using anisotropic diffusion specifically the Beltrami flow of [10], with different smoothing level for each version. Each level has a different smoothness level, therefore less texture is shown as the smoothing proceeds. Fig. 5 (bottom) shows the saliency results of each image level applying the local saliency of Eq. (6) to the full resolution image. The saliency figure of the most detailed image (on the left) shows the harvest machine as well as the hay field as salient. However, the back edge of the machine is not shown well in the saliency map. As the smoothing proceeds we see less of the hay as salient. In the saliency map of the smoothest level (on the right), the back of the harvest machine is shown well but the cutting knives are less distinctive. This comes to show that several levels are necessary to generate a good saliency map, and that no level is more important than the others.

Our algorithm uses n different levels of detail for the saliency map. Let $I_0 ... I_{n-1}$ be the different image levels, where I_0 is the original image and I_{n-1} is the smoothest level. These levels are generated using a progressively stronger smoothing in the Beltrami operator. At each level we apply the phase spectrum Fourier transform algorithm of Eq.(6). When merging information from multiple levels we wish to mitigate the impact of large outliers and aggravate the impact of small ones. Therefore, if a certain pixel is salient in a single level but not in the others, it should have only a little impact on the multi-level saliency map. To achieve this behavior in a fast and simple way we combine the multi-level saliency maps using the harmonic mean. Let Sl_i be the saliency map of image level I_i. The multi-level phase based saliency is defined as the harmonic mean of all these saliency maps, as follows:

$$Sl_{multi-level} = \left(\frac{1}{n} \sum_{i=0}^{n-1} \frac{1}{Sl_{single-level}(I_i)} \right)^{-1} \tag{7}$$

The harmonic mean presents the desired properties of increasing the saliency of pixels which are salient at multiple levels. We have evaluated empirically several other averaging techniques and the harmonic mean was found to generate the best results.

The multi-level phase-based saliency of Eq.(7) combines implicitly global and local information by using different levels of detail. Nevertheless, since each pixel is treated independently and no continuity constraints are applied, occasionally isolated pixels are detected as salient. To reduce this effect we further apply a 5×5 median filter to the saliency map of Eq.(7) .

$$S_{local} = MedianFilter(Sl_{multi-level}). \tag{8}$$

Fig. 6 shows the results of the multi-level saliency compared with our implementation of the algorithm of Guo et al. [11], which reduces the image to a resolution of 64×64. The results of the multi-level saliency are more precise and give better detail, showing only the important edges, which appear in all levels.

While the proposed multi-level saliency provides significant improvement over single-scale saliency, there are still images where this is not sufficient. For example, when there

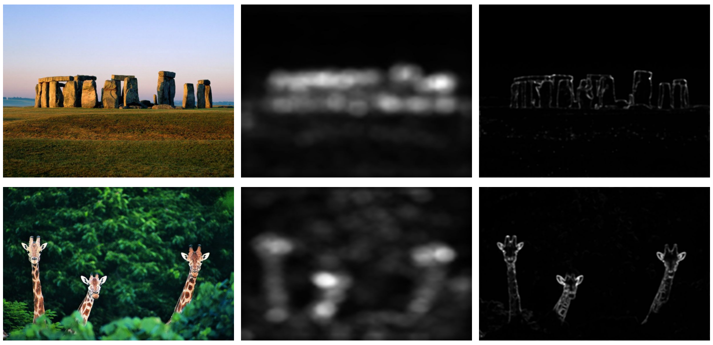


Fig. 6. Our multi-level local saliency results generate a saliency map with fine details, while excluding most of the background. The result of [11] are blurry, fine details are lost, and parts of the textured background appear.

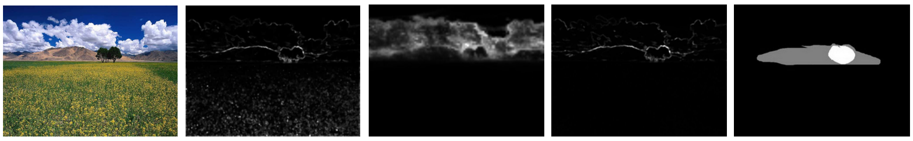


Fig. 7. Result of both stages of our algorithm on a highly textured image obtained from the database of [5]. The multi-level local saliency detects details on both the tree and mountains as well as the flower field. The regional analysis detects the mountains region. Our final result provides detailed edges of the trees and the mountains corresponding to the human labeling.

is background texture with strong edges that appear in all detail levels, as shown in Fig. 7(b). Here, part of the flower field appears as salient, despite the fact that it spreads almost over the entire image. This is since the flowers are locally salient. To exclude them and obtain the salient structures only we further need a global saliency measure.

2.4 Regional Saliency

In this section we propose a regional saliency measure which detects regions that are salient with respect to their surrounding. We propose using regional saliency rather than global saliency since requiring global saliency is often too restrictive. For example, see Fig. 6. Requiring global saliency as was proposed in [6] and [7], could result in marking the three giraffes as non salient since neither is unique. Requiring regional saliency, on the other hand, will detect all of them as salient with respect to their surrounding.

A salient object in the image can be distinguished from its background by color and contrast. The color histogram of a region holds statistical information of the object, and therefore the histogram of a salient object and the histogram of its surroundings are most likely different. In [6] this principle was adopted by randomly selecting rectangular regions and comparing their color histogram with that of their surrounding. This process is greedy since one needs to explore rectangles of all possible aspect ratios, sizes and locations. In addition, salient objects are not necessarily rectangular. Instead, we offer a more efficient approach, which also deals with regions of various shapes and sizes.

We divide the image into a grid of square blocks of size $k_i \times k_i$ pixels. Since we do not know a priori the characteristic scale of the texture or the size of the salient objects, we use a range of block sizes to capture objects at various scales. We start by describing how we determine the saliency for a single block size. Later on, we will describe how the saliency results obtained for different block sizes are merged.

An image block is considered salient if its appearance is highly different from its surrounding blocks. To capture this, we compute for each block the RGB color histogram with 48 bins (16 for each color). We compare RGB histograms of two blocks using χ^2 distance. We compare each block in the image with all the blocks in a neighborhood of $\pm m$ blocks around it. The similarity between far away blocks is less important, hence we penalize the color difference with the distance in position as follows:

$$d_H(B_k, B_l) = \chi^2(B_k, B_l)\left(1 + C\sqrt{\Delta x^2 + \Delta y^2}\right), \tag{9}$$

where Δx and Δy are the horizonal and vertical distances between the two blocks, normalized by the number of columns and rows, respectively. C is a constant set to 3. For each block we find the M most similar blocks within a neighborhood of $\pm m$ blocks around it, i.e., the blocks with the lowest d_H. The list of most similar neighbors of B_k is denoted by $B_k^{Neighbor}$. We then define the regional saliency of block B_k as:

$$S_r(B_k) = 1 - exp\left\{-\frac{1}{M}\sum d_H(B_k, B_l)\right\} \text{ s.t. } B_l \in B_k^{Neighbor}. \tag{10}$$

This means that blocks which have few or no neighbors with similar color histograms are defined as salient and get a value close to 1. Blocks which have more neighbors with the same color histogram, will get a lower value closer to 0. M is defined to be $1/6$ of the number of surrounding blocks in a region of $\pm m$, and m is set to 3.

Since S_r defines one saliency value for the entire block, we actually get a saliency map which is smaller by a factor of k_i^2 than the true image size, where k_i is the size of the block. We use a bilinear filter in order to re-scale the regional saliency map to the original image size, obtaining a map that is smooth rather than blocky:

$$\tilde{S_r} = F_s * S_r, \tag{11}$$

where F_s is a bilinear filter that scales the regional saliency map to the original image size. Even though this process is equivalent to scaling the image, we do not use a single resolution, but combine all scales together into one regional saliency map. In addition, we later combine this regional map with the fine detailed local map, therefore fine details are not lost.

Multi Level Regional Saliency To make no prior assumptions on object sizes and scales, we compute regional saliency for several sizes of blocks: $k_i = [8, 16, 32, 64, 128]$. For each block size we generate a regional saliency map $S_r(i)$. As mentioned above, all the saliency maps are re-scaled to the original image size. If a region is part of a texture of any size it is not salient therefore its saliency value should be low. A region is to be considered as salient if it is not part of a texture of any size in the image. To achieve this, the geometric mean is used in order to average the regional saliency maps:

$$S_{regional} = \left(\prod_{i=1}^{K} \tilde{S}_r(i) \right)^{\frac{1}{K}} \tag{12}$$

By using this function, only areas which have a high regional saliency value in all levels are considered as salient. Fig. 7(c) shows the regional saliency of the flower field image. Some of these flowers appear as salient in the local saliency map, but they are completely removed from the regional saliency map.

2.5 Combining Regional and Local Saliency

As a final step, we wish to combine the local saliency map S_{local}, which contains information regarding the important edges in the image, with the regional saliency map $S_{regional}$, which contains information on which of the regions are salient with respect to their neighborhood. Combining the two, we get a map which features the important edges that define the salient objects, without containing the edges of the textured areas. We define salient points as points which obtain a high value of S_{local} for residing on an important edge and obtain a high value of $S_{regional}$ for residing in a salient region. Therefore, we define the total saliency map as the product of the square regional saliency, and the local saliency :

$$S_T = S_{local} S_{regional}^2 \tag{13}$$

We square $S_{regional}$ since the regional saliency should have more impact on the total saliency. We then stretch S_T so that the lowest value is set to 0, and the highest value is set to 1. Fig. 7(d) shows the combinations of both maps of the flower field image. The edges of the salient structures are best depicted by the local map but some edges of the textured are included as well. The regional map has no edge information but successfully removes textured areas. Together, they create a detailed map which demonstrates only the edges of salient structures.

2.6 Computational Complexity

Consider an image resolution of $M \times N$ pixels. We first compute $n = 4$ levels of detail by applying the anisotropic diffusion of [10] with complexity O(MN). We then apply the phase transform which adds O(MN) for each level, i.e., a total of O(5MN). To obtain regional saliency we compute at most MN/64 histograms each of which is compared with at most 48 neighbors. Since we have 5 scales for the regional saliency this sums up to O(5MN). Putting together the local and regional saliency we still get an

O(MN) algorithm. The entire process, including both stages of the algorithm, takes a few seconds using Matlab for images of size 1M pixels. The generation of the different levels of detail using Beltrami or bilateral filter [13], takes a few seconds using Matlab, but can be performed in real time with better implementation. Therefore, we have an efficient and fast algorithm. In the next sections we show that although being fast, this algorithm still outperforms previous work.

3 Applications: Image Abstraction

There are many applications of saliency where saliency is a required input, including image re-targeting, video encoding, surveillance applications, smart sharpening and others. Our algorithm generates an image depicting the edges of the salient objects, hence, it could be beneficial in applications that use these edges. One such application is cartoonization. In [14], a method for video abstraction is presented, where the image is abstracted into a cartoon like image using anisotropic diffusion and color quantization. Important edges are emphasized using a threshold and overlayed on the smoothed image. This allows a simplified image where important structures are emphasized. However, highly textured images emphasize the texture as well as the salient structures in the image, resulting in a clatter of black lines, as can be seen in Figure 8(b). The building on the top figure and the grass at the bottom figure are both overly marked with black outlines. The code and images for this application were provided by [15].

Our salient edges detection algorithm can improve the abstraction results. This is achieved by taking the product of the saliency map of Eq.(13) and the edge map computed using [15]. Therefore, edges of structures which are not salient will not be emphasized in the cartoon image. We demonstrate our results in figure 8(c). When using our saliency map, mostly the borders of the salient objects are emphasized in black, and not texture such as bricks or grass.

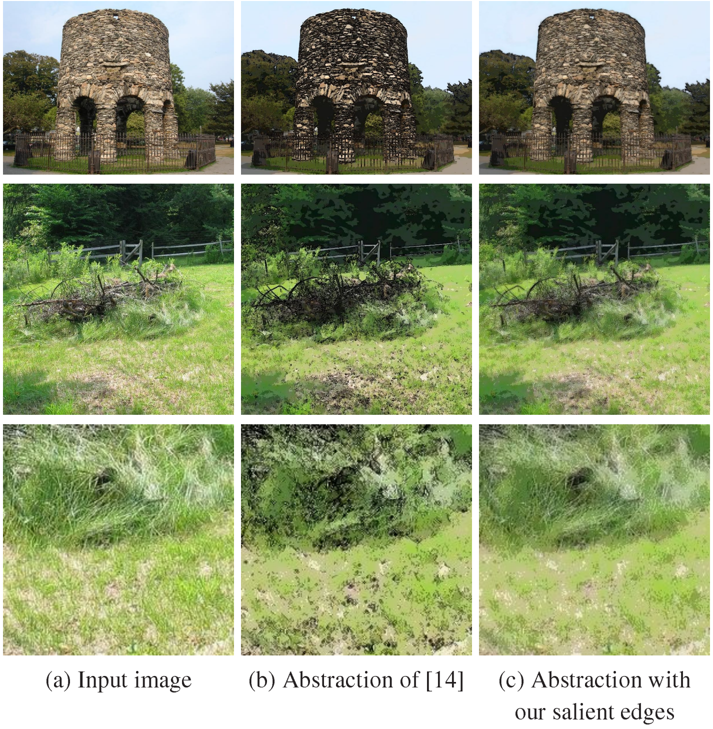
4 Experimental results

4.1 Qualitative evaluation

Fig. 2 and 9 illustrate visually the difference between our approach and the approach of [9] and [11]. The algorithm of [11] and its multi-scale extension [9] show a blurry saliency map, which sometimes detects parts of the background as salient e.g. the giraffes image, the Chinese wall image in Fig. 9, and the cows image in Fig. 2. Our results, on the other hand, detect only the edges of salient objects. These images demonstrate the importance of both parts of our algorithm. Using several levels of detail without reducing scale helps in finding the most important edges in the image. Analyzing the color histogram of a region of varying sizes with respect to its neighborhood, allows to extract only the edges of salient objects, removing edges of textured background.

4.2 Quantitative Evaluation

Edges vs. Regions Our algorithm produces a map of salient edges, which are useful for several applications, such as selective sharpening and video abstraction which we


(a) Input image

(b) Abstraction of [14]

(c) Abstraction with our salient edges

Fig. 8. Abstraction with and without saliency. (b) The abstraction of [14] detects too many edges in textured areas, hence the building in the top image and the branches in the middle are cluttered with black lines. (c) Incorporating our salient edges detection improves the abstraction results and highlights in black only the outlines of the salient objects. The bottom image zooms in on part of the middle image demonstrating the in [14] black edges are shown in textured background regions.

demonstrate in section 3. Other applications, require complete region detection. In order to obtain regions rather than edges we suggest to further add a smoothing step to Eq.(13). We apply Gaussian smoothing to S_T with a Gaussian filter 49 pixels wide with $\sigma = 16$. This is in line with the observation of [8], where it was shown that finding the foci of attention and smoothing around them provides saliency maps which correspond well with human eye tracking results. In the next section we compare our results to eye fixation data of [16]. Since the eye fixation data is extremely smooth, we use the smooth version of our saliency map.

Quantitative Results We present quantitative evaluation on two different databases. The first was presented by Bruce and Tsotsos [16] that generated eye fixation data for a set of 120 color images viewed by 20 subjects. The second database was presented in [5], where 4 subjects hand labeled in 62 images the regions they perceived as most salient. In the labeled map, the white region represents salient region, the black region represents the non-salient region, and the gray region was selected by some labelers but

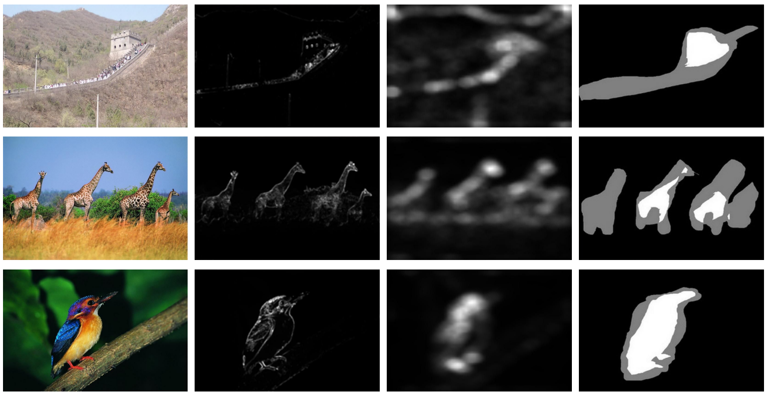


Fig. 9. Saliency results of our algorithm.

rejected by others. Based on [17] we compare performance using the area under the Receiver Operating Characteristic (ROC) curve. The closer it is to 1 the better.

We compare our algorithm, to our implementation of Guo et al. [11] (that showed their results are almost identical to [5]), to Bruce and Tsotsos [1] using their AIM software, to the results of Itti et al. [2], and to the results of the multi-scale [9]. The results of [2] were obtained from [16] and [5] for each database respectively. Since [9] do not supply code, their AUC value was obtained from their paper. In all our experiments we used $n = 4$ levels of details for calculating S_{local} and 5 block sizes $k_i = 8, 16, 32, 64, 128$ for calculating $S_{regional}$. In order to calculate the area under the ROC curve we used Harel's utility function [3]. There, the value of the density maps of the fixation data is used as a weight for each true positive. For each algorithm we calculated AUC (area under ROC curve) for each image. Table 1 shows the average value for AUC obtained by each algorithm over all images. The average value of our AUC is the best in one database and close to the best value of [9] in the second database. While we get a similar value as the best using our smooth maps, we also provide a detailed map of salient edges. This is not demonstrated by any other saliency method.

Table 1. ROC areas for different saliency models with respect to all human eye fixation or human labeling in the data sets of [16] and [5].

Data Set	**Itti [2]**	**AIM [1]**	**Guo[11]**	**Guo MS[9]**	**Ours**
[16]	0.729	0.774	0.77	0.824	**0.817**
[5]	0.832	0.804	0.8408	–	**0.8718**

5 Conclusion and Future work

This paper presented a novel method for saliency detection which integrates local and regional saliency at multiple scales. The proposed method does not favor one level over another as done in hierarchical multi-scale methods. All levels are used together in order to generate a saliency map which shows the edges of salient objects. Maintaining a low computational cost we present a robust method which works well even on highly textured images. We present quantitative and visual results, and show the benefits of our method for image abstraction. In future work we intend to extend our method to video, and explore how to incorporate motion into the generation of multi-levels of detail and saliency images.

References

1. Bruce, N.D.B., Tsotsos, J.K.: Saliency, attention, and visual search: An information theoretic approach. J. Vis. **9** (2009) 1–24
2. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Machine Intell. **20** (1998) 1254–1259
3. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In Schölkopf, B., Platt, J., Hoffman, T., eds.: Advances in Neural Information Processing Systems 19. MIT Press, Cambridge, MA (2007) 545–552
4. Gao, D., Mahadevan, V., Vasconcelos, N.: On the plausibility of the discriminant center-surround hypothesis for visual saliency. J. Vis. **8** (2008) 1–18
5. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: CVPR. (2007) 1–8
6. Liu, T., Sun, J., Zheng, N., Tang, X., Shum, H.Y.: Learning to detect a salient object. In: CVPR. (2007)
7. Goferman, S., Zelnik-Manor, L., , Tal, A.: Context-aware saliency detection. In: CVPR 2010. (2010)
8. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: IEEE International Conference on Computer Vision (ICCV). (2009)
9. Guo, C., Zhang, L.: A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. IEEE Transactions on Image Processing **19** (2010) 185 – 198
10. Spira, A., Kimmel, R., Sochen, N.: Efficient beltrami flow using a short time kernel. In: Proc. of Scale Space 2003, Lecture Notes in Computer Science (vol. 2695), Isle of Skye, Scotland, UK (2003) 511–522
11. Guo, C., Ma, Q., Zhang, L.: Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: CVPR. (2008)
12. Ell, T., Sangwin, S.: Hypercomplex fourier transforms of color images. IEEE Transactions on Image Processing **16** (2007) 22–35
13. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the IEEE International Conference on Computer Vision. (1998)
14. Winnemöller, H., Olsen, S.C., Gooch, B.: Real-time video abstraction. ACM Trans. Graph. **25** (2006) 1221–1226
15. (Lanman, D.) http://www.mathworks.com/matlabcentral/fileexchange/12191.
16. Bruce, N., Tsotsos, J.: Saliency based on information maximization. In Weiss, Y., Schölkopf, B., Platt, J., eds.: Advances in Neural Information Processing Systems 18, Cambridge, MA, MIT Press (2006) 155–162
17. Tatler, B.W., Baddeley, R.J., Gilchrist, I.D.: Visual correlates of fixation selection: effects of scale and time. Vision Research **45** (2005) 643–659