Proceedings of the 18th IEEE International Conference on Image Processing 2011,
Brussels, Belgium, 11-14 September 2011, Pages 1701-1704

REAL-TIME VISUAL SALIENCY BY DIVISION OF GAUSSIANS

Ioannis Katramados[1,2] *and Toby Breckon*[1]

[1]School of Engineering, Cranfield University, UK
[2]TRW Conekt, UK
{i.katramados, toby.breckon}@cranfield.ac.uk

ABSTRACT

This paper introduces a novel method for deriving visual saliency maps in real-time without compromising the quality of the output. This is achieved by replacing the computationally expensive centre-surround filters with a simpler mathematical model named Division of Gaussians (*DIVoG*). The results are compared to five other approaches, demonstrating at least six times faster execution than the current state-of-the-art whilst maintaining high detection accuracy. Given the multitude of computer vision applications that make use of visual saliency algorithms such a reduction in computational complexity is essential for improving their real-time performance.

Index Terms— division of gaussians, DIVoG, salient features, center-surround, ratiometric saliency

1. INTRODUCTION

As a concept, visual saliency started as a biologically inspired process for focusing visual attention to certain parts of an image, thus reducing the complexity of scene analysis [1]. Subsequently, it formed the basis of several computer vision applications, such as in automatic object detection [2, 3, 4, 5], medical imaging [6] and robotics [7]. Different saliency definitions exist, however, in this paper a generalised version of the definition by Achanta *et al.* [8] is used: "*Visual saliency is the perceptual quality that makes a group of pixels stand out relative to its neighbours*". As a research topic, visual saliency theory has evolved rapidly to produce a wide range of approaches. However, their computational cost remains significantly high for real-time applications that require execution at full frame rate ($\geqslant$ 25 frames per second (fps)). This paper proposes a fast alternative to calculating visual saliency maps by using Division of Gaussians (*DIVoG*), which delivers a multifold increase in performance when compared to the current state-of-the-art.

This research has been supported by the Engineering and Physical Sciences Research Council (EPSRC, CASE/CNA/07/85) and TRW Conekt.

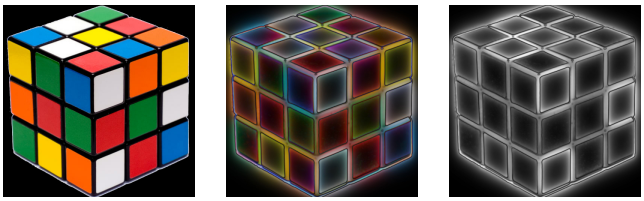

Fig. 1. Colour and greyscale saliency maps of Rubik's cube using the *DIVoG* approach. Darker colours/shades indicate areas of low-saliency and *vice-versa*.

2. EXISTING APPROACHES

Most of the visual saliency models can be categorised into two main groups, as proposed by Achanta *et al.* [9] and Ngau *et al.*[10]: a) biological models and b) computational models. The majority of biological models are using a bottom-up approach for feature extraction mainly based on colour, intensity and orientation [11]. Inspired by the structure of the human eye, this approach detects the contrast difference between an image region and its surroundings, which is also known as centre-surround contrast. Itti *et al.* [11] use the Difference-of-Gaussians (*DoG*) filter for deriving the centre-surround contrast, whereas Walther and Koch [12] take this algorithm further by adopting the concept of salient proto-objects. A common characteristic of these approaches is that they usually produce saliency maps that lack sharpness and detail [5]. Furthermore, the complexity of the biological models means that performance is slow, thus they are more suitable for use in non-real-time applications. One of the few exceptions is found in the approach proposed by Ma and Zhang [13], who calculate the centre-surround contrast by fuzzy growing. The computation takes approximately 60 milliseconds for a 320×240 image on a 2.6 Ghz CPU [8], which corresponds to 16.6 fps.

Examples of computational saliency methods include frequency-tuned salient region detection by Achanta *et al.* [8], graph-based visual saliency by Harel *et al.* [14], affine invariant salient region detection by Kadir *et al.* [15] and real-time visual attention system using integral images by Frintrop *et al.* [16]. The method by Frintrop *et al.* [16], is one

Fig. 2. Saliency map of a pedestrian using *DIVoG*.

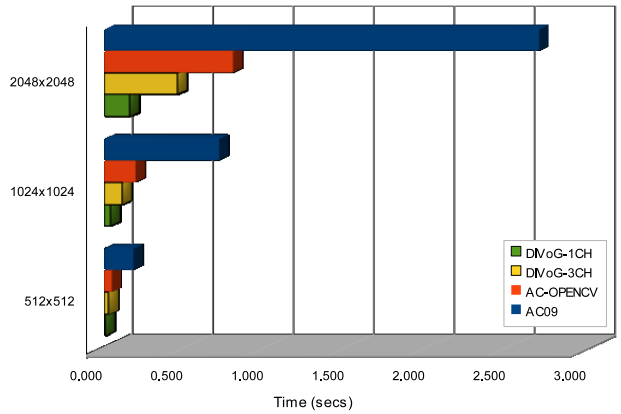

Fig. 3. Performance evaluation of *DIVoG* and "Frequency-tuned Salient Region Detection" by Achanta *et al*. [8] (*AC09*). *AC-OPENCV* is our *AC09* real-time implementation using the *OpenCV* library [18]. *DIVoG-3CH* denotes the *DIVoG* algorithm running on 3 channel input (i.e. RGB image), whereas *DIVoG-1CH* denotes the *DIVoG* algorithm running on a single channel input (i.e. greyscale 8-bit image).

of the most successful attempts to produce a real-time visual saliency algorithm (known as VOCUS) using integral images to reduce execution time. The improvement in performance is impressive with a 400×300 image being processed in approximately 50 milliseconds using a 2.8 Ghz CPU, which corresponds to 20 fps. In addition, the approach proposed by Achanta *et al*. [8] comes close to achieving real-time performance by using frequency domain analysis to produce full resolution saliency maps. The execution time for a 400×300 image is 100 milliseconds on a 2.4 Ghz notebook. Although, this algorithm is proportionally slower than Frintrop *et al*. [16], it generates maps with significantly higher quality.

Ultimately, the target of our algorithm was to produce saliency maps of similar quality to those by Achanta *et al*. [8, 17] at full frame rate ($\geqslant$ 25 fps). In fact, we will show that for a 400×300 image the *DIVoG* approach generates high-detail saliency maps at 50 fps (20 milliseconds per frame) using a 2.4 Ghz CPU.

3. ALGORITHM DESCRIPTION

The Division of Gaussians approach comprises of three distinct steps: 1) Bottom-up construction of Gaussian pyramid, 2) Top-down construction of Gaussian pyramid based on the output of *Step 1*, 3) Element-by element division of the input image with the output of *Step 2*.

Step 1: The Gaussian pyramid U comprises of n levels, starting with an image U_1 as the base with resolution $w \times h$. Higher pyramid levels are derived via downsampling using a 5×5 Gaussian filter. The top pyramid level has a resolution of $(w/2^{n-1}) \times (h/2^{n-1})$. Let us call this image U_n.

Step 2: U_n is used as the top level D_n of a second Gaussian pyramid D in order to derive its base D_1. In this case, lower pyramid levels are derived via upsampling using a 5×5 Gaussian filter.

Step 3: Element-by-element division of $U_1$and D_1 is performed in order to derive the minimum ratio matrix M (also called MiR matrix) of their corresponding values as described by the following equation:

$$M_{i,j} = min\left(\frac{D_{1_{i,j}}}{U_{1_{i,j}}}, \frac{U_{1_{i,j}}}{D_{1_{i,j}}}\right) \quad (1)$$

The saliency map S is then given by *equation 2*, which means that saliency is expressed as a floating-point number in the range $0 - 1$.

$$S_{i,j} = 1 - M_{i,j} \quad (2)$$

The described approach can be further expanded to include element-by-element division of all corresponding levels of pyramids U and D. In this case, the MiR matrix is initialised as a unit matrix (i.e. for each matrix element $M_{0_{i,j}} = 1$). Then each pair of pyramid levels U_n and D_n is scaled up to the input's resolution. Then the MiR matrix M_n is multiplied by M_{n-1} as described by the *DIVoG* equation below, which is a generalised form of *equation 1*.

$$M_{n_{i,j}} = min\left(\frac{D_{n_{i,j}}}{U_{1_{i,j}}}, \frac{U_{1_{i,j}}}{D_{n_{i,j}}}\right) M_{n-1_{i,j}} \quad (3)$$

for $n \geqslant 1$. The saliency map is then derived using *equation 2*. Deriving the MiR matrix through processing of all pyramid levels produces more accurate saliency maps than *equation 1*, but also increases the computational complexity of the algorithm. In practice, the difference between the two approaches is visually minimal, thus in this paper all MiR matrices have been calculated using *equation 1*. Finally, a major advantage of this approach is that it is colourspace-independent, thus it can derive saliency maps even from greyscale images, which significantly reduces computational cost.

Implementation notes: a) All operations are performed using 32-bit floating point matrices. b) To avoid division by zero, or division with floating point numbers in the range 0 to 1, we define the minimum pixel value equal to k^n, where k is the size of the Gaussian kernel. This ensures that pyramidal

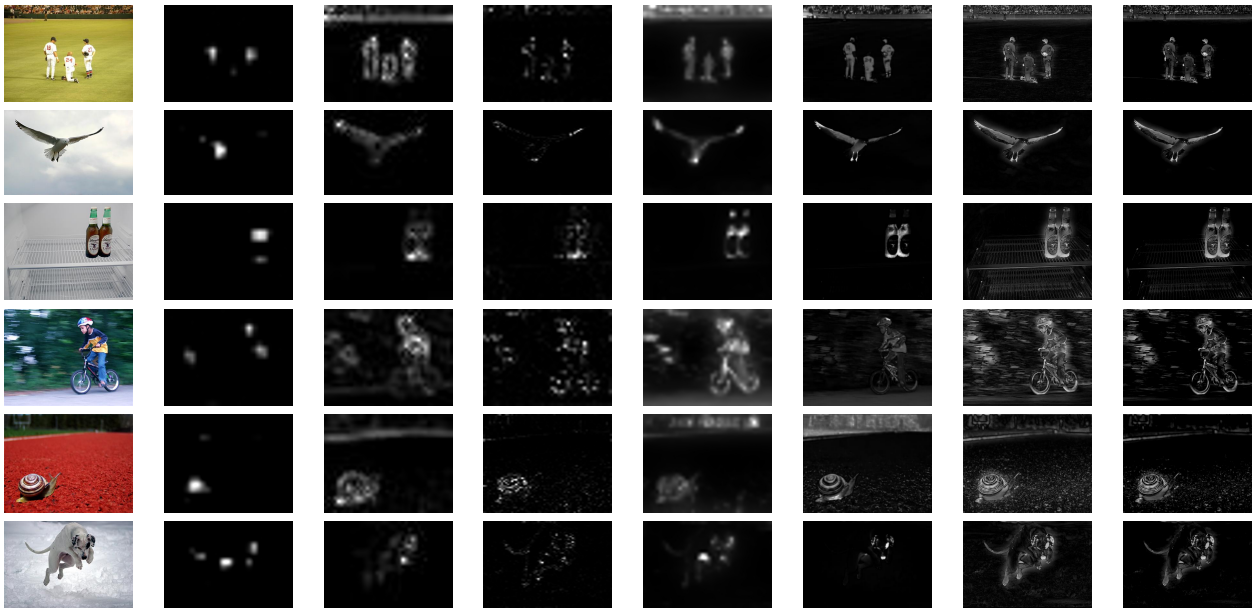

Fig. 4. A set of saliency maps generated using different approaches (based on work by Achanta *et al*. [8]). DIVoG-F enhances these results of the standard DIVoG algorithm by adding a low-pass filter to reduce background noise.

downsampling will always result into a value greater than 1. c) For colour images, the algorithm can be used with any colourspace. Each channel is processed separately to produce a salience map. d) All the saliency maps in this paper have been produced using 24-bit colour images in the RGB colourspace. The Gaussian pyramid is constructed with $n = 5$. e) All saliency maps in Fig. 1, 2, 4, have been normalised to fit the $0 - 255$ range.

4. RESULTS

The *DIVoG* approach is compared with five other saliency algorithms using an evaluation framework created by Achanta *et al [8, 17].* As part of this procedure, saliency maps are extracted for 1000 images using five different approaches [11, 13, 19, 14, 8], as illustrated in Fig. 4. These maps are then used to segment the images. Finally, the extracted segments are compared to the ground-truth in order to derive the algorithm's accuracy. This is a reasonable approach for simple scenes with a small number of distinct objects. However, for more complex images the specification of ground-truth is becoming subjective. Since the main contribution of this paper is related to the real-time performance of the algorithm, we compare the execution time of our approach with Achanta *et al. [8]*, which is one of the most efficient saliency methodologies for producing high-resolution maps.

For performance evaluation a mobile 2.4GHz Intel Core 2 Duo processor was used with 4GB RAM. Fig. 3 and Table 1 show a comparison in execution time between *DIVoG* and *[8]* at different resolutions using colour and greyscale images. Furthermore, Fig. 4 shows some examples of saliency maps generated using DIVoG and five other approaches.

The original implementation by Achanta *et al [8]* (*AC09*), produces much sharper saliency maps than *IT98, MA03, HO07* and *HA07*. In terms of computational performance *AC09* is at least comparable to the aforementioned approaches as presented in *[8].* On the other hand, the *DIVoG* approach demonstrates similar or higher quality saliency maps to *AC09*, but at a fraction of the time. *DIVoG* is faster than *AC09* by a factor of 6 when processing 24-bit colour images and by a factor of 16 when processing greyscale images. This massive gap could not be justified by the theoretical difference in computational complexity, thus the *AC09* was re-implemented using the *OpenCV* library [18] (*AC-OPENCV*). This way the execution time reduced by a factor of 3. Even so, *AC-OPENCV* remained 56% slower than *DIVoG*. An indication of performance can also be given by quoting the achieved framerate. At the lowest resolution of 320×240, *DIVoG* executed at 333 fps on greyscale images and 111 fps on colour images, showing a linear relationship between data size and execution time. Overall, the *DIVoG* approach has demonstrated an ability to calculate full resolution saliency maps with the minimum computational cost.

5. CONCLUSIONS

We presented a novel visual saliency algorithm for calculating full resolution saliency maps in real-time by using Divi-

	***AC09** [8]*		***AC-OPENCV***	
Resolution	*Time (s)*	*fps*	*Time (s)*	*fps*
320×240	0.078	12.8	0.015	66.6
512×512	0.187	5.3	0.052	19.2
640×480	0.218	4.6	0.057	17.5
1024×1024	0.718	1.4	0.200	5.0
2048×2048	2.699	0.4	0.803	1.6
	DIVoG-3CH		***DIVoG- 1CH***	
320×240	0.009	**111**	0.003	**333**
512×512	0.032	**31.2**	0.009	**111**
640×480	0.036	**27.7**	0.012	**83.3**
1024×1024	0.115	**8.7**	0.041	**24.3**
2048×2048	0.456	**2.2**	0.161	**6.2**

Table 1. Performance evaluation data showing execution time and framerate. *AC09* is the original implementation by Achanta *et al*. [8].

sion of Gaussians. Compared to recent work by Achanta *et al*. [8], *DIVoG* showed a significant increase in performance by a factor of 6 when using colour images. This paper also introduced a real-time implementation of Achanta's work using the OpenCV library [18], which is more than three times faster than the original implementation, but still 56% slower than the *DIVoG* approach. Given that for VGA resolution the achieved framerate exceeds 80 fps on greyscale images, this algorithm could significantly improve the performance of a wide range of applications including salient feature detection, object extraction and classification.

6. REFERENCES

[1] J.K. Tsotsos, S.M. Culhane, Winky Yan Kei Wai, Yuzhong Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 507 – 545, 1995.

[2] N. Ouerhani and H. Hugli, "Maps: multiscale attention-based presegmentation of color images," in *Proceedings of the 4th International conference on scale space methods in computer vision*, Berlin, Heidelberg, 2003, pp. 537–549, Springer-Verlag.

[3] S. Lee M. Won, W.J. Jeong, "Road traffic sign saliency map model," in *Proceedings of Image and Vision Computing New Zealand*, 2007, pp. 91–96.

[4] J. Sokalski, T.P. Breckon, and I. Cowling, "Automatic salient object detection in uav imagery," in *Proc. 25th International Unmanned Air Vehicle Systems*, April 2010, pp. 11.1–11.12.

[5] Haonan Yu, Jia Li, Yonghong Tian, and Tiejun Huang, "Automatic interesting object extraction from images using complementary saliency maps," in *Proceedings of the international conference on Multimedia*. 2010, pp. 891–894, ACM.

[6] Zhijun Gu and Binjie Qin, "Nonrigid registration of brain tumor resection mr images based on joint saliency map and keypoint clustering," *Sensors*, vol. 9, no. 12, pp. 10270–10290, 2009.

[7] N.J. Butko, Lingyun Z., G.W. Cottrell, and J.R. Movellan, "Visual saliency model for robot cameras," in *IEEE ICRA*, May 2008, pp. 2398 –2403.

[8] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *IEEE CVPR*, 2009, pp. 1597 –1604.

[9] R. Achanta, F. Estrada, P. Wils, and Sabine Süsstrunk, "Salient region detection and segmentation," in *Computer Vision Systems*, vol. 5008 of *LNCS*, pp. 66–75. Springer Berlin / Heidelberg, 2008.

[10] C.W.H. Ngau, L.M. Ang, and K.P. Seng, "Bottom-up visual saliency map using wavelet transform domain," in *IEEE ICCSIT*, 2010, vol. 1, pp. 692 –695.

[11] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE PAMI*, vol. 20, no. 11, pp. 1254 –1259, Nov. 1998.

[12] D. Walther and D. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395 – 1407, 2006.

[13] Yu-Fei Ma and Hong-Jiang Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *Proceedings of the 11th ACM international conference on Multimedia*. 2003, pp. 374–381, ACM.

[14] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," *Advances in Neural Information Processing Systems*, vol. 19, pp. 545–552, 2007.

[15] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," in *ECCV*, vol. 3021 of *LNCS*, pp. 228–241. Springer Berlin / Heidelberg, 2004.

[16] S. Frintrop, M. Klodt, and E. Rome, "A real-time visual attention system using integral images," *ICVS*, 2007.

[17] R. Achanta and S. Süsstrunk, "Saliency detection using maximum symmetric surround," in *IEEE ICIP*, 2010, pp. 2653 –2656.

[18] OpenCV, "Open computer vision library," *http://opencv.willowgarage.com/wiki/*, Last visited on 10/01/2011.

[19] Xiaodi Hou and Liqing Zhang, "Saliency detection: A spectral residual approach," in *IEEE CVPR*, 2007, pp. 1 –8.