RESEARCH ARTICLE | JULY 10 2018

A method for automatic detection of rectangular regions of interest in arbitrary images

Nelson Martins-Ferreira; Lino Ferreira; Paula Pascoal-Faria; Luis A. da Silva Cruz; Pedro Assuncao; Nuno Alves

AIP Conf. Proc. 1978, 160009 (2018)
https://doi.org/10.1063/1.5043819

A Method for Automatic Detection of Rectangular Regions of Interest in Arbitrary Images

Nelson Martins-Ferreira[1,4,a)], Lino Ferreira[1,2,3], Paula Pascoal-Faria[1,4], Luis A. da Silva Cruz[2,3], Pedro Assuncao[1,2] and Nuno Alves[1,2]

[1]*Instituto Politécnico de Leiria/ESTG Leiria, Portugal*
[2]*Instituto de Telecomunicações, Portugal*
[3]*Universidade de Coimbra/DEEC Coimbra, Portugal*
[4]*CDRSP-IPLeiria,Centro Empresarial da Marinha Grande, R. de Portugal, 2430-028 M.nha Grande, Portugal*

[a)]Corresponding author: martins.ferreira@ipleiria.pt

Abstract. This paper presents a computational method to extract optimum rectangular Regions of Interest (RoI) in images with an associated saliency map. Although saliency maps provide an individual relevance measure for each pixel, to find the sub-image (i.e., rectangular region) that contains the set of the most relevant pixels requires an optimisation procedure to define the boundaries of the best RoI. This is achieved by the method devised in the paper, by following an approach based on balancing the amount of relevant information that is included and excluded from the RoI. The results show that such method is capable of finding the most relevant rectangular RoI and thus to extract the optimum sub-images according to the relevance measure given by a generic saliency map. Since the method is not tied to any particular type of images, it finds application in quite different fields, such as salient object extraction and processing in industry and surveillance, image compression using attention modelling, biomedical imaging, etc.

INTRODUCTION

Detection of Regions of Interest (RoI) in images and video has been a research field of interest for different types of applications. In general an RoI is an image region with higher relevance than the remaining regions, according to some predefined criterion. For instance, regions that attract higher human visual attention than others may be less compressed than others to improve subjective quality, objects with higher relevance than others in specific application scenarios may be necessary to be identified (e.g. biomedicine, image compression, remote surveillance), etc [1][2]. The first step in automatic RoI detection algorithms is to compute a saliency map representing the relevance of each pixel in the image [3]. In more complex systems, 2D and 3D images/video maybe integrated in a single framework to increase the efficiency of saliency map computation through fusion of relevant features in the two visual domains [4][5]. Then, based on the saliency maps, the second step is to define the best RoI that meets the specific requirements of a target application [6][7][8].

To extract sub-images containing the most relevant visual content of higher resolution images/video, it is necessary to compute the best rectangular RoIs from saliency maps. Since these are defined by irregular shapes and non-uniform intensity values representing the relevance of each pixel, the problem of finding the optimum sub-image (i.e., the best rectangular RoI) is a computational challenge open for further research. This paper addresses this problem by proposing an efficient method capable of finding the boundaries of the best rectangular RoI inside an image of higher resolution. To this aim we define a metric associated to each point in the saliency map, which measures the amount of relevant visual information beyond that point (to include in the RoI) as well as the loss of information in the region to be excluded from RoI, when such point is chosen to be the upper left corner of the RoI. The efficiency of the proposed method is compared with other state-of-the-art methods based on maximisation of the signal energy contained in the RoI [7].

International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2017)
AIP Conf. Proc. 1978, 160009-1–160009-10; https://doi.org/10.1063/1.5043819
Published by AIP Publishing. 978-0-7354-1690-1/$30.00

Proposed Method

The purpose of this method is to obtain a submatrix S of a given matrix A. The original matrix is assumed to have some points of high interest with high intensity and some others of residual interest and lower intensity. In general, we are interested in finding a submatrix that contains the most relevant information with the smallest possible size (i.e., image region). In order to attain such a goal, we have designed the following procedure:

1. Re-sample the original matrix A and obtain a 100 by 100 matrix, which we continue to call A;
2. Assign to each entry of the re-sampled matrix a value which measures the likelihood that that entry has for being an upper left corner or a lower right corner for the submatrix; this is achieved using a formula that weights the amount of information, which is enclosed in the submatrix and the one that it is left outside of it (further details are given below); the metric $d_{UL}(x, y)$ measures the likelihood that the point $(x, y) \in \{1, \ldots, 100\} \times \{1, \ldots, 100\}$ has of being a potential upper left corner, while $d_{LR}(x, y)$ does the same but for a lower right corner,

$$d_{UL}(x,y) = \sum_{(i,j)\in D_1(x,y)} \frac{|A(x,y)|}{r} - \sum_{(i,j)\notin D_1(x,y)} |A(x,y)|e^{-r^2}$$

$$d_{LR}(x,y) = \sum_{(i,j)\in D_2(x,y)} \frac{|A(x,y)|}{r} - \sum_{(i,j)\notin D_2(x,y)} |A(x,y)|e^{-r^2}$$

   here $r = \sqrt{(x-i)^2 + (y-j)^2}$, and the two regions $D1$ and $D2$ are defined as

$$D_1(x,y) = \{(i,j) \mid (x,y) < (i,j)\}$$

$$D_2(x,y) = \{(i,j) \mid (i,j) < (x,y)\}$$

   where $(x, y) < (i, j)$ means $x < i$ and $y < j$; it is clear that this task is computationally the most demanding, it takes approximately 20 seconds to run on a desktop computer; for each entry in the re-sampled matrix there is the need to sum over all entries in the matrix, it involves a summation procedure with 10^8 terms, $(100 \times 100)^2$;
3. Normalize the likelihood values in to the interval $]-\infty, 1]$; this normalization gives rise to a linear ordering of the possible threshold values which will be searched between 0 and 1, the most appropriate of it will define the two corners (upper left and lower right) of the desired submatrix, and thus the expected output;
4. For each normalized threshold value $t \in [0, 1]$ compute the associated corners for the submatrix accordingly to the following steps:
   (a) Find all entries $(x, y) \in \{1, \ldots, 100\} \times \{1, \ldots, 100\}$ for which $d_{UL}(x, y) \geq t$;
   (b) Define $ul = (ul_x, ul_y)$ as the **minimum** of all those entries encountered in the previous item (this defines the upper left corner);
   (c) Similarly, find all entries $(x, y) \in \{1, \ldots, 100\} \times \{1, \ldots, 100\}$ for which $d_{LR}(x, y) \geq t$, and take $lr = (lr_x, lr_y)$ to be the **maximum** of all those entries (this defines the upper left corner);
   (d) For the set of all entries (x, y) obtained in item (a), let us call it $XYUL$ we compute its mean absolute deviation, `MEAN(ABS(XYUL-MEAN(XYUL))`, this measure gives us an idea of the dispersion of the results, more specifically we also compute the min absolute deviation `MEAN(ABS(XYUL-MEAN(XYUL))`, this measure is particular for the upper left corner;
   (e) We repeat the previous step for the values obtained in item (c) and compute, in addition, the max absolute deviation, this is particular to the lower right corner and it gives a measure for its dispersion values;
   (f) Having the candidate values for the upper left (ul_x, ul_y) and lower right (lr_x, lr_y) corner as obtained in items (b) and (c), we can determine the ratio of energy captured in the submatrix, as well as its area in proportion to the entire matrix A, for convenience let us denote by $S(t)$ the submatrix of A which is obtained as $S(t) = A(ul_x : lr_x, ul_y : lr_y)$, this gives us

$$E(t) = \frac{\texttt{Total Energy in } S(t)}{\texttt{Total Energy in } A}$$

$$A(t) = \frac{\texttt{Total Area of } S(t)}{\texttt{Total Area of } A}$$

   where by energy of a matrix we mean the sum in absolute value of all entries in the matrix and by area we mean the number of its elements; this values are used to define a performance value to each threshold and then to select the best possible one;

(g) The best possible threshold is the value $t \in [0, 1]$ for which $E(t)$ is the greatest possible and at the same time the value of $A(t)$ is the smallest possible one; these two goals are contradictory as one can see in Figure 1.

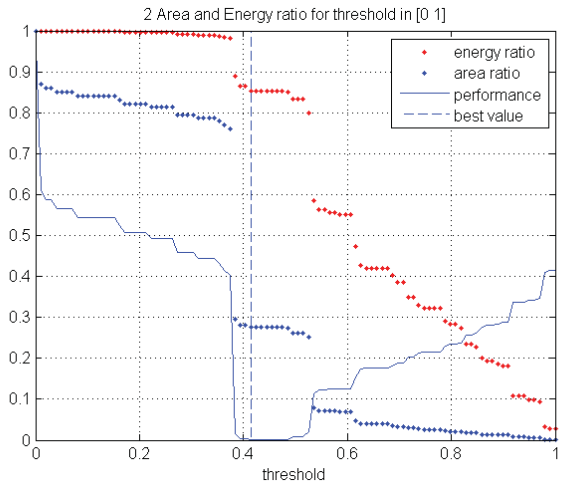


FIGURE 1: Antagonism between energy and area in a concrete example

This observation suggests the need for a suitable performance measure, from which the information on the best possible threshold can be extracted; we have done several experimental tests and the best formula for the evaluation of the performance for a given threshold value $t \in [0, 1]$ turned out to be the following one

$$p(t) = 1 - \frac{1 + \sqrt{E(t)}}{2 - A(t)^2}$$

it is easy to observe that $-1 \leq p(t) \leq \frac{1}{2}$, and that the best values are the ones for which the performance is closer to 0. This means that the best value for the threshold is obtained by taking the value t for which the absolute value of $p(t)$ is minimal;

5. Use the best possible value obtained on item (g) and return the associated values;
6. Another possibility to step (g) is to ask the user to specify a minimum value for the energy ratio $E(t)$, a maximum value to the area ratio $A(t)$ and then find the two values $t_e, t_a \in [0, 1]$ such that $E(t_e)$ is greater of equal than the specified value for the minimum energy value, $A(t_a)$ is less or equal than the prescribed value for the minimum area ratio and take the average value

$$\frac{t_e + t_a}{2}$$

as the best possible threshold value;
7. In addition to the best possible value obtained as any one of the two previous items, the user may also be asked of what threshold it is the most desirable one, having at his disposal all the information that is gather associated to each threshold value.

Several tests have been done and the procedure is robust and has proven to have a good overall performance. Here are some examples of its performance.

Example with 80% enery and 10% area

We have run a battery of tests and here we show the results for an implementation of the previous procedure with an input of four images with a resolution `<2160x3840x4 uint8>` where the user has the option for using the proposed threshold value as obtained in item 6 above, or to change its value. Figures 2 to 5.

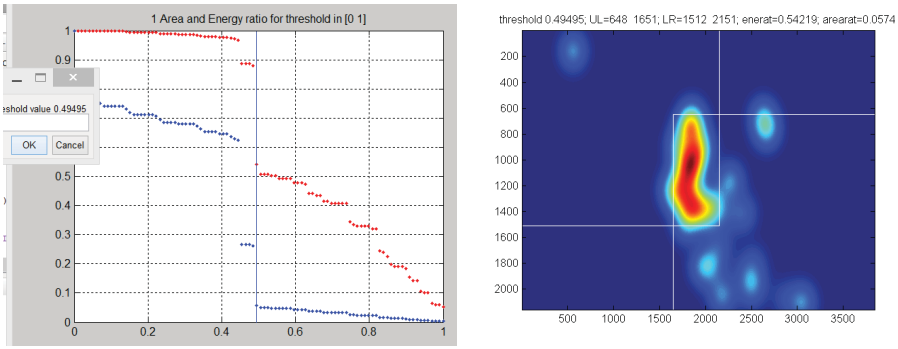


FIGURE 2: Case study 1. Plot of Area and energy for all threshold values in the unit interval on the left, and the result for the best threshold on the right.

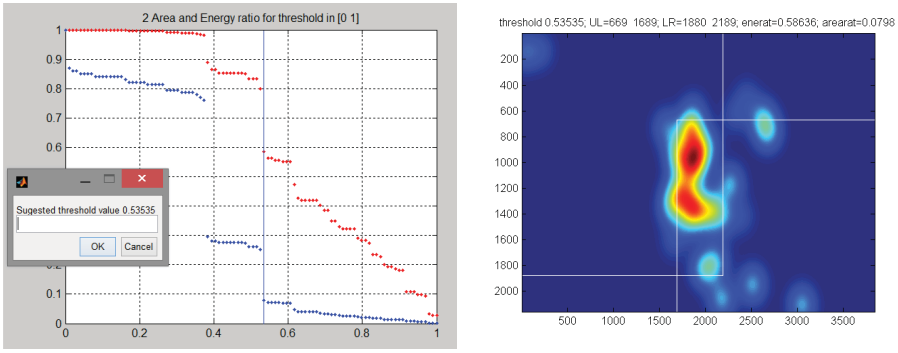


FIGURE 3: Case study 2. Plot of Area and energy for all threshold values in the unit interval on the left, and the result for the best threshold on the right.

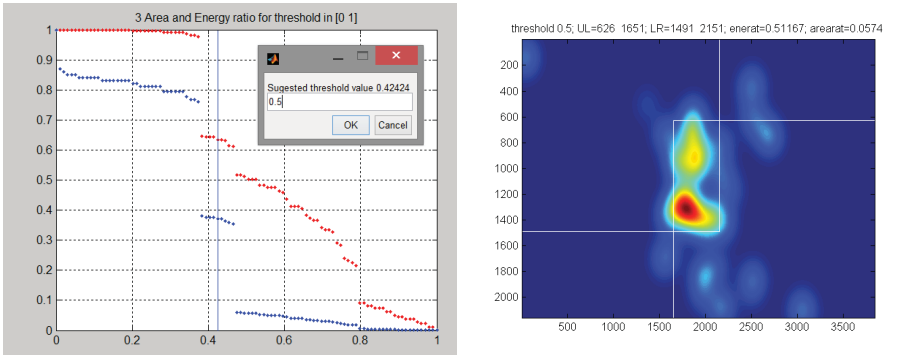


FIGURE 4: Case study 3. Plot of Area and energy for all threshold values in the unit interval on the left, and the result for the best threshold on the right.

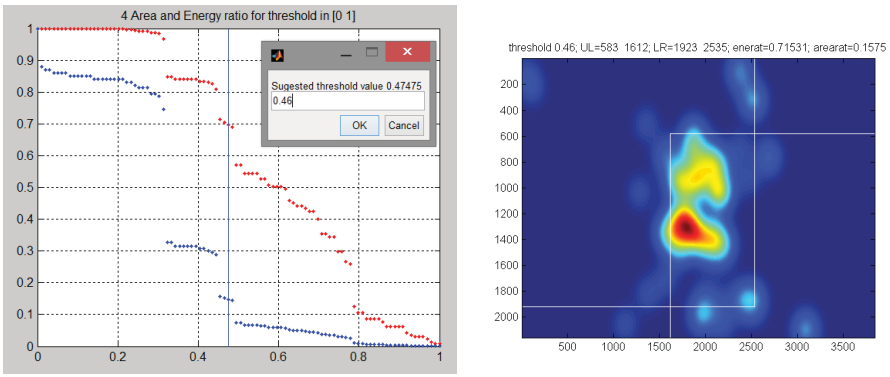


FIGURE 5: Case study 4. Plot of Area and energy for all threshold values in the unit interval on the left, and the result for the best threshold on the right.

Example with best performance threshold

We now run the same battery of tests and show the implementation using the best value for threshold based on the item 5 of our proposed procedure.

On the graphics displayed below, see Figure 6, in addition to the information of energy and area ratios, which was already presented in the previous examples, we now observe the plot for the absolute value of $p(t)$, the performance formula as defined in item 4(g) of the proposed procedure, and the suggested best value which is its minimum value.

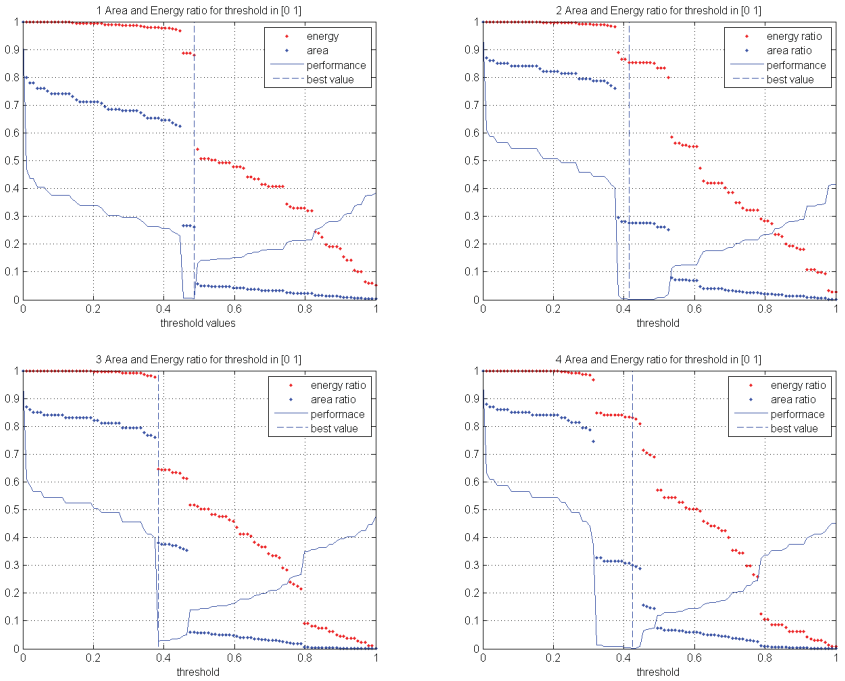


FIGURE 6: Area, energy, performance and best value for the running examples 1 to 4 presented from top to bottom and left to right the

By using the suggested threshold values determined for each picture, from 1 to 4, we observe the submatrix obtained by that value. We have also plotted all the candidate values that were captured above the threshold values, from which the corners are computed, see Figure 7.

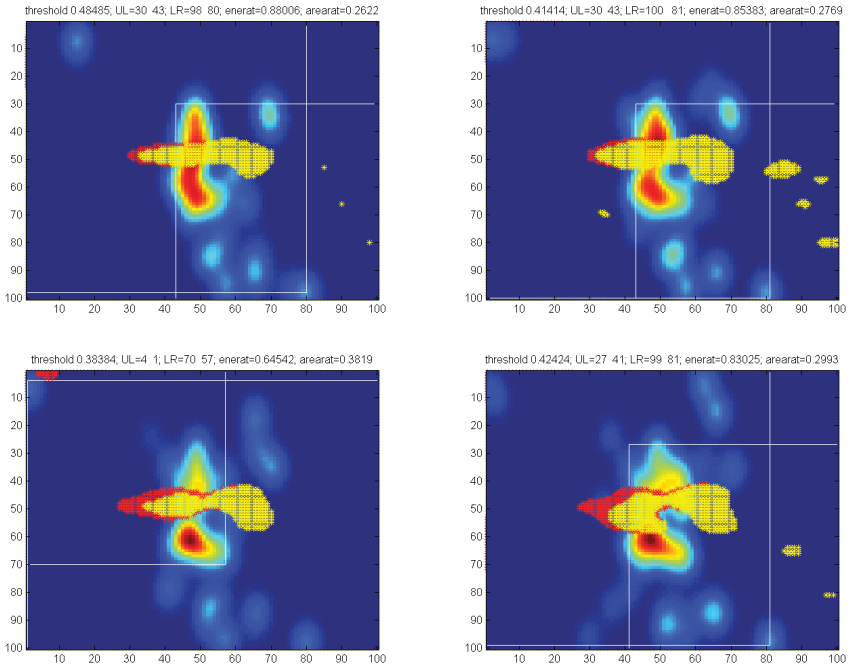


FIGURE 7: Results from the suggested threshold for each one of the running examples 1 to 4 from top to bottom and left to right

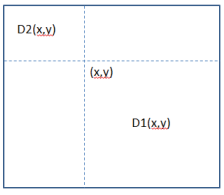
Further details on the explanation of the proposed procedure

The likelihood values associated to each entry in the matrix depend on whether we are computing the upper left corner or the lower right corner. These are computed with two different formulas, which are displayed in item 4(f) of the proposed procedure and where obtained after several experimental tests. Their meaning is the following. Let $A(x, y)$ denote the entry in the position (x, y) of the re-sampled matrix into a resolution 100-by-100. We will use the usual order of the product to relate pairs of indices in the matrix, this means that whenever we write $(x, y) < (i, j)$ we mean $x < i$ and $y < j$. With this notation we define to domain sets, $D_1(x, y)$ and $D_2(x, y)$, for each (x, y) in the grid 100-by-100,

$$D_1(x, y) = \{(i, j) \mid (x, y) < (i, j)\}$$

$$D_2(x, y) = \{(i, j) \mid (i, j) < (x, y)\}$$

these two regions are used to distinguish the two formulas for the likelihood of a point to be an upper left corner or a lower right one, as illustrated.



We observe that D1 is the region associated with the upper left corner and D2 is the region that is associated with the lower right corner. We will also need two functions f_1 and f_2, whose graph is shown in Figure 8, in order to determine

the likelihood of a point (x, y). We will use $r = r(x, y, i, j)$ to represent the euclidean distance between the two points (x, y) and (i, j), so that $r = \sqrt{(x-i)^2 + (y-j)^2}$, and we define

$$f_1(x, y, i, j) = \frac{|A(x, y)|}{r}$$

$$f_2(x, y, i, j) = |A(x, y)|e^{-r^2}$$

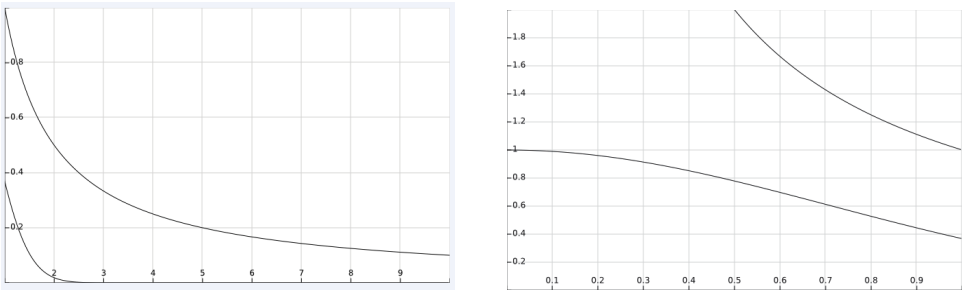


FIGURE 8: Graphical representations of f_1 and f_2 with two different window ranges

We observe that the value of f_1 is always above the value of f_2, while the value of f_2 is defined for values of r closer to zero.

The key idea for assigning a likelihood value for a pair (x, y) and to consider it as an upper left corner is to measure the sum of the values of $f_1(x, y, i, j)$, with (i, j) in the region $D_1(x, y)$ and subtracting the sum of the values of $f_2(x, y, i, j)$ with (i, j) not in D_1. The procedure for finding the likelihood of a point (x, y) to be considered as a lower right corner is obtained in a similar way just replacing the domain set of indexes $D_1(x, y)$ by $D_2(x, y)$.

This gives us a metric $d_{UL}(x, y)$ and $d_{LR}(x, y)$ for the likelihood of a point (x, y) in the grid 100-by-100 to be considered as an upper left corner (UL) or as a lower right corner (LR)

$$d_{UL}(x, y) = \sum_{(i,j)\in D_1(x,y)} f_1(x, y, i, j) - \sum_{(i,j)\notin D_1(x,y)} f_2(x, y, i, j)$$

$$d_{LR}(x, y) = \sum_{(i,j)\in D_2(x,y)} f_1(x, y, i, j) - \sum_{(i,j)\notin D_2(x,y)} f_2(x, y, i, j)$$

We observe that the first step, re-sampling the original matrix into a resolution of 100-by-100, has two immediate consequences: it reduces the complexity of the procedure, indeed step (2) requires the double summing along all elements for each single entry, and it frees the calculations from numerical instability, namely dividing by zero. Note that the values of $r = \sqrt{(x-i)^2 + (y-j)^2}$ are either zero, at $(x, y) = (i, j)$ or are greater than one; when it is zero, it belongs to the complement of the domain D_1 and D_2, hence it is only used in the formula f_2. This proves the consistency of the method.

We now explain the remaining steps on our method.

Step (3) is a simple normalization process, so, we do, for each (x, y) in the 100-by-100 grid, if $d_{UL}(x, y)$ is not zero then we replace it by itself divided by the maximum of all the values d_{UL}. Similarly we normalize d_{LR} and so we obtain, for each (x, y), $d_{UL}(x, y)$ and $d_{LR}(x, y)$ with the relevant values in the unit interval $[0, 1]$. Note that there can be some negative values, but those are of marginal importance.

Step (4) depends on a threshold value. If we let $t \in [0, 1]$ denote the threshold value, then the procedure is completed in the following manner: Find all the entries (u, v) on the 100-by-100 grid whose value d_{UL} is greater or equal to t and compute either the minimum or maximum value of all the entries according to whether we are interested on the upper left corner or in the lower right one, respectively. The specific details are well explained on the substeps at item (4).

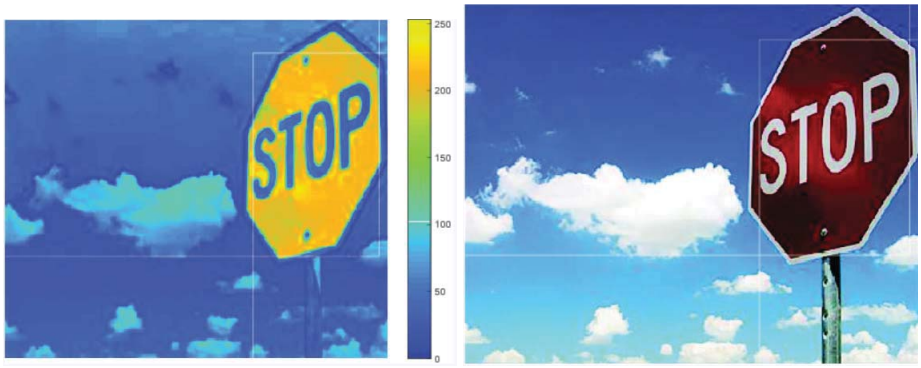


FIGURE 9: **Left:** Visual saliency image with RoI. **Right:** Original Image with RoI - Image 2310 [9]

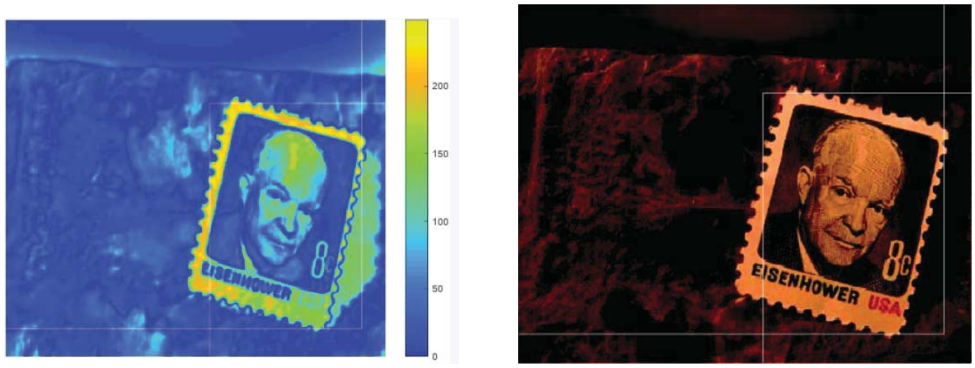


FIGURE 10: **Left:** Visual saliency image with RoI. **Right:** Original Image with RoI - Image 4038 [9]

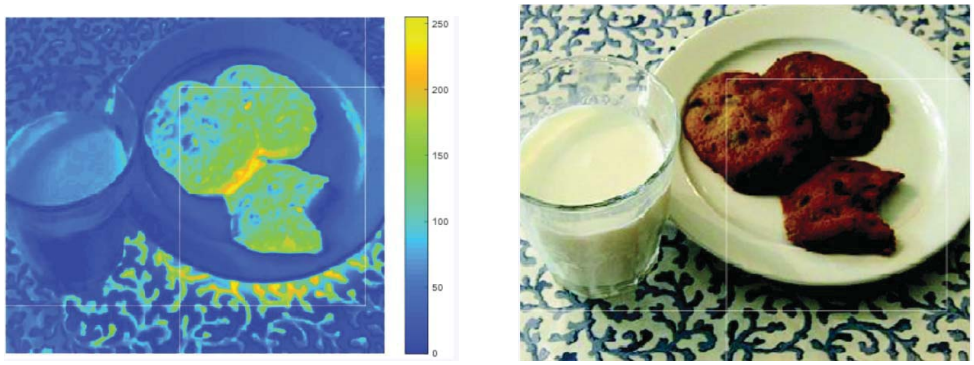


FIGURE 11: **Left:** Visual saliency image with RoI. **Right:** Original Image with RoI - Image 4976 [9]

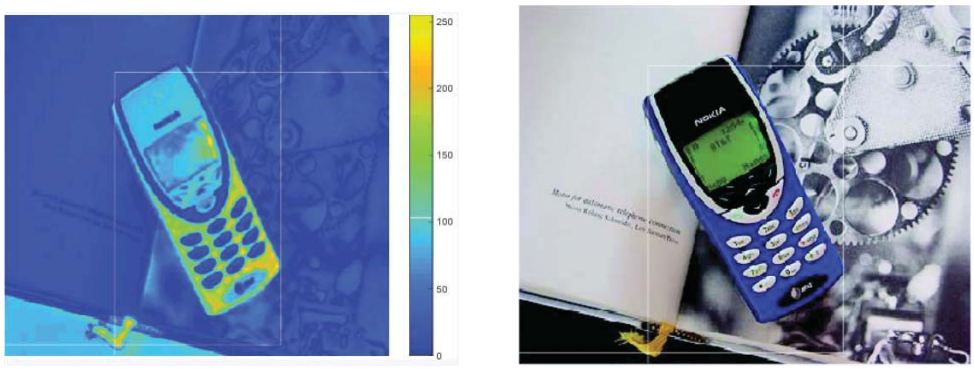


FIGURE 12: **Left:** Visual saliency image with RoI. **Right:** Original Image with RoI - Image 8454 [9]

PERFORMANCE EVALUATION

The proposed method was evaluated by testing its accuracy in selecting salient objects from different images using their saliency maps available from [9]. The results are shown in the following Figures 9, 10, 11 and 12, where the important aspects are: (i) the method effectively identifies the image region where the most important object is located; (ii) by adjusting the energy and the dimensions of the RoI, it is possible to include almost all image area belonging to the salient object; (iii) the method relies on the available saliency map and it is agnostic in regard to the image content.

CONCLUSIONS

This paper presented a method for the automatic detection of rectangular regions, based on saliency maps representing any possible measure of visual relevance, such as human visual attention. The experimental results show that the method proposed in this paper is good enough in identifying the rectangular box where the most salient object is located. This was confirmed by observation of the results obtained from images and saliency maps publicly available. Applications of this method range from computational vision systems, multimedia object identification and recognition and remote surveillance.

ACKNOWLEDGMENTS

Part of this work was supported by R&D Unit UID/EEA/50008/2013 (Project 3DVQM), and PhD Grant SFRH/BD/37510/2007, co-funded by FEDER-PT2020, FCT/MEC, Portugal. This research work was also supported by the Portuguese Foundation for Science and Technology (FCT) through the Project reference UID/ Multi/04044/2013.

REFERENCES

[1] M. Yakno, J. Mohamad-Saleh, and B. A. Rosdi, "New technique for larger RoI extraction of hand vein images," in *2015 International Conference on BioSignal Analysis, Processing and Systems (ICBAPS)* (2015), pp. 82–87.

[2] K. K. Ade and M. V. Raghunadh, "RoI based near lossless hybrid image compression technique," in *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (2015), pp. 1–5.

[3] A. Volokitin, M. Gygli, and X. Boix, "Predicting when saliency maps are accurate and eye fixations consistent," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 544–552.

[4] L. Ferreira, L. A. da Silva Cruz, and P. Assuncao, *A generic framework for optimal 2D/3D key-frame extraction driven by aggregated saliency maps*, Signal Processing: Image Communication **39**, 98 – 110 (2015).

[5] Y. Wo, X. Chen, and G. Han, *A saliency detection model using aggregation degree of color and texture*, Signal Processing: Image Communication **30**, 121–136 (2015).

[6] M. Casares, S. Velipasalar, and A. Pinto, *Light-weight salient foreground detection for embedded smart cameras*, Computer Vision and Image Understanding **114**, 1223 – 1237 (2010), special issue on Embedded Vision.

[7] L. Ferreira, L. A. S. C. Cruz, and P. Assuncao, *UHD Video Retargeting based on Visual Attention Models and Temporal Consistency*, IEEE COMSOC MMTC Communications - Frontiers **11**, 79–84January (2016).

[8] T. Cane and J. Ferryman, "Saliency-based detection for maritime object tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2016), pp. 1257–1264.

[9] M. M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. M. Hu, IEEE Transactions on Pattern Analysis and Machine Intelligence **37**, 569–582March (2015).

[10] N. Martins-Ferreira and L. Ferreira, On a procedure for the automatic detection of rectangular regions of interest for arbitrary images, CDRSP-IPLeiria Technical Report, GTLab(Images2roi-1) (88) (2016).

[11] L. F. P. F. N. Martins-Ferreira, P. Assuncao and N. Alves, A method for automatic detection of rectangular regions of interest in arbitrary images, CDRSP-IPLeiria Technical Report, GTLab(Images2roi-1) (91) (2017).