# RANKING REGIONS OF VISUAL SALIENCY IN RGB-D CONTENT

*Dylan Seychell, Carl J. Debono*

Department of Computer and Communications Engineering, Faculty of ICT, University of Malta

## ABSTRACT

Effective immersion takes place when the user can relate to the 3D environment presented and interact with key objects. Efficiently predicting which objects in a scene are in the user's attention, without using additional hardware, such as eye tracking solutions, provides an opportunity for creating more immersive scenes in real time and at lower costs. This is nonetheless algorithmically challenging. In this paper, we are proposing a technique that efficiently and effectively identifies the most salient objects in a scene. We show how it accurately matches user selection within 0.04s and is over 95% faster than other saliency algorithms while also providing a ranking of the most salient segments in a scene.

***Index Terms***— Saliency Ranking, Immersion, Segmentation

## 1. INTRODUCTION

Immersion and interactive technologies are a priority in domains of augmented and virtual reality [1] [2]. One of the main challenges in these media is the lack of adequate interactive devices [2] and this amplifies the need of having intelligent algorithms capable of quickly and efficiently understanding what the user is looking at in the 3D or virtual environment.

Knowing which objects in a scene are of interest to the user provides an opportunity to edit the scene to maximise immersion. In this paper, we propose a technique that can efficiently and accurately predict the most salient objects in a scene and rank them accordingly by saliency.

This solution is designed such that it can be used independently from the algorithm employed to generate the saliency map and can thus be used as a module in other systems. Its efficient design minimizes the overheads and provides a further opportunity for such deployment. The output of this technique can be exploited in various applications such as in transmission over noisy bandwidth-limited channels, where decisions can be taken on which regions to transmit first or which segments are allocated more bits. Moreover, this solution can be applied to image editing applications where the rank can be used to expedite processing by initiating object removal before user selection, such as [3], or prompt the region where a foreign object can be inserted to gain prominence or otherwise.

The first part of this paper provides a background on the different approaches of saliency detection. These different approaches were investigated to ascertain that the proposed technique is independent of the saliency algorithm used to generate the saliency map. A detailed description of the solution architecture together with its respective modules is presented in Section 4. This is accompanied by a description of the online test in Section 5 and the overall comparative evaluation in Section 6. Finally, Section 7 concludes the paper.

## 2. BACKGROUND

There are different frameworks that define immersion [1] and these underline the importance of quality visual feedback for the user to feel immersed in the environment. Teather and Stuerzlinger, [2] provide a list of guidelines for positioning elements in a 3D environment. One of these guidelines states the importance of only having visible objects with the possibility of manipulation, implying the need to know what is most visible in the scene. Perspective and occlusions are challenges in 3D scenes [2] and the prioritization of objects with respect to their depth is important for better immersion. Two Degree of Freedom (DOF) devices provide better precision [2] and a system that can efficiently define salient object in the same dimension enables more immersive applications.

These guidelines show the importance of having intelligent algorithms that can precisely identify which regions or objects in a scene are most salient.

## 3. LITERATURE REVIEW

The main approaches in image saliency detection can be categorized into: the eye-fixation approach, the traditional approach, the information theory approach and the deep learning approach. This section briefly reviews these four slaiency approaches.

### 3.1. Eye-Fixation Approach

The main motivation and inspiration behind the early saliency map algorithm was the visual attention system of primates [4]. This approach is used to rapidly interpret complex scenes by

978-1-5386-7590-8/18/$31.00 ©2018 IEEE

detecting features and decompose the input into a topographical map. Saliency Maps therefore provide a method to distinguish between the foreground and background in an image. This technique starts by filtering linear features in the image considering color, pixel intensity and orientation [4]. Subsequently, these features are normalized and mapped such that an across-scale normalization can take place. This results in a saliency map that is processed in the final stage where a "winner-take-all" neural network ensures that only the most active locations in the image remain while the others are suppressed.

Users tend to fixate on objects that are closer to the center of the image [5]. This effect, also known as center-bias, is further confirmed by experiments where the average of saliency maps were computed [6] and others with mean position of object locations [7]. In such cases, results show that a 2D Guassian distribution is followed across the image, reflecting the probability of fixation.

Fixation also varies between regions with respect to the foreground or the background of an image. Different works show that objects in the foreground are more salient than those in the background [5]. Thus, background and foreground information should be handled with different priorities [8].

### 3.2. Traditional Approach

The traditional approach in saliency is categorized as bottom-up and top-down methods. The bottom-up approach is independent from any target and results from the computation of information around a specific small region and its neighboring features. Moreover, such approaches are also based on elementary attributes of a scene [5] [9] and are normally evaluated against eye-fixation models [7]. Top-down approaches are on the other hand based more on a target interpretation of the image that results from a given training set [8] and are generally slow [5].

### 3.3. Information Theory

The use of information theory in the context of finding where users look at in an image is also adopted. The human visual system evolved to look at regions of maximal information [10]. Shannon's self-information was used to implement such an approach in local regions of images [11]. Furthermore, self-information was also used to process eye-tracking data of 13 users over 1003 images and generate a respective model to predict human fixation in an image through a machine learning approach in [6]. Similar assumptions were also used in the construction of super-pixels and their respective traversal [12]. In such an approach, the image is represented as a graph where the super-pixels are translated to nodes and the similarity defines the edges between nodes. This computational cost of superpixels is inherent in such approaches [12] [13]. These techniques are used as an alternative method to generate a saliency map and the output is a complete saliency map rather than a specified region of interest as presented in this paper.

### 3.4. Deep-Learning Approach

The recent advances of deep learning are also playing a relevant role as an approach of generating saliency maps and is adopted in various work [14] [15] [16] [7]. These approaches are using hand-crafted models to train their models within a convolutional neural network architecture. Training can also entail different saliency mechanisms and therefore approximate to a more general solution [7]. This set of approaches is classified as a top-down approach. The main drawback of using deep-learning is the need of a uniformly sampled and extensive labelled dataset for training together with the resources required for this training [8].

## 4. PROPOSED SOLUTION

This section presents a detailed description of the solution architecture. The adopted approach is modular and accepts a saliency map generated using any of the approaches presented in Section 3 as input. In this paper, we are presenting a technique that does not require generation of superpixels such as [12] [13] and instead uses a more efficient grid approach to provide a starting point for the identification of regions with highest entropy.

### 4.1. Architecture

The architecture of the ranking visual saliency algorithm is organized into two stages. The solution requires the texture and depth frames of a scene as input. Depth maps can be obtained from depth sensors or through disparity measurement of stereo or multiview images. However, if depth maps are not available, the solution can still return a result that is still indicative of the relative saliency of the regions, albeit with a different ranking given that in such instances the computation assumes a flat depth.

The first stage, shown in Figure 1, is initialized by providing the texture and depth images together with the number of segments required in the grid to specify the grid template $T$. The implementation of Saliency Maps [4] developed and made available by [17] was used in the generation of results of the proposed solution. Once generated, the saliency map together with the depth map are processed according to $T$.

The second stage, presented in Figure 2, processes the segments to compute a respective score for every element $t$ in $T$. The scores reflect the proximity to the center of the image as explained in Section 4.2, the entropy of each segment $t$ of the saliency map as explained in Section 4.3 and the depth score of each segment of the depth map as presented in Section 4.4. These scores are then combined to return a single summation score for each segment of the image and are used

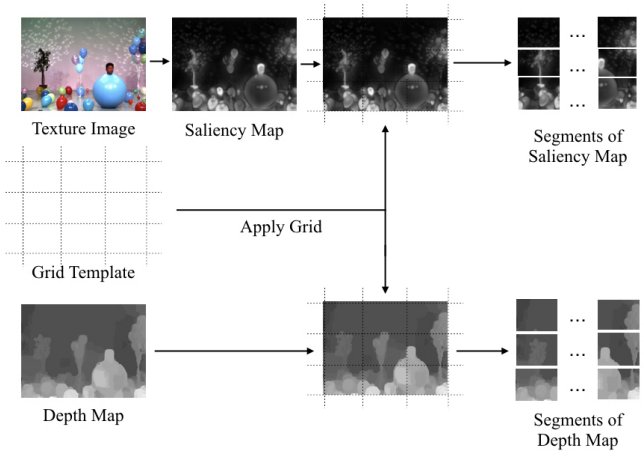


**Fig. 1**. Stage 1 of the proposed solution consists of generating a saliency map with the saliency algorithm of choice and initializing a grid-template that specifies how many grids will be processed. Once these are completed, the grid template is used to split both the saliency map and depth map in segments that are then processed in Stage 2.

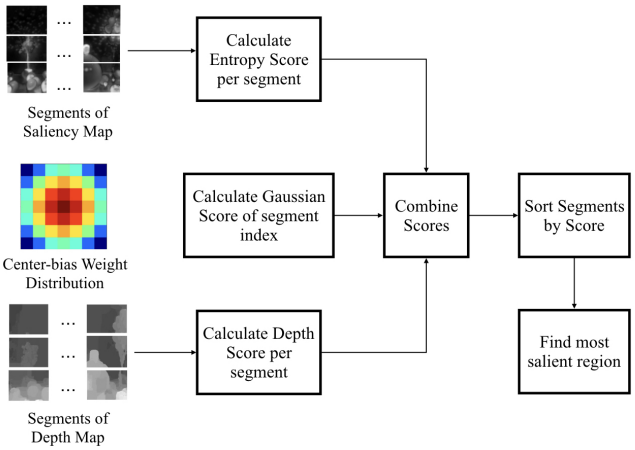


**Fig. 2**. Stage 2 starts with three parallel modules. The segments of the saliency map are processed to calculate their entropy, the segments of depth are processed for the calculation of a depth score and the 2D Gaussian distribution is used to calculate the score of a segment's position. These scores are then combined and sorted to identify the most salient region.

to sort the segments by the same score that represents the most salient segments.

### 4.2. Center-Bias

The proposed technique makes use of a 2D Gaussian distribution to factor the importance of center-bias that is reported in Section 3.1. Equation (1) is used to generate a distribution of weights where every segment in the grid would be assigned a value depending on its position where $x$ and $y$ are the dimensions of the 2d-array and $\sigma$ is the effective radius of the distribution. In this implementation, the full-width-half-maximum is used as a value for $\sigma$. The segment at the centre carries the maximum weight while the outlying segments are assigned a minimal value. A visual representation of this distribution can be seen in the center-left of Figure 2.

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

### 4.3. Information Theory

Information theory principles were applied to this technique in a novel approach. The techniques and approaches presented in Section 3.3 demonstrate how information theory can be used as an alternative to generate saliency maps. This technique was inspired by the use of information theory but is nonetheless applying it in a very different manner.

$$H(X) = -\sum_{i=1}^{|t|} P(x_i) log_2(P(x_i)) \tag{2}$$

In the proposed solution, Shannon's Entropy presented in Equation (2) is applied on the resultant saliency map where $P(x_i)$ is the probability of finding the pixel value of pixel $i$ in the designated segment. More precisely, it is used to calculate the entropy in every grid-segment. In this case, the entropy value $H$ is representing the rate-of-change or chaos in a given grid-segment. This means, that the grid-segments of the saliency map with higher values of $H$ are the same grid-segments that have significant chaos with respect to pixel values in the saliency map. It also follows that the same grid-segment on the original texture image contain features that draw attention.

### 4.4. Depth Score

As discussed in Section 3.1, objects that are closer to the user are more salient than those in the background. Since saliency maps are usually generated without any knowledge of depth, the concept of calculating a depth score from the depth map of the same scene is proposed.

Algorithm 1 outlines a technique that generates a score for depth for every segment in $T$. This therefore returns a score for every segment and the segments with objects closer to the user are rewarded. Since depth maps from different datasets vary with respect to different shades used to represent depth, the histogram of the entire depth map is processed prior to the processing of each grid segment $t$ for calibration. The histogram $H_D$ of the entire depth map is processed with the first and last non-zero bin indexes being recorded. The midpoint is then stored in $mid$ as shown in Algorithm 1.

The depth map is then split in the same number of seg-

**Algorithm 1** CalculateDepthScores

**Input:** $DepthMap, T.size$
**Output:** $DepthScores$
  $H_D \leftarrow DepthMap.histogram$
  $mid \leftarrow (H_D.maxBin + H_D.minBin)/2$
  $Segments \leftarrow splitIntoSegments(DepthMap, T.size)$
  **for all** $segment$: $s \in Segments$ **do**
    $H_S \leftarrow s.histogram$
    $tempScore \leftarrow 0$
    **for all** $bin$: $b \in H_S$ **do**
      **if** $b.index > mid$ **then**
        $tempScore \leftarrow tempScore + b$
      **end if**
    **end for**
    $DepthScores.append(tempScore)$
  **end for**

ments $T.size$ used to split the saliency map and to generate the 2D Gaussian Distribution. The histogram $H_S$ for each segment is computed and its bins' values traversed. Since the aim of this algorithm is to provide a list of scores for each segment and reward the segments with objects closer to the user, the bins with index larger than $mid$ are considered. The summation of the bin values is then considered as the depth score for the given segment. This depth score is then appended to a list of scores whose index relates to the same index of the segment $t$ in $T$ as explained in Section 4.1. This list of depth scores is returned as an output by Algorithm 1.

### 4.5. Output

The output represents a collection of segments sorted by the combined score. The data-structure containing the output stores the index of the segment, the segment itself in RGB together with the top-left and bottom-right coordinates. This data structure facilitates the integration of this algorithm in other systems.

For easier analysis and evaluation, the grid-segments in the visual output were colored in four batches. The first 25% of segments, which includes the most salient regions, is colored in red, the second in orange, the third in yellow and the forth in green.

## 5. ONLINE TEST

A widespread approach for user evaluation was essential in this study. Thus, a website was set up to facilitate the dissemination and availability. This approach allowed for a very detailed data collection methodology. A web infrastructure that presents the images to the users through HTML, collects information through JavaScript and stores it in a hosted database was developed. In the background, the usage information was collected. This information included information about the session together with the operating system and browser being used by the user. A set of images were presented to the user and or each image, the users were asked the following question:

**Task 1:** *Click/Tap on the point that attracts your attention when you first see the image. The point can be anywhere and includes persons or other objects.*

The coordinates of the user click or tap on an image were recorded. These were then used to generate the heatmap of the regions of interest in the images as shown in Figure 5.For every image, the time between the presentation of the image to the user and the click or tap was recorded, allowing for better understanding of the time taken by users to choose an object from the scene. This was also useful in the cleaning of data.

The URL was promoted on social media and a total of 987 respondents participated in the study. The data was then cleaned by the *time to click* variable and the entries with values less than 0.1s and more than 50s were discarded. This left 854 records for evaluation. The mean time for users to click on the object was 4.3s with the fastest click taking place at 0.48s and the slowest at 41.3s. It is also interesting to note that the mean x-coordinate of the clicks took place at 53% of the image width and the mean y-coordinate of the clicks took place at 51.1% of the height. This shows that the user clicks reflect the effect and relevance of center-bias.

## 6. EVALUATION

The proposed algorithm was evaluated using a selection of multiview datasets that can be used to perceive natural images and scenes in 3D. These are Microsoft's *Ballet* and *Breakdance* sequence [18], Nagoya University's *Balloons* sequence [19], Fu *et al*'s *Bear* sequence [20] and Tsukuba's University stereo dataset [21].

These datasets were carefully selected for the evaluation of this technique since while they are commonly used to evaluate other techniques in the area they also provide texture and depth images for each scene. Furthermore, these scenes provide a selection of different objects and compositions. There are scenes with persons who are close to the camera and others that are far. There are also other scenes that only feature objects or a mix of objects and persons. Across the selection of these scenes one can find well illuminated scenes and darker scenes. On the other hand, saliency algorithms are normally evaluated using datasets such as the MSRA [22] where there is a single object in every image. However, this paper is focusing on ranking the saliency of different objects in a single scene and therefore these datasets were not appropriate for a realistic evaluation.

### 6.1. The Optimal Grid Size

The comparative results of this algorithm against the resultant user clicks from the online experiments were computing using a 9×9 grid or segment dimension. This section lists a series of experiments that were carried out to investigate how the size of the grid affects the result. These experiments show that there is no effective difference, other than resolution of output, taking place when the grid segment dimension (SegDim) exceeds 9×9.

A set of four scenes was chosen as a sample to evaluate the ideal SegDim. These were the *Ballet* sequence, the *Balloons* sequence, the *Bear* sequence and the *Tsukuba* dataset. These scenes feature a selection of objects that were both detected by the proposed technique and also clicked by users on the online test.

In an effort to decide which is the best SegDim to be used for adequate comparison, different segment dimensions were generated for each of these scenes. These ranged from a SegDim of 2, hence returning a 2×2 grid to that of 20. A visual sample is presented in Figure 3.

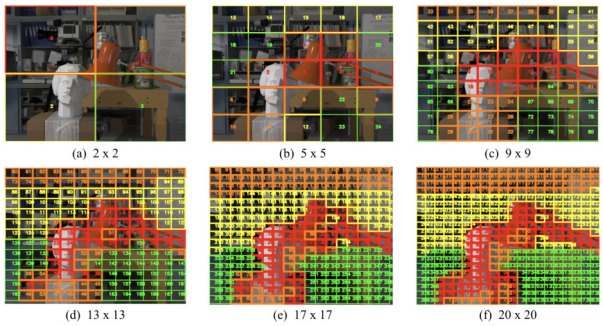


**Fig. 3**. A selection of the results generated on the *Tsukuba* dataset with different segment dimensions as illustrated under each image.

For every scene, a set of objects was chosen as discussed above. It followed that the segments used to cover the area of every designated object was counted for every segment dimension. This was repeated for different objects in the same scene and then for every other object in the other scenes. A sample of the data collected is presented in Table 1.

This procedure was carried out for each of the four scenes and the results were plotted on the chart presented in Figure 4. This chart shows that after a particular value of SegDim, the curve tends to settle and the variation between each value from that value onwards tends to remain stable. The SegDim value at which the curve settles was recorded for every curve representing a different object in different scenes. This was calculated by taking the average '% Hit' for each object and identifying at which segment dimension this value was first encountered. From the results presented in Table 1, the average value of the resultant segment dimensions is 8.45 and therefore it was decided that the segment dimension of 9 be chosen for the comparison of results in subsequent sections of this paper.

**Table 1**. A sample of the data collected in the counting of segments required to represent a specific object for every segment dimension. This includes SegDim, Hits refers to the number of segments covering the objects, Max Hits refers to the total number of segments in the given segment dimension and % Hits refers to the ratio of Hits against Max Hits.

| Dataset | Object | SegDim | Hits | Max Hits | % Hits |
|---|---|---|---|---|---|
| Tsukuba | Head | 2 | 2 | 4 | 0.500 |
| Tsukuba | Head | 9 | 11 | 81 | 0.136 |
| Tsukuba | Head | 20 | 36 | 400 | 0.090 |
| Tsukuba | Lamp | 2 | 4 | 4 | 1.000 |
| Tsukuba | Lamp | 9 | 15 | 81 | 0.185 |
| Tsukuba | Lamp | 20 | 54 | 400 | 0.130 |

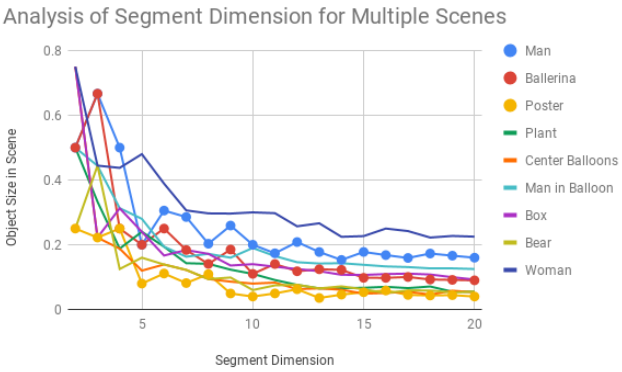


**Fig. 4**. A chart depicting the results of each object in every tested scene. Every curve shows that the number of segments representing an object tends to settle after a certain value.

### 6.2. Comparison of Algorithm with User Clicks

The results of the proposed technique were compared to the results collected from the online test. This part of the evaluation process compares the prioritization of the salient regions of the images detected by the algorithm with where the users clicked during the test. The choice of this test is based on the motivation that such a technique is more relevant and valuable if it successfully mimics human behavior.

Following the data collection process explained above, a data processing algorithm was devised to return the online test results in a comparable format to that generated by the proposed technique. This evaluation algorithm processed the click coordinates of every user per image and returned an image with a grid of the same segment dimension of $9 \times 9$.

Every click coordinate $(x, y)$ had to be mapped with a grid

segment with index $(i, j)$. Every image has its own width $w$ and height $h$ pixels. The segment dimension is denoted as $D$. This follows that $x$ falls in the range $x = [0, w)$ and $y$ in the range $y = [0, h)$. The segment $S(i, j)$ follows Equation (3) that gives the index of the respective cell as presented in Equation (4).

$$S(i,j) = \left\{ (x,y) \;\middle|\; \begin{array}{c} i\frac{w}{D} \leq x < (i+1)\frac{w}{D}, \\ j\frac{h}{D} \leq y < (j+1)\frac{h}{D} \end{array} \right\} \quad (3)$$

$$i = \left\lfloor \frac{xD}{w} \right\rfloor,\ j = \left\lfloor \frac{yD}{h} \right\rfloor \quad (4)$$

Following the results presented in Section 6.1, the segment dimension was set to a 9×9 grid.

The following procedure was followed for each scene in an effort to provide a comparison between the ranking of the algorithm with the user clicks. The objects clicked by users were identified and listed. The algorithm was evaluated to compare with the choice of objects by users. For each object, the number of red, orange, yellow and green labelled segments were separately counted in the result returned by the algorithm and the user clicks.

The $\chi^2$-test was used to determine whether there is a relationship between the resultant labelled segments of the algorithm and the user clicks. The following hypothesis was therefore selected for evaluation:

$HypothesisH_0$ There is no significant difference between the algorithm ranking result and the user clicks

It follows that when the null-hypothesis $H_0$ is not rejected, the algorithm successfully predicted the segment where the user clicked. The results of the $\chi^2$-test with the algorithm using depth information are presented in Table 2. The cases where $H_0$ was rejected were two items in the Balloons dataset and the man in the Ballet dataset. While in the Balloons sequence the algorithm discarded the plant and the center balloons, the case of the man in the Ballet sequence deserves further investigation. In this scene, the man covers 25% of the image, which is a significant amount when compared to the number of pixels in the image where a user can potentially click. The algorithm detected the man from top to bottom while on the other hand, users only clicked on the man's head and vest. This therefore resulted that when all the algorithm result of segments representing the man were counted and tested against the user clicks, $H_0$ was rejected. However, given this situation, one can safely consider that the algorithm predicted the choice of the man and can be also considered as another successful prediction of the algorithm. All this results in the algorithm correctly predicting 91.3% of the target objects. The same precision was achieved by the algorithm when it did not use the depth information. However, the objects not detected when depth was enabled, were captured when it was not. This shows that the technique can offer a complete coverage of user selection.

**Table 2**. Table showing the $\chi^2$-square statistic and the respective $\rho$-value for the test of hypothesis $H_0$ for each object in the respective scenes.

| Dataset | Object | $\chi^2$ | $\rho$-value | $H_0$ rejected? |
|---|---|---|---|---|
| Ballet | Man | 10.095 | 0.0178 | Yes |
| Ballet | Ballerina | 2.9706 | 0.3962 | No |
| Ballet | Poster | 1.5 | 0.6823 | No |
| Balloons | Man | 3.8294 | 0.2805 | No |
| Balloons | Center Ballons | 6.5714 | 0.0468 | Yes |
| Balloons | Plant | 8.0563 | 0.0448 | Yes |
| Tsukuba | Lamp | 3.3333 | 0.3430 | No |
| Tsukuba | Head | 1.1778 | 0.7583 | No |
| FuBear1 | Bear | 3.2667 | 0.3523 | No |
| FuBear1 | Box | 1.1818 | 0.7574 | No |
| FuBear1 | Woman | 1.3518 | 0.7169 | No |
| FuBear2 | Bear | 0.4 | 0.9402 | No |
| FuBear2 | Box | 6.2667 | 0.0993 | No |
| FuBear2 | Woman | 0.6421 | 0.8867 | No |
| Sofa | Table | 3.4961 | 0.3213 | No |
| Breakdance | Dancer | 2.5 | 0.4753 | No |
| Breakdance | Radio | 1.0667 | 0.7851 | No |
| Breakdance | RightMan | 0.7222 | 0.8680 | No |
| Breakdance | LeftMan | 1.9762 | 0.5774 | No |
| FuCup | Kettle | 4.6095 | 0.2027 | No |
| FuCup | WallSocket | 1.6667 | 0.6444 | No |
| FuCup | Right Objects | 0.66667 | 0.881 | No |
| FuCup | Center Objects | 1.1667 | 0.761 | No |

### 6.3. Performance and Computational Complexity

An instance of the above architecture was implemented in Python 3.6 and OpenCV 3.0 as a proof of concept. Performance testing was carried out on a machine with a 2.6 GHz Intel Core i5 processor and 8 GB 1600 MHz DDR3 memory running a MacOSX High Sierra n v0.13.5 operating system. An instance was also executed without the depth information. The closest similar techniques found in literature use other saliency techniques and generate segmentation of the most salient objects using a binarized map of each technique. The visual comparison can be seen in Figure 5 and performance results in time (s) can be found in Table 3.

The computational complexity of the proposed solution depends on the size of the texture and depth image. The traversal of each pixel in each image is required and this results in $2\mathcal{O}(N) \Rightarrow \mathcal{O}(N)$ where $N$ is the number of pixels in the image. This also means that the segment dimension does

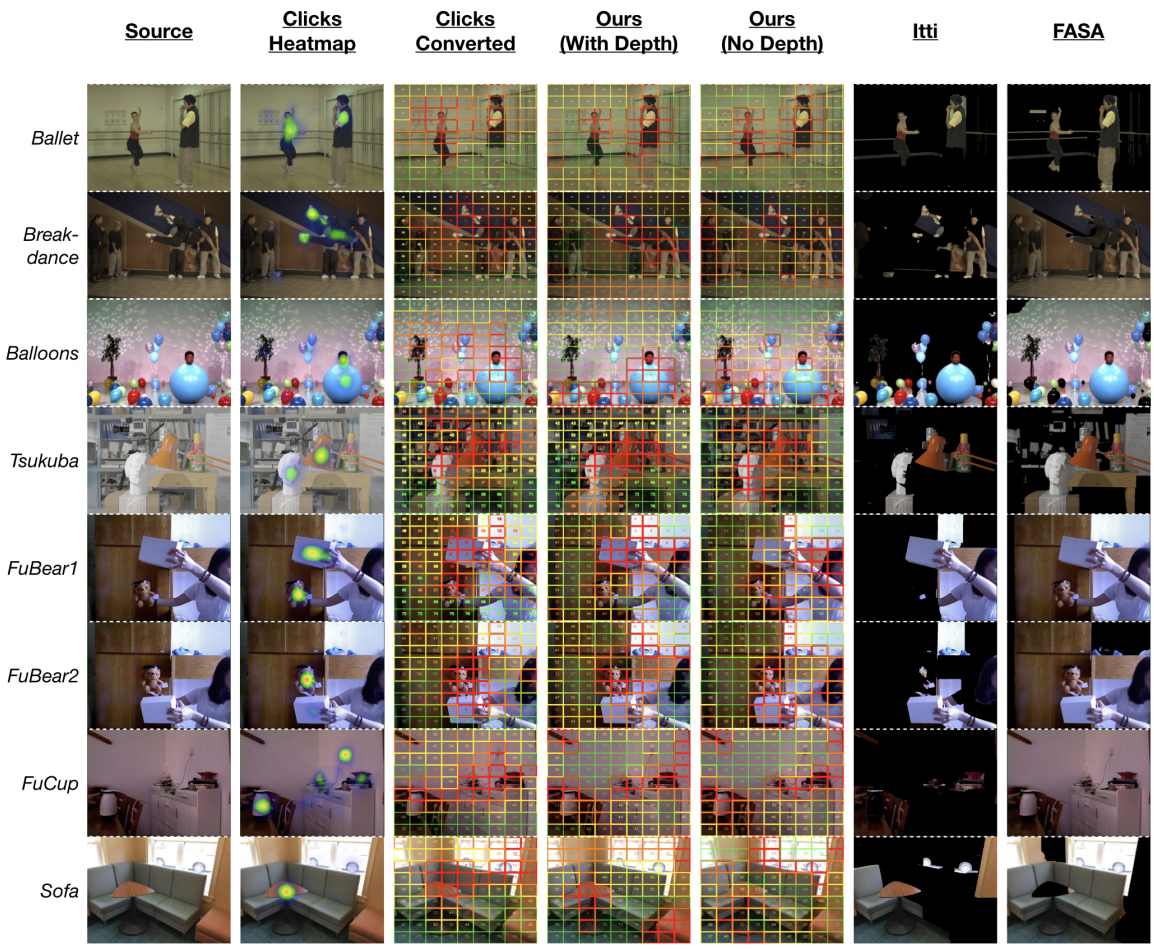


**Fig. 5**. A visual comparison of how the proposed solution compares to the user clicks and other techniques.

**Table 3**. Table showing the time taken for the algorithms to execute the segmentation. The first column refers to the image being processed, followed by the results in seconds of our algorithm with depth enabled and disabled respectively. The last two columns present the time taken in seconds for Itti's [4] and FASA's [9] algorithm to carryout similar segmentation.

| Dataset | t(Depth) | t(NoDepth) | t(Itti) | t(FASA) |
|---|---|---|---|---|
| Ballet | 0.1498 | 0.0530 | 2.4306 | 4.7573 |
| Balloons | 0.1264 | 0.0590 | 2.0461 | 4.1101 |
| Breakdance | 0.1537 | 0.0568 | 17.9918 | 4.8980 |
| FuBear1 | 0.1086 | 0.0383 | 1.9396 | 1.6093 |
| FuBear2 | 0.0911 | 0.0387 | 1.7726 | 1.6805 |
| FuCup | 0.0893 | 0.0327 | 2.1946 | 1.5265 |
| Table | 0.0940 | 0.0420 | 2.7215 | 2.8111 |
| Sofa | 0.1086 | 0.0390 | 3.1986 | 1.5280 |
| Tsukuba | 0.1141 | 0.0417 | 1.0810 | 1.5082 |

not affect the complexity of the algorithm. This therefore provides more flexibility in the implementation of the proposed technique in other applications since the segment dimension can scale depending on the designated requirements of the end application.

## 7. CONCLUSION

The technique presented in this paper shows how the most salient regions of an image can be ranked into a number of grid segments without the need of prior training or knowledge. This provides various opportunities in the fields of data compression, image editing and segmentation since, due to its efficiency, it can easily be integrated within existing techniques. Knowing how objects or elements in the users' view are ranked in terms of saliency, provides an effective way of delivering a better immersive experience by providing an efficient way of identifying the most important regions in the scene and hence ensure that better quality is present in these regions.

## 8. REFERENCES


[1] M. G. Barrio J. L. Rubio-Tamayo and F. Garcia Garcia, "Immersive environments and virtual reality: Systematic review and advances in communication, interaction and simulation", *Multimodal Technologies and Interaction*, vol. 1, no. 4, 2017.

[2] R.J. Teather and W. Stuerzlinger, "Guidelines for 3d positioning techniques", in *Proceedings of the 2007 Conference on Future Play*, New York, NY, USA, 2007, Future Play '07, pp. 61–68, ACM.

[3] D. Seychell and C.J. James Debono, "Monoscopic inpainting approach using depth information", in *Proceedings of the 18th IEEE Mediterranean Electrotechnical Conference*, 2016.

[4] 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, Nov 1998.

[5] P Sharma, "Evaluating visual saliency algorithms: Past, present and future", *Journal of Imaging Science and Technology*, vol. vol. 59, no. 5, pp. 50501 - 1 - 50501-17.

[6] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look", in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 2106–2113.

[7] A. Borji, "Boosting bottom-up and top-down visual features for saliency estimation", in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 438–445.

[8] C. Zhu, K. Huang, and G. Li, "Automatic salient object detection for panoramic images using region growing and fixation prediction model", *Journal of Computer Research Repository (CoRR)*, 2017.

[9] G. Yildirim and S. Süsstrunk, "Fasa: Fast, accurate, and size-aware salient object detection", in *Computer Vision – ACCV 2014*, Cham, 2015, pp. 514–528, Springer International Publishing.

[10] T. Lee and S. Yu, "An information-theoretic framework for understanding saccadic behaviors", in *Advances in Neural Information Processing Systems (NIPS)*. 2000, MIT Press.

[11] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization", in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, Cambridge, MA, USA, 2005, NIPS'05, pp. 155–162, MIT Press.

[12] J. Lu, X. Wen, H. Shao, Z. Lu, and Y. Chen, "An effective visual saliency detection method based on maximum entropy random walk", in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–6.

[13] J. G. Yu, J. Zhao, J. Tian, and Y. Tan, "Maximal entropy random walk for region-based visual saliency", *IEEE Transactions on Cybernetics*, vol. 44, no. 9, pp. 1661–1672, Sept. 2014.

[14] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection", in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 202–211.

[15] Y. Hu, Z. Chen, Z. Chi, and H. Fu, "Learning to detect saliency with deep structure", in *Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics*, Oct 2015, pp. 1770–1775.

[16] G. Lee, Y. W. Tai, and J. Kim, "Eld-net: An efficient deep learning architecture for accurate saliency detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[17] Akisat Kimura, "Saliency map implementation", [Online]. Available: https://github.com/akisato-/pySaliencyMap/ , Accessed: 2017-11-1.

[18] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation", in *ACM SIGGRAPH 2004 Papers*, New York, NY, USA, 2004, SIGGRAPH '04, pp. 600–608, ACM.

[19] "Nagoya university sequences", [Online]. Available: http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/, Accessed: 2015-06-1.

[20] H. Fu, D. Xu, S. Lin, and J. Liu, "Object-based rgbd image co-segmentation with mutex constraint", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 4428–4436.

[21] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui, "Towards a simulation driven stereo vision system", in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov 2012, pp. 1038–1042.

[22] 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, 2015.