https://doi.org/10.2352/ISSN.2470-1173.2017.17.COIMG-433

Augmenting Salient Foreground Detection using Fiedler Vector for Multi-Object Segmentation

Michal Kucer, Nathan D. Cahill*, Alexander C. Loui**, David W. Messinger*; *Rochester Institute of Technology, **Kodak Alaris Inc.; Rochester, NY*

Abstract

In this paper we present multiple methods to augment a graph-based foreground detection scheme which uses the smallest nonzero eigenvector to compute the saliency scores in the image. First, we present an augmented background prior to improve the foreground segmentation results. Furthermore, we present and demonstrate three complementary methods, which allow for detection of the foregrounds containing multiple subjects. The first method performs an iterative segmentation of the image to "pull out" the various salient objects in the image. In the second method, we used a higher dimensional embedding of the image graph to estimate the saliency score and extract multiple salient objects. The last method, using a proposed heuristic based on eigenvalue difference, constructs a saliency map of an image using a predetermined number of smallest eigenvectors. Experimental results show that the proposed methods do succeed in extracting multiple foreground subject more successfully as compared to the original method.

Introduction

As we move through our daily lives, we are bombarded with an immense amount of visual data. Processing all of this information is physically impossible. However, our brain possesses a mechanism known as visual attention for selecting a subset of the relevant data that we want to focus on. Modeling of visual attention is an extremely important task with many important applications in robotics and computer vision including image compression, object detection, and computer graphics [1].

The notion of relevance in the visual attention models is mainly determined by two processes: bottom-up and top-down processes. Bottom-up attention modeling, also called visual saliency, uses various low-level features including image color, intensity and orientation to determine the contrast of objects with respect to their surroundings [1]. On the contrary, the top-down attention selects the relevant image areas based on task-driven factors such as knowledge, expectation or current goals.

Related work

We focus the review on the relevant literature regarding the bottom up visual saliency, especially as it relates to the various saliency estimation approaches that are used to benchmark against the algorithm of [2]. Bottom-up saliency models can in general be described as belonging to one of the following categories: biologically inspired, purely computational and a combination [3]. For a more exhaustive treatment, please see reference [1].

Biologically inspired models, e.g. the model proposed by Itti et al. [4], are often based upon the architecture presented by Koch et al. [5], which used biologically inspired features processed by center-surround operations to determine the saliency score and correctly predict eye fixations.

Computation-oriented models, which use low level image features such as color, emphasize the practical aspect of models such as speed and aim to create saliency maps which segment whole objects and preserve edges [2]. Recently, several models [2, 6, 7, 8, 9, 10] use a variation of super-pixel segmentation methods akin to the SLIC (Simple Linear Iterative Clustering) method [11], to accomplish those goals. Methods such as SLIC oversegment the image into perceptually coherent patches (whose number is much smaller than the number of image pixels) which are able to both preserve the local color information and edges, while abstracting away unnecessary details (i.e. non-significant pixel-to-pixel intensity). Cheng et al. [9] use spatially weighted region contrast to estimate the saliency based on the color histogram differences. Perazzi et al. [10] show the possibility of modeling the saliency estimation in a unified way using high dimensional Gaussian filters, where they combine measures of image patch uniqueness and spatial distribution to estimate the saliency score. Wei et al. [7] build an image graph out of the super-pixel segmentation and estimate the saliency of a patch to be proportional to the shortest path distance from the virtual background node to the said patch. Yang et al. [12] construct an image graph and enforce a background assumption, which assumes most of the borders belong to the background. The authors use a ranking function, which given a query, determines how similar are the remaining nodes to the query nodes. The authors construct a scheme in which they compute the score by determining the saliency of a patch being proportional to the similarity / dissimilarity from the foreground / background queries. Chang et al. [8] use initial saliency maps, measures of objectness, and a measure of how likely an area is to contain an object, to optimize a novel energy function and obtain an improved saliency map.

Algorithm

Original algorithm

In order to efficiently represent the image, Perazzi et. al [2] use the modified version of the SLIC Superpixel Segmentation algorithm [11] proposed in [13], where the image is segmented into superpixels using k-means clustering in the Color-XY space ([13] uses CIELab color space instead of the traditional RGB space). After Superpixel segmentation, the image is represented as a Graph $\mathscr{G} = \{\mathscr{V}, \mathscr{E}\}$ also known as the image Region Adjacency Graph (RAG), where each vertex $v \in \mathscr{V}$ is representing a superpixel from SLIC and is assigned a value of the mean Lab color of the superpixel. To model the local relationships in the image, the edge set $\mathscr{E}$ consists of the edges connecting vertices i and j, if their corresponding superpixels share a border in the seg-

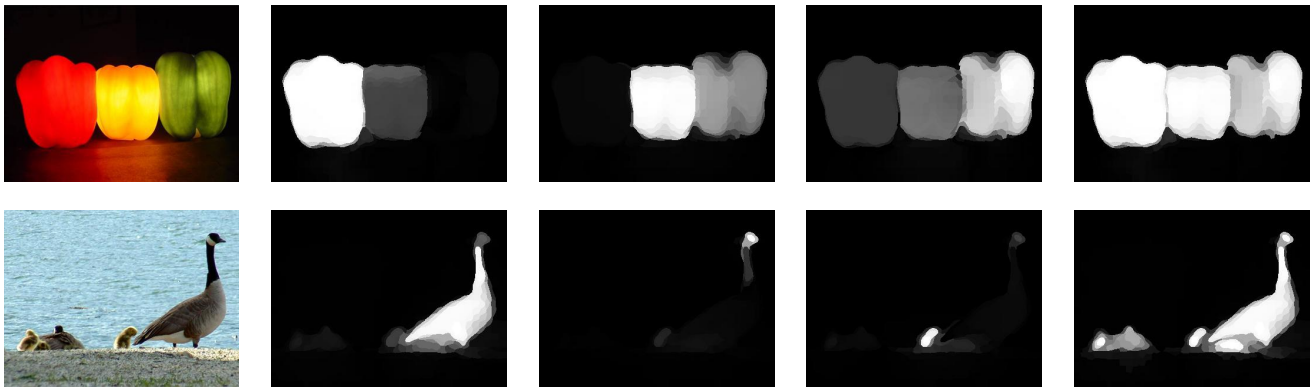

Figure 1. *Images that show the presence of separate objects / object parts in the higher eigenvector dimensions. From left: Original image, saliency map constructed from first non-zero eigenvector, saliency map constructed from second non-zero eigenvector, saliency map constructed from third non-zero eigenvector, and the final saliency map, whose construction will be described in later section.*

mented image. Each edge is assigned a weight that is proportional to the Lab color difference between neighboring superpixels,

$$w_{i,j} = \frac{1}{\|c_i - c_j\|^2 + \varepsilon} \tag{1}$$

where c_i is a mean Lab color of the i^{th} superpixel and ε is a small constant (e.g., $\varepsilon = 10^{-4}$) to ensure the numerical stability of the algorithm, in case the color difference is too small. In order to represent the assumption that most of the border pixels belong to the background, Perazzi et al. [2] augment the graph $\mathcal{G}$ with a background node b, which is assigned the mean Lab color of the boundary. A set of edges and their weights that connect the background node and the superpixels on the border of the image are computed by equation 1.

In order to assign saliency score to each of the superpixels of the image, Perazzi et al. compute the Eigen decomposition of the graph Laplacian matrix L of the Image RAG. Then the Fiedler vector, the second smallest eigenvector, is used compute the saliency scores. Given the Fiedler vector f, the saliency score S is computed as

$$S = -sign(f_b) \cdot f \tag{2}$$

and S then scaled to the range $[0,1]$, where f_b represents the entry of the Fiedler vector corresponding to the background node.

Since one of our proposed approaches considers a high dimensional node embedding, we also propose to compute the saliency scores as

$$S(i) = \|\vec{f}_i - \vec{f}_b\| \tag{3}$$

where S(i) is the saliency score for i^{th} superpixel, and $\vec{f}_i$ and $\vec{f}_b$ are the embeddings of the i^{th} and background superpixels.

Augmenting the background prior

There are images in which the background is often very cluttered, and thus computing the edge weights by considering the average background color will fail to capture the background prior effectively by computing very small edge weights, since the average background color will be sufficiently different from each of the border superpixels and thus resulting in an unsatisfying saliency map (see the top right image of Figure 2). To correct for such a pitfall, instead of assigning to the image background node the average border background color (average color of the border super-pixels), a set of colors representing the background is assigned to the background node. We first perform a K-Means clustering of the border colors and then use the cluster centers, $\{c_1^b, \ldots, c_k^b\}$, to represent the background prior in the node. To compute the edge weight between the background node and the border regions, we simply take the maximum of the weights computed between region i and each of the k cluster center colors

$$w_{i,b} = max_{j \in \{1,\ldots,k\}} \frac{1}{\|c_i - c_j^b\|^2 + \varepsilon}. \tag{4}$$

Augmenting the background prior with multiple "colors", we are able to better enforce the background prior as we can see in Figure 2.

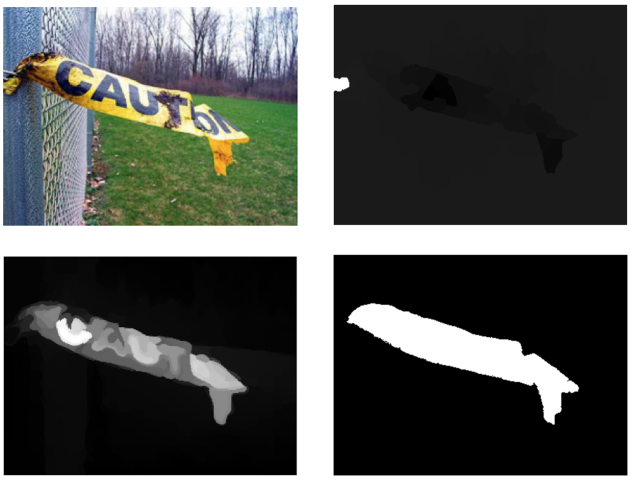

Figure 2. *Comparison of the Saliency maps after augmenting the background prior: original image(top left), Perazzi et al. saliency map (top right), our saliency map (bottom left) and ground truth (bottom right).*

Detecting multiple objects

To extend the foreground segmentation algorithm to allow for detecting multiple salient subjects in the image, we propose the following schemes: an iterative segmentation scheme and two alternative multi-object foreground segmentation methods which use multiple eigenvectors of the Image RAG as an embedding for the nodes and analysis of the presence of additional objects. This embedding is then used to calculate an alternative saliency score. Both of the schemes will use a metric to determine the ideal foreground segmentation. Next we will describe the Silhouette score and the metric we propose for picking the best saliency map.

Silhouette score

In order to judge the quality of the foreground segmentation, we use k-Means clustering to cluster the saliency scores of each super-pixel into two clusters (Foreground / Background) and then compute a metric knows as the Silhouette score, first introduced by Rousseeuw [14]. The Silhouette score is one possible metric that is used in the interpretation and validation of cluster analysis.

To compute the Silhouette score, we need the resulting clustering and the matrix of distances (or dissimilarities as used by [14]) between the different points (e.g. superpixels and the Saliency score assigned to them in our algorithm). For each point i we compute:

- $a(i)$: average distance to the points in the same cluster as i (label that cluster A)
- $D(i,C)$: average distance to the points in cluster C
- $b(i) = \min_{C \neq A} D(i,C)$: by choosing minimum of the $D(i,C)$, we compute the distance to next best cluster assignment for i.

The final score for point i is computed as

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{5}$$

which is then combined into a final score f_{sil} for our image by taking the average of $s(i)$ for all of the superpixels.

Stopping criterion / Metric

Both of the multi-object schemes detailed in the next section rely on some sort of stopping criterion / metric, which would determine either the ideal number of iterations or eigenvectors to consider when computing the saliency map for images with multiple objects. In order to determine the ideal iteration / number of eigenvectors, we propose a metric which combines the Silhouette score, f_{sil}, and mean image saliency of the image

$$score_{image} = f_{sil} \cdot \frac{\sum_{x=1}^{m} \sum_{y=1}^{n} S(x,y)}{A(I)} \tag{6}$$

where $S(x,y)$ is the image saliency score at the location (x, y) and $A(I)$ represents the area of the image.

Then in order to pick the final saliency map, we choose the map with the highest score defined in equation 6.

Presence of objects in eigenvectors

One of the things that we have observed is the presence of multiple salient objects embedded in higher dimensions of the RAG Laplacian matrix eigendecomposition. This can be seen in Figure 1, where we show an example of an image and the saliency maps of its eigenvectors (we compute the saliency of an eigevector by computing the scaled distance of each superpixel to the background node). However the same cannot be said of many of the images that only contain a single salient object, as we can see in Figure 3. The Fiedler vector will pick out the most salient object in the image and the subsequent eigenvector (at times several) will contain redundant information regarding the object. Such observations were originally part of the exploration in creating an appropriate stopping metric.

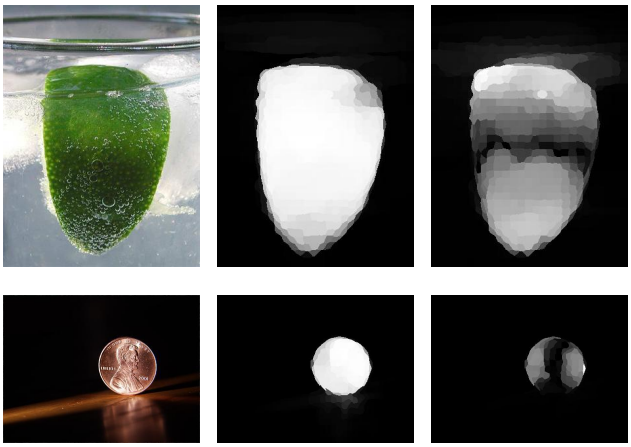

Figure 3. *Plot of the the saliency maps for the first two eigenvectors of the images with a single salient object. From left: orinal image, first non-zero eigenvector, second non-zero eigenvector.*

Stopping criterion based on the eigenvalue difference

A different stopping criterion that we consider is based on the percentage eigenvalue difference between subsequent dimensions. First we compute the full eigendecomposition of the augmented RAG. Then we take a subset of the first k non-zero eigenvalues, and compute the percentage difference between the subsequent dimensions:

$$\Delta_i = \frac{\lambda_{i+1} - \lambda_i}{\lambda_{i+1}} \tag{7}$$

Then in order to get the ideal dimension n, we choose the dimension which produces the largest difference:

$$n = argmax_{1 \geq i < k} \{\Delta_i\}. \tag{8}$$

Multi-object segmentation schemes

The main idea behind the first method, iterative foreground segmentation, is simple: each of the foreground objects are segmented one by one by looking at the most salient object in the image graph at each step of the iteration.

The Iterative Foreground segmentation can be described as:

- Perform an initial foreground segmentation as described in [2] with the improved background prior model, and compute the $score_{image}$ for this map.

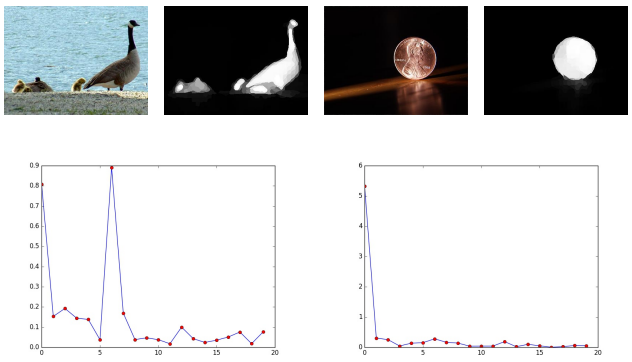

Figure 4. *Plots showing the eigenvalue percentage difference plots for a sample images with single / multiple salient objects.*

- Now, iteratively perform the following steps:
  - Find the set, $\mathscr{S}$, of nodes / super-pixels for which the saliency S_i for super-pixel i is greater than a threshold S_{th}.
  - Modify the Image RAG by cutting out the nodes that belong to the set $\mathscr{S}$ (store the saliency scores of these nodes for later processing).
  - Find new Saliency scores for the region which remained in RAG by computing the Fiedler Vector of the new graph and computing and modifying it the same way described in [2].
  - Combine the Saliency scores of the smaller region with the scores for the nodes from the set $\mathscr{S}$, to obtain the new saliency image and compute its $score_{image}$.
  - Repeat for predetermined number of iterations.
- Choose the map with highest $score_{image}$.

Based on the previous observations of the presence of additional salient objects in different eigenvectors, we prepose two alternative ways of constructing an image Saliency map based on considering multiple eigenvectors.

The first method for foreground segmentation proceeds as follows:

- Construct the RAG of the image as described in [2] and augmented with the improved background node.
- Construct the Laplacian matrix of the Image RAG.
- Consider the k smallest eigenvectors corresponding to non-zero eigenvalues and use them as a k-dimensional embedding of the graph nodes.
- Calculate the new saliency scores by:
  - Calculating the distance between the k-dimensional embedding of the background node and node i.
  - Renormalize all of the distances to lie in the range between $[0,1]$, which will give us the relevant Saliency scores S.
- Compute a metric for maps created by considering projections with varying number of eigenvectors (we consider up to four eigenvectors for the embedding of our graph) and choose the map with highest score achieved by the metric.

In order to observe the map chosen by the score defined above, please refer to the Figure 5 and Figure 6, which show examples of the original images and the corresponding sequences of saliency maps.

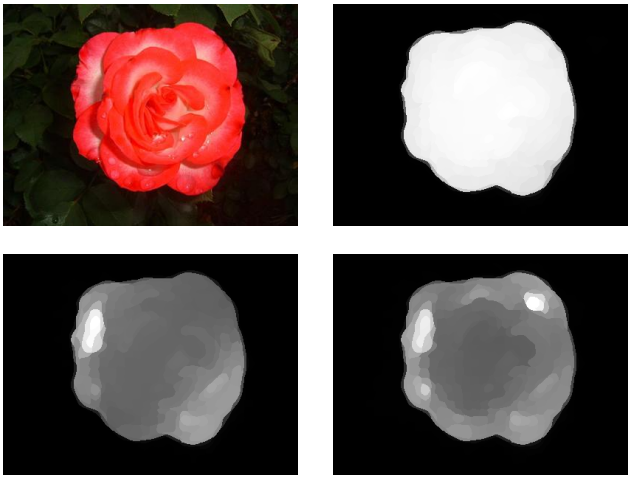

Figure 5. *Original image (top left) of a scene which one salient object and its corresponding saliency maps as we vary the number of eigenvectors considered for the superpixel embedding: 1 (top right), 2 (bottom left), 3 (bottom right). Map with 1 eigenvectors was chosen as the best by our score.*

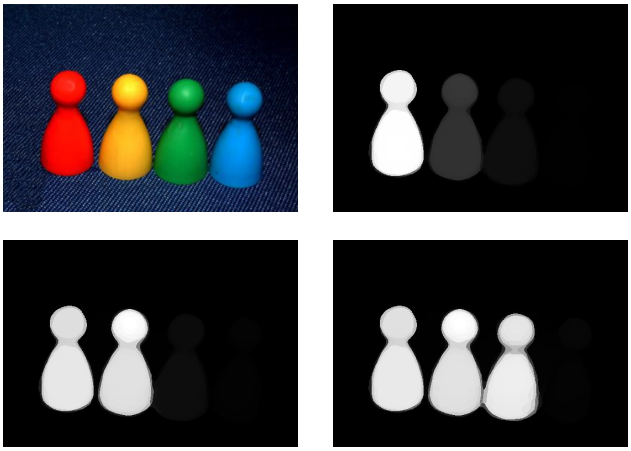

Figure 6. *Original image (top left) of a scene which multiple salient objects and its corresponding saliency maps as we vary the number of eigenvectors considered for the superpixel embedding: 1 (top right), 2 (bottom left), 3 (bottom right). Map with 3 eigenvectors was chosen as the best by our score.*

For the purpose of binarizing a floating point image, we will utilize the adaptive threshold proposed in [3] defined as twice the mean image saliency:

$$T_a = \frac{2}{W \times H} \sum_{x=1}^{m} \sum_{y=1}^{n} S(x,y) \tag{9}$$

Secondly, the following method first computes the desired number of eigenvectors to consider and subsequently constructing the saliency map in the following way:

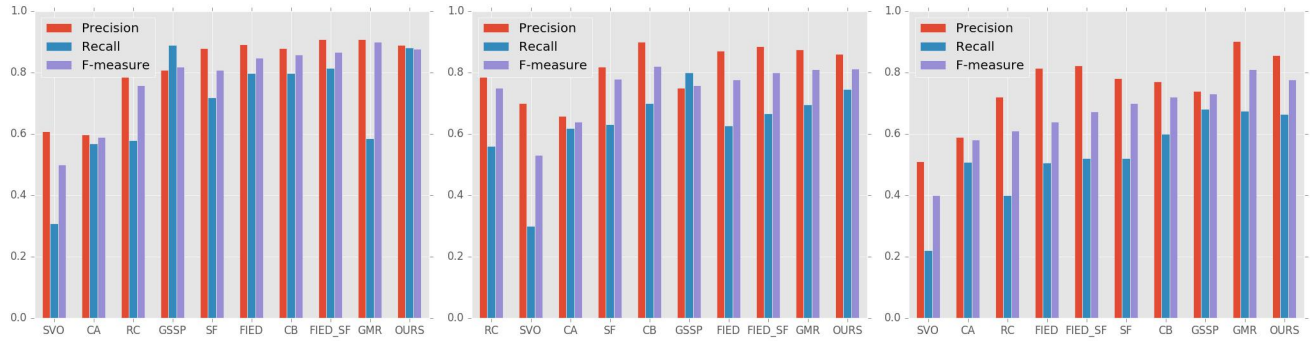

Figure 7. *Benchmarks. Performance of the various algorithms on the following datasets: MSRA [3] (left), ImgSal [15] (middle), and SED1 [16] (right).*

- First precompute the number, n, of eigenvectors to consider using equation 8.
- Compute the vector of Saliency scores, S, for the superpixels using the improved background prior.
- If the $n = 1$, then we are done otherwise repeat the following procedure for $n \geq 2$. Assume we have computed the saliency scores for the first $k, k < n$ dimensions, which we will call S_k. To incorporate the $k+1^{th}$ dimension in the computation of the final Saliency scores S, proceed as follows:
  - Compute the saliency scores for the $k+1^{th}$ dimension, S_{k+1} by computing the distance of each superpixel to the background node and rescaling the score between $[0,1]$.
  - Compute the threshold T_a^{k+1} based on S_{k+1} and extract the set of superpixels i for which it is true that $S_{k+1}^i \geq T_a^{k+1}$ and call the set $\mathscr{N}$.
  - For $i \in \mathscr{N}$, let $S_{k+1}^i := \max\{S_k^i, S_{k+1}^i\}$, otherwise $S_{k+1}^i := S_k^i$.
  - If $k+1 < n$, repeat the procedure, else construct the image saliency map.

Results

In order to provide a direct comparison of our algorithm with the original version proposed in [2], we evaluate the algorithm on the same three datasets used in the original paper: MSRA [3], SED1 [16] and ImgSal [15].

In order to benchmark the results of our algorithm, we will compare to the results obtained by Perazzi et al. (FIED) [2] and reporting the results published in [2] for the recent top-performing methods that include: context-aware saliency (CA) [6], context-prior (CB) [17], geodesic saliency(GSSP) [7], generic objectness (SVO) [8], global-contrast (GC)[9], graph manifold ranking (GMR) [12], and saliency filters (SF) [10] and their combination with FIED (FIED_SF).

Quantitative results and evaluation

To compare our algorithm with the above mentioned algorithms, we create binary maps from the computed saliency maps by first computing the adaptive threshold T_a of equation 9 proposed in [3] and assigning the values above and below T_a to the foreground and background classes respectively. We evaluate the proposed algorithm by computing the Precision, Recall and F-measure of the binary saliency maps compared to the ground truth maps. The F-measure is computed by

$$F_\beta = \frac{(1+\beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{10}$$

where $\beta^2 = 0.3$ to emphasize the importance of precision as seen in previous experimental setups [2, 3, 12].

The performance evaluation on the three datasets are shown in Figure 7, where we benchmark our algorithm with the augmented background model combined with the last multi-object extraction method (as it is the best performing foreground extraction method). As we can see from Figure 7, we achieve comparable results to the original algorithm for the MSRA and ImgSal datasets and a slight improvement for the SED1 dataset in terms of precision, which is defined as the fraction of retrieved pixels that actually belong to the foreground. Further, we see a good improvement in the recall value, which can be attributed to the improvement in extraction of multiple subjects, as recall is defined as the ratio of correctly detected pixels compared to the ground truth.

Limitations

Although we were able to augment the algorithm, the new algorithm still has difficulty with detecting foreground objects whose color is too similar to its surroundings. Furthermore, the first two foreground extraction-methods rely on the image metric to pick the best saliency map. A problem arises when taking the next step results in a larger increase in the average saliency than the decrease in the quality of the map (Silhouette score). In such a case, the algorithm might choose the worse map, and thus one of the possible avenues for future work is to explore alternative stopping criteria.

Conclusion

We proposed several improvements to a graph-based foreground detection method. First, we showed that by modeling the background to consist of several colors can lead to an improved foreground extraction. Furthermore, we have presented three approaches and shown their ability in segmenting multiple salient objects. The evaluation of the algorithm showed an equivalent/slightly improved results in precision and improvement in the recall over the original algorithm as can be seen from the benchmarking results.

As part of the future work we would like to gain a more thorough understanding of the spectral properties of the image graphs. Furthermore, we would like to explore several methods to enhance graph creation process, in which we could incorporate different shape priors to alternate the edge creation process. Several Deep Learning methods were recently developed which allow for processing of graphs, knows as Graph Convolutional Neural Networks [18]. We would like to further explore the application of such methods to foreground detection using a reduced image representation with the Region Adjacency Graph.

References

[1] A. Borji, D. N. Sihite, and L. Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 22(1):55–69, Jan 2013.

[2] Federico Perazzi, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. Efficient Salient Foreground Detection for Images and Video using Fiedler Vectors. In W. Bares, M. Christie, and R. Ronfard, editors, *Eurographics Workshop on Intelligent Cinematography and Editing*. The Eurographics Association, 2015.

[3] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Ssstrunk. Frequency-tuned Salient Region Detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1597 – 1604, 2009. For code and supplementary material, click on the url below.

[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*, 20(11):1254–1259, Nov 1998.

[5] Christof Koch and Shimon Ullman. *Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry*, pages 115–141. Springer Netherlands, Dordrecht, 1987.

[6] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, Oct 2012.

[7] Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun. Geodesic saliency using background priors. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 29–42, Berlin, Heidelberg, 2012. Springer-Verlag.

[8] Kai-Yueh Chang, Tyng-Luh Liu, Hwann-Tzong Chen, and Shang-Hong Lai. Fusing generic objectness and visual saliency for salient object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[9] Niloy J. Mitra Xiaolei Huang Shi-Min Hu Ming-Ming Cheng, Guo-Xin Zhang. Global contrast based salient region detection. pages 409–416, 2011.

[10] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.

[12] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. H. Yang. Saliency detection via graph-based manifold ranking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3166–3173, June 2013.

[13] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.

[14] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[15] J. Li, M. D. Levine, X. An, X. Xu, and H. He. Visual saliency based on scale-space analysis in the frequency domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):996–1010, April 2013.

[16] S. Alpert, M. Galun, A. Brandt, and R. Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):315–327, Feb 2012.

[17] Zejian Yuan Tie Liu Huaizu Jiang, Jingdong Wang and Nanning Zheng. Automatic salient object segmentation based on context and shape prior. In *Proceedings of the British Machine Vision Conference*, pages 110.1–110.12. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.110.

[18] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

Author Biography

Michal Kucer received his B.Sc. in Microelectronic Engineering and Applied Mathematics from the Rochester Institute of Technology (2014) and is currently pursuing a Ph.D. in Imaging Science at RIT. His work focuses on the development of methods for predicting the aesthetic value of images. His broader interests include Computer Vision, Remote Sensing and Machine Learning.

Nathan D. Cahill received his Ph.D. (2009) in Engineering Science from the University of Oxford, United Kingdom. He is the Associate Dean for Industrial Partnerships in the College of Science at RIT, where he is also an Associate Professor in the School of Mathematical Sciences. In addition, he is a graduate faculty member in RIT's Center for Imaging Science, and he is on the extended faculty of RIT's Ph.D. Program in Computing and Information Sciences. His personal areas of research include medical image analysis and computer vision.

Alexander Loui received his Ph.D. (1990) in Electrical Engineering from the University of Toronto, Canada. He is currently a Technical Lead and Senior Principal Scientist at Kodak Alaris in Rochester, NY. He is also an Adjunct Professor of ECE Department at Ryerson University and University of Toronto. He has been directing research on computer vision, video summarization, machine learning, image aesthetics, image event analysis, and multimedia applications. He is a Fellow of IEEE and SPIE.

David W. Messinger received a Bachelors degree in Physics from Clarkson University and a Ph.D. in Physics from Rensselaer Polytechnic Institute. He is currently a Professor, the Xerox Chair in Imaging Science, and Director of the Chester F. Carlson Center for Imaging Science at the Rochester Institute of Technology. His personal research focuses on projects related to remotely sensed spectral image exploitation using physics-based approaches and advanced mathematical techniques.