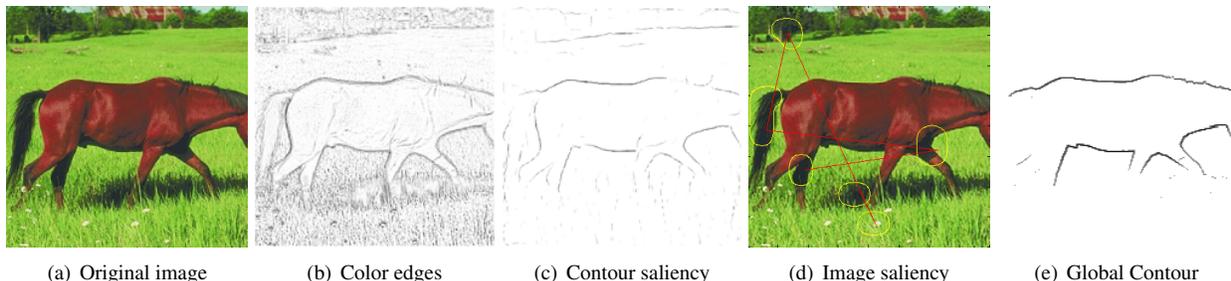


# Global Contours

Sujal Bista and Amitabh Varshney  
Department of Computer Science  
University of Maryland, College Park



**Figure 1:** The color edges, salient contours, image saliency and Global Contours are calculated for Figure (a). The color edges shown in Figure (b) and the salient contours shown in Figure (c) are calculated by Vonikakis et al. [2006]. Figure (d) shows the image saliency computed by the method of Itti et al. [1998]. In this image, salient regions are marked by yellow circles. Image saliency tends to focus on the local changes. Figure (e) shows the Global Contours calculated by our method. The darker regions in the contour lines are considered more salient. As we perform global computation, our calculated contours tend to reflect the most dominant property of the image.

## Abstract

We present a multi-scale approach that uses Laplacian eigenvectors to extract globally significant contours from an image. The input images are mapped into the Laplacian space by using Laplacian eigenvectors. This mapping causes globally significant pixels along the contours to expand in the Laplacian space. The measure of the expansion is used to compute the Global Contours. We apply our scheme to real color images and compare it with several other methods that compute image and color saliency. The contours calculated by our method reflect global properties of the image and are complementary to classic center-surround image saliency methods. We believe that hybrid image saliency algorithms that combine our method of Global Contours with center-surround image saliency algorithms will be able to better characterize the most important regions of images than those from just using contours calculated using bottom-up approaches.

**Keywords:** Laplacian eigenmaps, saliency, contours, global feature

## 1 Introduction

Contours are a natural way of representing and segmenting objects. Our visual system detects contours and segregates them from complex background for object recognition [Li and Gilbert 2002; Mundhenk and Itti 2005]. Better detection of contours can aid object segmentation and image saliency. By knowing which contours are globally distinct and important, we can better represent images and objects.

In this paper we present a multi-scale approach that extracts global and distinct contours from real images. Multiple scales of regular images are mapped onto the Laplacian space by using the first two Laplacian eigenvectors related to the two smallest non-zero eigenvalues. Laplacian eigenmaps, which is a well known machine learning technique for dimension reduction and data representation, transforms regular images into its natural basis [Belkin and Niyogi 2003]. The transformed images have similar patches

grouped together while dissimilar regions are partitioned apart [Shi and Malik 2000; Belkin and Niyogi 2003]. Pixels that are part of the contours are expanded in the Laplacian space. This property is used to detect contours that are globally distinct and the amount of expansion is used to measure their importance. We propose the measure of expansion of a pixel in the Laplacian space as a way of extracting globally-important properties that can be used for representing images. We validate our approach on several images and compare our method with other image and contour saliency algorithms. Due to the global nature of Laplacian eigenvectors, the contours that are calculated by our technique are successful in capturing the dominant properties of the image. We have found that Global Contours represent an image better than the contours calculated using bottom-up approaches. As can be seen from the results of our work Global Contours are complementary and greatly augment the saliency results computed using existing image saliency approaches.

## 2 Related Work

Contours are usually calculated in a bottom-up manner by combining small edges in various ways to generate long and smooth contours [Vonikakis et al. 2006; Zhu et al. 2008]. Since combining contours in these approaches is a local process, the final contours calculated by these methods might not necessarily be the globally best.

Williams and Jacobs [1997] present a method for computing a representation of objects using random walks. They compute illusory contours and occluded surface boundaries based on the probability that a particle following a stochastic motion will pass through the contour. Our scheme cannot compute illusory or occluded contours. However for the contours that are visible (which comprises most of the real-world images), our approach can not only compute the most dominant contours, it can also give us a measure of their importance.

Zhu et al. [2008] calculate salient contours in a bottom-up fashion by contour grouping at a single scale. Long contours detected by their approach are distinctive and capture more global information

than previous approaches. Vonikakis *et al.* [2006] construct an artificial network inspired by human visual system (HVS) that detects salient contours. Their method uses 60 kernels with different orientations. The kernels with equal excitation on the two lobes get high output which favors good continuation. We extract contours by a multi-scale method that uses the measure of expansion of a pixel in the Laplacian space.

Mundhenk and Itti [2005] present a method that computes visual saliency of an image by augmenting contour integration with the center-surround mechanism. They use a bottom-up approach to compute salient contours. Our method of computing contours is different because we do not use the bottom-up center surround method. We map the image into the Laplacian space and compute a global measure of contour saliency.

2D saliency maps have been applied to selectively compress [Privitera and Stark 1999] or shrink [Chen *et al.* 2003; Suh *et al.* 2003] images. DeCarlo and Santella [2002] use saliency determined from a person’s eye movements to simplify an image producing a non-photorealistic, painterly rendering. Torralba *et al.* [2006] have proposed a contextual guidance model that consists of two parallel pathways - one that computes local saliency features and the other that computes global (scene-centered) features. However, none of these methods determine contours that are global.

Shi *et al.* [2000] presented the method of normalized cuts for image segmentation. Their method uses a single eigenvector that corresponds to the smallest non-zero eigenvalue to segment the image. Our method uses two non-zero eigenvectors to compute the Global Contours. We believe better contours were achieved using two eigenvectors. This is further explained in the approach section.

Arbelaez *et al.* [2009] presented a very efficient algorithm that can produce hierarchical image segmentation from the output of any contour detector. They introduce Oriented Watershed Transform (OWT) and Ultrametric Contour Map (UCM) algorithm that are used to detect regions in an image based on the contours. They also allow user assisted segmentation. Rather than focusing on contours to generate good image segmentation, we focus on finding a small number of contours that are global in nature and can represent the image well.

### 3 Approach

In this section, we describe steps required to the compute Laplacian eigenvectors and the Global Contours.

#### 3.1 Computation of Laplacian Eigenvectors

The Laplacian eigenvectors of an image are computed using an approach similar to one described by Belkin *et al.* [2002].

The first step is to convert the input image into a graph. We take a regular 2D image with dimensions  $w$  and  $h$  as input. Let us denote a pixel at location  $i = (x_i, y_i)$  as  $p_i$  with color  $c_i$ . Each pixel of the image is considered as a node. We then construct an adjacency graph using  $\epsilon$ -neighborhoods. The two pixels  $p_i$  and  $p_j$  are connected by an edge if  $\|i - j\|_2 < \epsilon$ .

The second step is to compute the weight matrix. Entry  $W_{ij}$  of the weight matrix corresponds to the weight of an edge between pixels  $p_i$  and  $p_j$ . If  $p_i$  and  $p_j$  are two connected pixels, then the weight of their edge  $W_{ij}$  is computed by a heat kernel :

$$W_{ij} = e^{-\frac{(\|c_i - c_j\|_2 + \|i - j\|_2)}{t}}$$

where  $\|c_i - c_j\|_2$  is the difference in color value and  $\|i - j\|_2$  is the distance between pixels  $p_i$  and  $p_j$ . If two pixels are not connected, then the corresponding weight is set to 0. The weight matrix  $W$  is of dimension  $n \times n$  where  $(n = w \times h)$  as it contains an entry for an edge between every pair of pixels. Fortunately, due to the presence of the heat kernel that limits the presence of edges in the local vicinity of a node, this is a very sparse matrix.

The diagonal matrix  $D$  is defined as:

$$D_{ii} = \sum_j W_{ij}$$

The Laplacian matrix  $L$  is calculated as:

$$L = D - W$$

Finally we compute the eigenvalues and eigenvectors for the generalized eigenvalue problem:

$$Lf = \lambda Df$$

During the calculation of Laplacian eigenmaps a couple of user-defined values,  $\epsilon$  and  $t$ , are used. The local neighborhood is defined by  $\epsilon$ , which is the radius of the neighborhood. We have used  $\epsilon = 5.7$  for all the examples in this paper. The heat kernel variance is defined by  $t$  and its value is based on the type of input image. Smaller  $t$  values are used for smoother (less detailed) images. A small value of  $t$  results in a larger expansion of pixels around the contour region in the Laplacian space. We have used  $0.1 \leq t \leq 1.0$  for the various examples in this paper.

#### 3.2 Computation of Global Contours

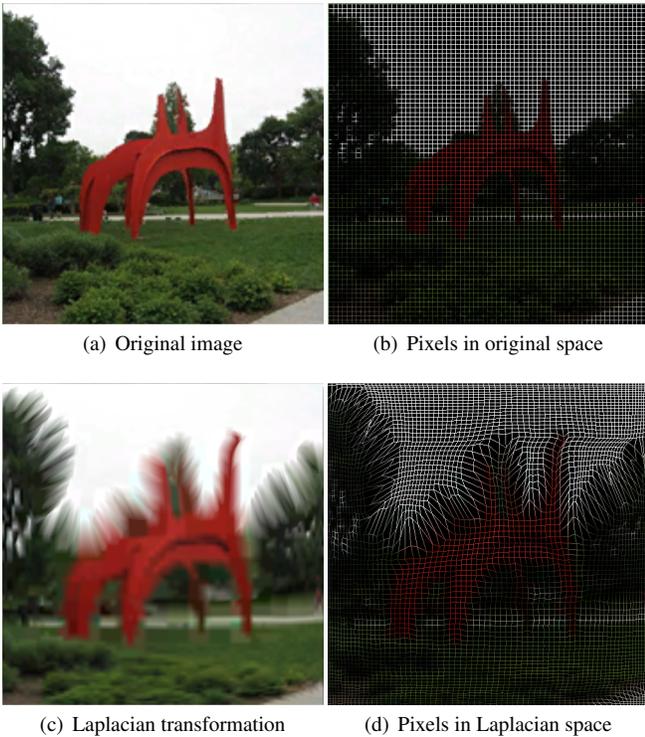
Each pixel of the image is mapped into the Laplacian space using the second and third eigenvectors. The first eigenvector is associated with the zero eigenvalue and has a constant value. We use the second and the third eigenvectors ( $E_2, E_3$ ) to define the Laplacian space coordinates.

We adjust the columns of the eigenvectors so that  $E_2$  corresponds to  $x$  and  $E_3$  corresponds to  $y$ . Generally, the pixel  $p_i$  in the original image with coordinates  $i = (x_i, y_i)$  will have coordinates  $(E_2[y_i w + x_i], E_3[y_i w + x_i])$  in the Laplacian space. Specifically, we define the Laplacian space coordinate for pixel  $p_i$  as  $L_i^s = (E_2[y_i w + x_i], E_3[y_i w + x_i])$ . However, sometimes eigenvectors may be flipped ( $x_i$  is mapped to  $E_3[y_i w + (w - x_i)]$  or  $y_i$  is mapped to  $E_2[(h - y_i)w + x_i]$  or exchanged ( $x_i$  is mapped to  $E_3[y_i w + x_i]$ ,  $y_i$  is mapped to  $E_2[y_i w + x_i]$ ) and this needs to be carefully addressed.

Figure 2 shows the mapping of an image to its Laplacian space coordinates. We note that the Laplacian space image is warped and the pixels are stretched accordingly. Also we can see that in the Laplacian space contours of the objects are stretched by varying amounts. Patches that share same properties are grouped together while the patches that are dissimilar are separated apart [Shi and Malik 2000]. The pixels that are expanded in the Laplacian space are associated with regions around contours and the expansion is caused by the graph partitioning. Our algorithm uses the measure of separation in the Laplacian space to compute Global Contours.

For each pixel  $p_i$  at location  $i = (x_i, y_i)$ , we measure the expansion  $\mathcal{E}_i^s$  in the Laplacian space at scale  $s$  by taking the maximum length between  $p_i$  and its two neighbors.

$$\mathcal{E}_i^s = \text{Max} \left\{ \begin{array}{l} \|L_i^s(x_i + 1, y_i) - L_i^s(x_i, y_i)\|_2, \\ \|L_i^s(x_i, y_i + 1) - L_i^s(x_i, y_i)\|_2 \end{array} \right.$$

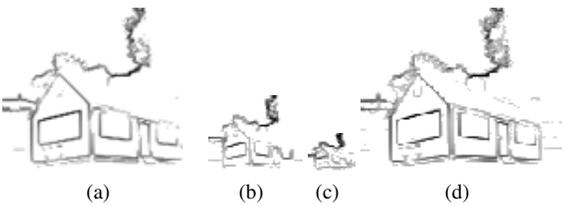


**Figure 2:** Illustration of conversion of an image to the Laplacian space. The original figure is shown in Figure (a). The pixels in the original image space are shown in Figure (b). The image in the Laplacian space is shown in Figure (c). Figure (d) shows the pixels in the Laplacian space. In Laplacian space, image is warped and the pixels that correspond to the salient contours are stretched.

This step is calculated at multiple scales and the weighted sum of the expansion value for each scale is added to compute the weighted expansion value for each pixel. For most of the images in this paper, three scales,  $s = [1.0, 0.5, 0.25]$ , are used.

$$\text{Weighted expansion value } \mathcal{W}_i = \sum_s w_s * \mathcal{E}_i^s$$

where  $w_s$  are weights used to combine different scales. Figure 3 shows the expansion value and the weighted expansion value calculated using three different scales. Using multiple scales allows the detection of contours that are salient at different scales.



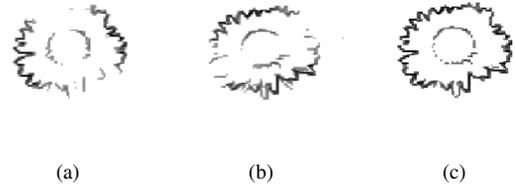
**Figure 3:** Expansion value is calculated at multiple scales. Figure (a), Figure (b) and Figure (c) show result of expansion value calculation at 1.0, 0.5 and 0.25 scales, respectively. The weighted expansion value shown in Figure (d) is the weighted sum of the expansion values calculated at all scales. Using multiple scales allows the detection of contours that are salient at different scales.

Now, the Global Contour value  $g_i$  is calculated by checking if the

weighted expansion value  $\mathcal{W}_i$  is greater than  $\delta$ . If  $\mathcal{W}_i$  is greater than  $\delta$ , then  $g_i$  is set equal to  $\mathcal{W}_i$  otherwise  $g_i$  is set to 0. The value of  $\delta$  corresponds to the normalized width of a pixel in the original image space.

$$g_i = \begin{cases} \mathcal{W}_i, & \text{if } (\mathcal{W}_i \geq \delta) \\ 0, & \text{otherwise} \end{cases}$$

While calculating expansion value  $\mathcal{E}_s$ , we use the first two eigenvectors. Figure 4 shows the result of using just the first, the second, and both eigenvectors. By using both eigenvectors, we are able to capture more contours which is not possible by using a single eigenvector.



**Figure 4:** Figure (a) shows the expansion of the pixels in Laplacian space when just the first eigenvector that corresponds to the smallest non-zero eigenvector is used. Figure (b) uses only the second eigenvector. Figure (c) uses both eigenvectors. From the figure, we can observe that using both eigenvectors produces better result than either one.

## 4 Results and Discussion

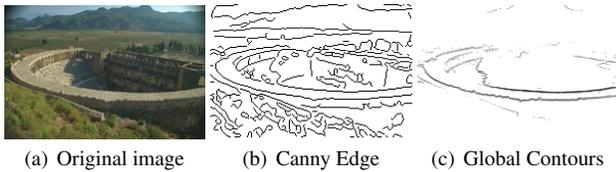
We illustrate the computation of distinctive and dominant Global Contours by first showing a simple example. Figure 5 shows the original image and the Global Contours calculated using our method. The color of the contour indicates the importance (darker region is associated with greater importance). The contours around the eagle and the branch are regarded as globally significant by our method.



**Figure 5:** Figure (b) shows contours calculated using our algorithm for Figure (a). Our algorithm is able to detect contour for both the eagle and the branch as being globally significant.

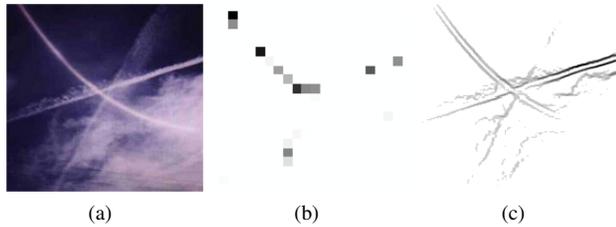
One advantage of applying Global Contours to the images is that the resulting contour captures global properties and is less susceptible to noise. To show this, we compare our method with Canny-edge detector as shown in Figure 6. The Canny edge detection algorithm is really effective in finding all the edges that are present in the image while our method only computes contours that are of global significance.

Next, we study our method on an image where background and foreground are hard to separate. Figure 7 shows the original image,



**Figure 6:** A comparison between the Canny edge detector and our method of Global Contour. Figure (b) is the result of applying Canny edge detection algorithm. Figure (c) shows the Global Contours of the same image using our method. As we can see, Canny edge detector computes every edge while our method is more selective and chooses contours that are of global significance.

the saliency map (discussed later in this section), and the Global Contours of clouds. Note that rather than considering the entire trail as salient, the image saliency algorithm only selects few patches of the trail as salient. Our method of Global Contours correctly identifies the entire trails as salient.



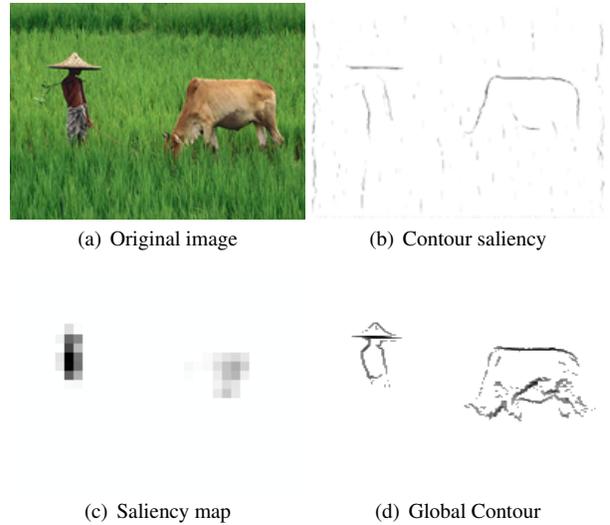
**Figure 7:** Figure (b) shows the saliency map of the image shown in Figure (a) using Itti et al.'s method [1998]. The darker regions are considered more salient. Figure (c) shows the contours calculated using our algorithm. In the original image, the background and the foreground distinction is not easy to make and yet our method computes a reasonably significant contours well.

Next, we compare our method with the model used by Vonikakis et al. [2006]. Figure 1, Figure 8, and Figure 9 show the comparison using three different images. Vonikakis et al. use a bottom-up approach to compute salient contours. This approach tends to capture more local contrast characteristics than the global property of the image. It can be seen in Figure 1(c), Figure 8(b), and Figure 9(c) where more noise is present and often contours are not detected even when they seem very obvious.

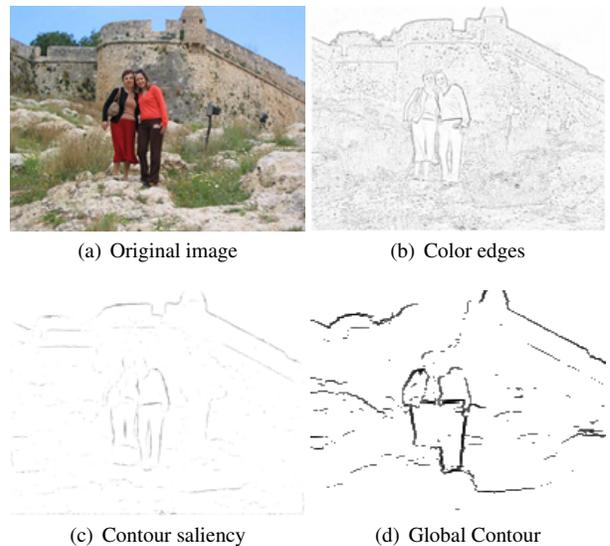
Sometimes our method outputs very small edges as Global Contours. It can be seen in Figure 9(d). This is because we do not currently threshold contours based on the contour length. We hope to address this by either removing or merging the contours that are shorter than a certain threshold.

We compare our method of computing Global Contours with the classical image saliency algorithm presented by Itti et al. [1998]. For this comparison we use the SaliencyToolbox which is re-implementation of work by Itti et al. [2009]. In Figure 10, input image, its saliency map, final image saliency, and the Global Contours are shown. Saliency map (Figure 10(b)) and image saliency (Figure 10(c)) show the patches that are most salient computed using Itti et al.'s method. Note that the selected salient regions might have different contrast or orientation locally but globally they are insignificant. On the other hand, the Global Contours are better able to capture the essence of the image.

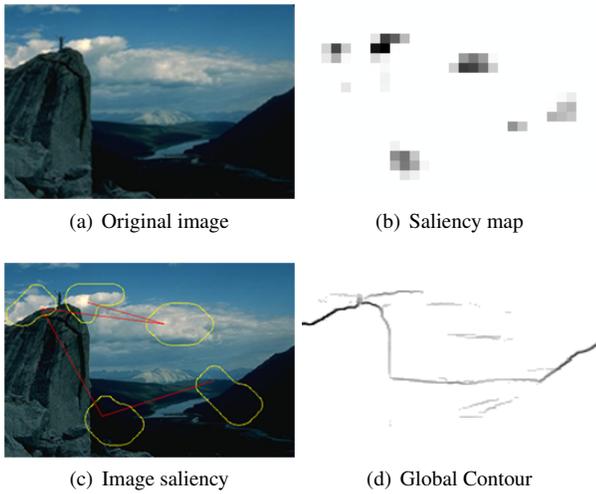
We have also compared our model with image saliency algorithms which integrate contour information. Saliency model presented



**Figure 8:** A comparison with the contour saliency model. The input image is shown in Figure (a). Figure (b) shows the salient contours calculated by Vonikakis et al. [2006]. Figure (c) shows the saliency map of the image. Figure (d) shows the Global Contours calculated by our method. The darker regions in the contours are considered more salient. Our method is able to detect features such as the hat better than the contour saliency algorithm.



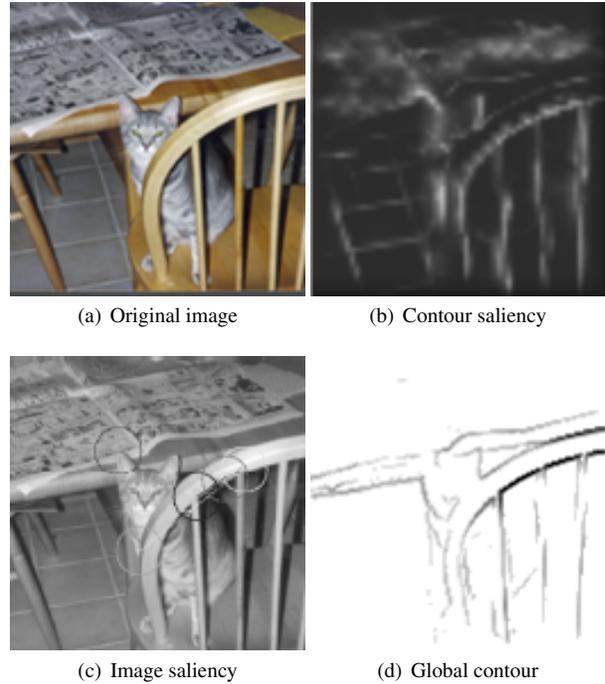
**Figure 9:** A comparison with the contour saliency model. The input image is shown in Figure (a). Figure (b) shows the color edges and Figure (c) shows the salient contours calculated by Vonikakis et al. [2006]. Figure (d) shows the Global Contours calculated by our method. The darker regions are considered more salient. Our algorithm is able to produce result that is less noisy than color edges algorithm and more complete than the contour saliency algorithm.



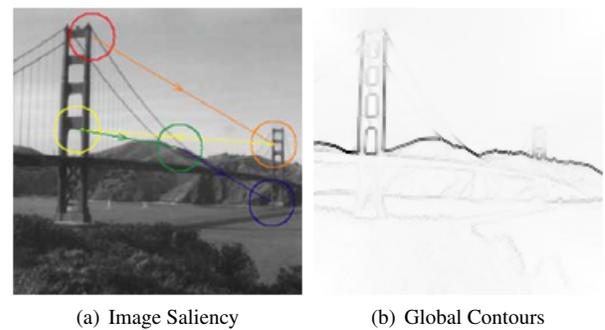
**Figure 10:** A comparison with the image saliency algorithm for the image in Figure (a) from the Berkeley Segmentation Dataset. Figure (b) is the saliency map computed using Itti et al. [1998] image saliency algorithm. In this image, darker regions are considered more salient. Figure (c) shows the final image saliency. Figure (d) shows Global Contours. The image saliency method marks the bright cloudy region and the dark region on the ground as salient as it relies on the local contrast. However these regions seem globally insignificant. Our method is able to identify contours that are globally significant.

by Mundhenk and Itti [2005] computes the entire image saliency by taking contours into account. The contours are calculated in a bottom-up fashion which is a local operation. In Figure 11, we compare our Global Contours to salient contours computed by Mundhenk and Itti [2005]. In Figure 11(b), the grout lines in the floor are marked as very salient. This is mainly due to the local nature of the bottom-up contour calculation approach. However in the final image saliency, shown in Figure 11(c), the floor is not regarded as highly salient. This observation matches the result of our Global Contour technique which suggest that the grout lines in the floor are not very salient.

Figure 12(a) shows the image saliency computed using the model presented by Mundhenk and Itti [2005]. The red region is the most salient followed by orange, yellow, green and blue. Figure 12(b) shows the Global Contours. In this case our Global Contour model is consistent with the image saliency.



**Figure 11:** A comparison between contour saliency computed using a bottom-up approach and our Global Contour approach for Figure (a). Figure (b) is the contour saliency computed by Mundhenk and Itti [2005] of the original Figure (a). Figure (c) is the image saliency computed by taking contours into consideration computed by Mundhenk and Itti. The salient regions are marked by circles. Figure (d) shows the Global Contours computed using our method. In these images darker regions are considered more salient. The grout lines in the floor are marked salient by the contour saliency algorithm. They are not considered salient by the image saliency algorithm as shown in Figure (c). This matches the result of our Global Contour method.



**Figure 12:** A comparison between image saliency algorithm with contour integration and our Global Contour method. Figure (a) is the result of using image saliency algorithm with contour integration by Mundhek et al. [2005]. The salient regions in this image are marked by colored circles. Figure (b) shows the Global Contours computed using our algorithm. Darker regions are considered more salient. These images show some similarity between the results computed using the image saliency algorithm and our Global Contour algorithm.

## 5 Conclusions and Future Work

We have shown that by using multiple scales and measuring the expansion of pixels in the Laplacian space, we can extract globally unique and important contours of the image. We have also compared the Global Contours computed using our method against the salient contours calculated using several bottom-up approaches. The Global Contours tend to capture global properties of the images that cannot be adequately captured by the contours calculated by local bottom-up approaches. We believe that the addition of the Global Contours to the existing image saliency algorithms will greatly help in enhancing the saliency computation. This will be an avenue for further research.

## Acknowledgments

This work has been supported in part by the NSF grants: CNS 04-03313 and CCF 05-41120. We gratefully acknowledge the support provided by the NVIDIA CUDA Center of Excellence award to the University of Maryland. Any opinions, findings, conclusion, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the research sponsors.

## References

- ARBELAEZ, P., MAIRE, M., FOWLKES, C., AND MALIK, J. 2009. From contours to regions: An empirical evaluation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2294–2301.
- BELKIN, M., AND NIYOGI, P. 2002. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. *Advances In Neural Information Processing Systems 1*, 585–592.
- BELKIN, M., AND NIYOGI, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput. Vol. 15*, 6, 1373–1396.
- CHEN, L., XIE, X., FAN, X., MA, W., ZHANG, H., AND ZHOU, H. 2003. A visual attention model for adapting images on small displays. *ACM Multimedia Systems Journal* 9, 4, 353–364.
- DECARLO, D., AND SANTELLA, A. 2002. Stylization and abstraction of photographs. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002)* 21, 3, 769–776.
- ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 20*, 11 (Nov), 1254–1259.
- LI, W., AND GILBERT, C. D. 2002. Global contour saliency and local colinear interactions. *Journal of Neurophysiology Vol. 88*, 5, 2846–2856.
- MUNDHENK, T. N., AND ITTI, L. 2005. Computational modeling and exploration of contour integration for visual saliency. *Biological Cybernetics Vol. 93*, 3 (Sep), 188–212.
- PRIVITERA, C., AND STARK, L. 1999. Focused JPEG encoding based upon automatic preidentified regions of interest. In *Proceedings of SPIE, Human Vision and Electronic Imaging IV*, 552–558.
- SHI, J., AND MALIK, J. 2000. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22, 8 (Aug), 888–905.
- SUH, B., LING, H., BEDERSON, B. B., AND JACOBS, D. W. 2003. Automatic thumbnail cropping and its effectiveness. *CHI Letters (UIST 2003)* 5, 2, 95–104.
- TORRALBA, A., OLIVA, A., CASTELHANO, M. S., AND HENDERSON, J. M. 2006. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological Review Vol. 113*, 4 (October), 766–786.
- VONIKAKIS, V., ANDREADIS, I., AND GASTERATOS, A. 2006. Extraction of salient contours in color images. In *SETN*, 400–410.
- WALTHER, D. B., 2009. Saliency toolbox. <http://www.saliencytoolbox.net>. [Online; accessed Dec-2009].
- WILLIAMS, L. R., AND JACOBS, D. W. 1997. Stochastic completion fields: a neural model of illusory contour shape and salience. *Neural Comput. Vol. 9*, 4, 837–858.
- ZHANG, B., AND ALLEBACH, J. 2008. Adaptive bilateral filter for sharpness enhancement and noise removal. *Image Processing, IEEE Transactions on* Vol. 17, 5 (May), 664–678.
- ZHU, Q., WANG, L., WU, Y., AND SHI, J. 2008. Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV08, II*: 774–787.