# MAIN SUBJECT DETECTION VIA ADAPTIVE FEATURE SELECTION

*Cuong T. Vu and Damon M. Chandler*

Image Coding and Analysis Lab
Oklahoma State University
Stillwater, OK 74078 USA
{cuong.vu, damon.chandler}@okstate.edu

## ABSTRACT

In this paper we present an algorithm which uses adaptive selection of low-level features for main subject detection. The algorithm first computes low-level features such as contrast and sharpness, each computed in a block-based fashion. Next, the algorithm quantifies the usefulness of each feature by using both statistical and geometric information measured across blocks. Finally, the saliency of each block is determined via a weighted linear combination of the features, where the weights are chosen based on each feature's estimated usefulness. Our results demonstrate that the adaptive nature of this algorithm allows it to perform competitively with other techniques, while maintaining very low computational complexity.

***Index Terms***— Main subject detection, low-level feature, adaptive feature selection, block-based.

## 1. INTRODUCTION

Most photographers convey their ideas via one or more main subjects in their photos. Locating the main subject in an image can be very useful for a variety of image processing applications. For example, main subject detection (MSD) plays a key role in auto cropping a photo [1]. In applications such as image compression and unequal error protection, the ability to find the main subject would allow one to devote more bits to that region. Studies have also shown that MSD can be useful for object recognition (e.g., [2]).

While a human can effortlessly identify the main subject and other salient objects in an image, MSD is quite challenging for a computer. Researchers have proposed some methods to detect the main subject in an image. For example, the method of Luo *et al.* [3] employs segmentation, perceptual clustering, and then feature extraction; the features are finally combined via a Bayesian network. Ma *et al.* [4] perform MSD based on local contrast analysis followed by fuzzy growing. A recent algorithm by Liu *et al.* [5] uses three features: (1) multiscale contrast, (2) a feature based on center surround histograms, and (3) a feature based on color spatial distribution; these features are combined via a Conditional Random Field. There also exist various algorithms which have been designed to predict visual fixation points (e.g., [6, 7]). Such points can also be useful for locating the main subject (see, e.g., [7]).

Although the human visual system (HVS) operates by using a variety of low-level and high-level features, we argue that the HVS is also effective because it can adaptively determine which features to use. In this paper, we build an algorithm to model adaptive selection of low-level features for MSD. Our algorithm first computes five low-level features for each block of the input image. We then estimate the utility of each feature based on statistical and geometric properties of the collection of each feature across blocks. The five features are then adaptively combined (per block, based on estimated utility) to generate a baseline saliency map. Two stages of feature refinement are then employed to generate a final saliency map, which is then used for MSD.

This paper is organized as follows: Section 2 describes the features and the baseline saliency map. Section 3 explains two stages of refinement. Section 4 shows our results and compares with other approaches. General conclusions are provided in Section 5.

## 2. BASELINE SALIENCY MAP

### 2.1. Features

Viewing MSD as a low-level vision problem, we might choose an object as the main subject because it is in focus, is different from the background in color, lightness or contrast, or has more edge pixels than others. Therefore, to perform main subject detection, we first measure five low-level features for each block in the input image. The features are lightness distance, color distance, contrast, sharpness, and edge strength. In this section, we describe how each feature is computed.

Let $\mathbf{X}$ denote the $M_1 \times M_2$ pixel input image and let $f_i(\mathbf{X})$ denote the $i^{th}$ feature map where each pixel value in $f_i(\mathbf{X})$ denotes the feature value measured for the corresponding block in $\mathbf{X}$. We divide $\mathbf{X}$ into blocks of size $m \times m$ with 50% overlap between neighboring blocks. Let $\mathbf{x}$ denote an block of $\mathbf{X}$.

#### *2.1.1. Lightness and Color Distance*

Let $f_1(\mathbf{x})$ denote the Euclidean distance between the average lightness of block $\mathbf{x}$ and the average lightness of the back-

ground. Let $f_2(\mathbf{x})$ denote the Euclidean distance between the average color of block $\mathbf{x}$ and the average color of the background. Here the average lightness and color of the background is taken as the average lightness and color of the image. These two features are given by

$$f_1(\mathbf{x}) = \left|\bar{L}^*(\mathbf{x}) - \bar{L}^*(\mathbf{X})\right| \tag{1}$$

$$f_2(\mathbf{x}) = \sqrt{\left(\bar{a}^*(\mathbf{x}) - \bar{a}^*(\mathbf{X})\right)^2 + \left(\bar{b}^*(\mathbf{x}) - \bar{b}^*(\mathbf{X})\right)^2} \tag{2}$$

where $\bar{L}^*$, $\bar{a}^*$, $\bar{b}^*$ denote the average $L^*$, $a^*$, $b^*$ measured in the CIE 1976 ($L^*$, $a^*$, $b^*$) color space (CIELAB).

#### *2.1.2. Contrast*

Let $f_3(\mathbf{x})$ denote the RMS luminance contrast of block $\mathbf{x}$ given by

$$f_3(\mathbf{x}) = \begin{cases} \sigma_{\mathbf{l}(\mathbf{x})}/\mu_{\mathbf{l}(\mathbf{x})}, & \mu_{\mathbf{l}(\mathbf{x})} > 0 \\ 0, & \mu_{\mathbf{l}(\mathbf{x})} = 0 \end{cases} \tag{3}$$

where $\mathbf{l}(\mathbf{x}) = (k\mathbf{x})^\gamma$ denotes the luminance-valued block, with $k = 0.02874$ and $\gamma = 2.2$ assuming sRGB display conditions. The quantities $\sigma_{\mathbf{l}(\mathbf{x})}$ and $\mu_{\mathbf{l}(\mathbf{x})}$ denote the standard deviation and the mean of $\mathbf{l}(\mathbf{x})$, respectively.

#### *2.1.3. Sharpness*

Let $f_4(\mathbf{x})$ denote the relative sharpness of block $\mathbf{x}$. From $\mathbf{X}$, we first compute a sharpness map $\mathbf{S}$ as described in the Appendix. The feature $f_4(\mathbf{x})$ is then given by

$$f_4(\mathbf{x}) = \mu_{\mathbf{s}} = \frac{1}{m^2}\sum_j s_j \tag{4}$$

where $\mathbf{s}$ is the $m \times m$ block of $\mathbf{S}$ corresponding to the same location as $\mathbf{x}$ in $\mathbf{X}$.

#### *2.1.4. Edge Strength*

Let $f_5(\mathbf{x})$ denote the relative edge strength of block $\mathbf{x}$. Let $\mathbf{E}$ denote the binary edge map computed by running the Roberts edge detector [8] of $\mathbf{X}$. The feature $f_5(\mathbf{x})$ is then given by

$$f_5(\mathbf{x}) = \mu_{\mathbf{e}} = \frac{1}{m^2}\sum_j e_j \tag{5}$$

where $\mathbf{e}$ is the $m \times m$ block of $\mathbf{E}$ corresponding to the same location as $\mathbf{x}$ in $\mathbf{X}$.

#### *2.1.5. Center Weight Modification*

The main subject is usually located near the center of the image, therefore each feature $f_i(\mathbf{x}), i = 1, ..., 5$ is modified as:

$$\tilde{f}_i(\mathbf{x}) = f_i(\mathbf{x}) f_c(\mathbf{x}) \tag{6}$$

where $f_c(\mathbf{x})$ denotes the relative distance of block $\mathbf{x}$ from the center of the image. The quantity $f_c(\mathbf{x})$ is given by:

$$f_c(\mathbf{x}) = 1 - \frac{\sqrt{(r - M_1/2)^2 + (c - M_2/2)^2}}{\sqrt{(M_1/2)^2 + (M_2/2)^2}} \tag{7}$$

where $r$ and $c$ denote the row and column value of the top-left pixel of $\mathbf{x}$.

### 2.2. Adaptive feature selection based on statistics

Given all $f_i(\mathbf{X})$, we next compute weights which represent the utility of each feature based on the statistic of the feature map. Let $\alpha_i$ denote the statistic and $w_i$ denote the weight for each feature map $f_i(\mathbf{X})$. We define $\alpha_i$ as:

$$\alpha_i = \sigma_i^2 + \kappa_i \tag{8}$$

where $\sigma_i^2$ and $\kappa_i$ denote, respectively, the variance and kurtosis of $f_i(\mathbf{X})$ (i.e., the variance and kurtosis of all $f_i(\mathbf{x})$ measured for the $i^{\text{th}}$ feature).

After $\alpha_1, \ldots, \alpha_5$ are computed, these statistics are used to determine the weight $w_i$ for each feature map $f_i(\mathbf{X})$ via

$$w_i = \begin{cases} 1, & \text{if } \alpha_i = \tilde{\alpha}_1 \\ 2/3, & \text{if } \alpha_i = \tilde{\alpha}_2 \\ 1/3, & \text{if } \alpha_i = \tilde{\alpha}_3 \\ 0, & \text{otherwise} \end{cases}, \quad \text{where} \quad \tilde{\alpha} = \text{sort}\{\alpha_i\} \tag{9}$$

where the sorting operation is used to sort the statistics in descending order. The feature with the greatest statistic is assigned the weight of 1, the feature with the second greatest statistic is assigned the weight of 2/3, the feature with the third greatest statistic is assigned the weight of 1/3, and the other features are assigned a weight of 0.

Let $\mathbf{R}_\mathbf{X}$ denote the saliency map of $\mathbf{X}$. $\mathbf{R}_\mathbf{X}$ is computed as the weighted sum of all five features maps

$$\mathbf{R}_\mathbf{X} = \frac{\sum_i w_i \tilde{f}_i(\mathbf{X})}{\sum_i w_i} \tag{10}$$

where $\tilde{f}_i(\mathbf{X})$ is the normalized version of $f_i(\mathbf{X})$ given by

$$\tilde{f}_i(\mathbf{X}) = \frac{f_i(\mathbf{X}) - \min_{\mathbf{x}\in\mathbf{X}} f_i(\mathbf{x})}{\max_{\mathbf{x}\in\mathbf{X}} f_i(\mathbf{x}) - \min_{\mathbf{x}\in\mathbf{X}} f_i(\mathbf{x})}.$$

The final baseline saliency map that we use is $\tilde{\mathbf{R}}_\mathbf{X}$ which is the normalized version of $\mathbf{R}_\mathbf{X}$ whose values have been rescaled to occupy the range $[0, 1]$.

## 3. SALIENCY MAP REFINEMENT

Using $\tilde{\mathbf{R}}_\mathbf{X}$ from above, we refine the saliency map in two stages to achieve better MSD.

### 3.1. Stage 1

#### *3.1.1. Feature Modification*

In this first stage, we determine the rectangle that contains all values of $\tilde{\mathbf{R}}_\mathbf{X} > 1.5 \times \text{mean}(\tilde{\mathbf{R}}_\mathbf{X})$. Locations within this rectangle are considered an initial guess of the main subject.

The lightness and color distance of each block $\mathbf{x}$ are then recomputed in the same way as in Equation (1) and (2) except the background is now considered as the region outside of the rectangle. Again, all five features are then center weight modified as in Equation (7) except that $f_c(\mathbf{x})$ in this case denotes the distance of block $\mathbf{x}$ relative to the center of the rectangle.

*3.1.2. Adaptive feature selection based on cluster density*

For each feature map $f_i(\mathbf{X})$ we compute an index $\beta_i$ which denotes how clustered are the high value pixels of the feature map. $\beta_i$ is given by

$$\beta_i = \frac{\sum_{(r,c)\in P_i} \sqrt{(r_0 - r)^2 + (c_0 - c)^2}}{|P_i|^{1.25}} \qquad (11)$$

where $P_i$ is the set of all coordinates $(r, c)$ corresponding to locations in the $i^{th}$ feature map with values greater than 0.5. $(r_0, c_0)$ denotes the centroid coordinate of these locations.

After $\beta_1, \ldots, \beta_5$ are computed, these indexes are used to determine the weight $w_i$ for each feature map $f_i(\mathbf{X})$ via

$$w_i = \begin{cases} 1, & \text{if } \beta_i = \tilde{\beta}_5 \\ \frac{\tilde{\beta}_1 - \tilde{\beta}_4}{\tilde{\beta}_1 - \tilde{\beta}_5}, & \text{if } \beta_i = \tilde{\beta}_4 \\ 0, & \text{otherwise} \end{cases}, \quad \text{where} \quad \tilde{\beta} = \text{sort}\{\beta_i\} \qquad (12)$$

where, as in Equation (9), the sorting operation is in descending order. The new $\tilde{\mathbf{R}}_\mathbf{X}$ is computed as in Equation (10) with the new sets of feature maps and weights, and then normalized to occupy the range $[0, 1]$.

### 3.2. Stage 2

In the second stage, we recompute the bounding rectangle based on the refined $\tilde{\mathbf{R}}_\mathbf{X}$ computed in Stage 1. The bounding rectangle is selected to be the smallest rectangle which contains at least 75% of the values of $\tilde{\mathbf{R}}_\mathbf{X} > 2 \times \text{mean}(\tilde{\mathbf{R}}_\mathbf{X})$. Locations within this rectangle are considered a refined guess of the main subject.

Lightness and color distance of each block $\mathbf{x}$ are then recomputed via $f_1(\mathbf{x}) = -\left|\bar{L}^*(\mathbf{x}) - \bar{L}^*(\mathbf{F})\right|$ and $f_2(\mathbf{x}) = -\sqrt{(\bar{a}^*(\mathbf{x}) - \bar{a}^*(\mathbf{F}))^2 + (\bar{b}^*(\mathbf{x}) - \bar{b}^*(\mathbf{F}))^2}$, respectively, where $\mathbf{F}$ denotes the region of $\mathbf{X}$ within the rectangle. As in Stage 1, all five features are then center-weight modified based on the center of the refined rectangle.

Using the modified features, we repeat the adaptive feature selection based on cluster density as described for Stage 1. This adaptive feature selection gives rise to a final set of feature weights, and thus a final saliency map $\tilde{\mathbf{R}}_\mathbf{X}$. From $\tilde{\mathbf{R}}_\mathbf{X}$, we select the rectangle which contains all values of $\tilde{\mathbf{R}}_\mathbf{X} > 1.5 \times \text{mean}(\tilde{\mathbf{R}}_\mathbf{X})$. Locations within this rectangle are considered a final guess of the main subject.

## 4. RESULTS

To assess the performance of our MSD algorithm, we use 5000 images in *Image Set B* from the MSRA Salient Object Database [5]. These are 24 bits/pixel color images with sizes ranging from $222 \times 165$ to $400 \times 400$ pixels. Each image in this set contains only one main subject and has been consistently labeled by nine human observers. The ground truth rectangle surrounding the main subject is averaged from results of observers as described in [5].

### 4.1. Evaluation

We evaluate our results based on four criteria used in [5]: Precision, Recall and F-measure for region-based measurement, and Boundary Displacement Error (BDE) for boundary-based measurement. Precision/Recall is the ratio of correctly detected salient regions to the detected/ground truth salient regions. Let $D$ and $G$ denote the detected and ground truth salient regions, respectively. Then $Precision = \frac{A(D \cap G)}{A(D)}$ and $Recall = \frac{A(D \cap G)}{A(G)}$ where the $A(\cdot)$ operator computes the area of the region. The overall performance measurement F-measure is given as: $F_\alpha = \frac{(1+\alpha) \times Precision \times Recall}{\alpha \times Precision + Recall}$ with $\alpha$ = 0.5. The BDE is the displacement error between the boundaries of two rectangles (see [5]).

We compare our algorithm with three competing methods. The first two methods come from Yu Fei Ma *et. al* [4] and Tie Liu *et. al* in [5]. These two methods also output a rectangle. The third one is the Saliency Toolbox presented in [6]. Since this method outputs a saliency map, we draw a rectangle which contains 95% of the fixation points according to [5]. We also compare with results from using the set of optimized weights that we set in [9] for our five features to see the effect of adaptive vs. fixed feature selection.

| | Precision | Recall | F-Measure | BDE |
|---|---|---|---|---|
| **Yu Fei Ma et al.** | 0.55 | **0.93** | 0.62 | 40.6 |
| **Saliency Toolbox** | 0.66 | 0.83 | 0.68 | 33.4 |
| **Tie Liu et al.** | **0.83** | 0.82 | **0.80** | **21.0** |
| **Using fixed weights** | 0.67 | **0.85** | 0.68 | 28.5 |
| **Our algorithm** | **0.79** | 0.81 | **0.78** | **22.2** |

**Table 1**. Comparison of different MSD algorithms

Table 4.1 shows results of these three methods and our algorithm evaluated using four criteria described above. Note that Recall is not necessarily an appropriate measure for MSD, since a 100% Recall can be easily obtained by selecting the whole image. The main challenge in MSD is to simultaneously obtain high Precision and F-measure, and low BDE. As can be seen from this table, on these three criteria, our algorithm is the second best. Even though our algorithm does not perform as well as the method of Tie Liu *et al.*, our results are very competitive and we believe that we have an advantage in computational efficiency since we use only low-level features. These results also demonstrate that using adaptive feature selection brings great improvement to our algorithm vs. using the fixed weights from [9].

### 4.2. Representative Results

Figure 1 shows several examples with ground truth rectangles, our baseline and refined saliency maps, and final MSD rectangles.[1] Notice that the refined saliency map in (d) demonstrates a marked improvement over the baseline saliency map in (c). This refinement is a crucial step for MSD. Our detection results on other images in Figure 2 show that our approach produces quite good results. However, we also show in Figure

[1] Code from [4] and [5] are not available for us to make qualitative comparisons

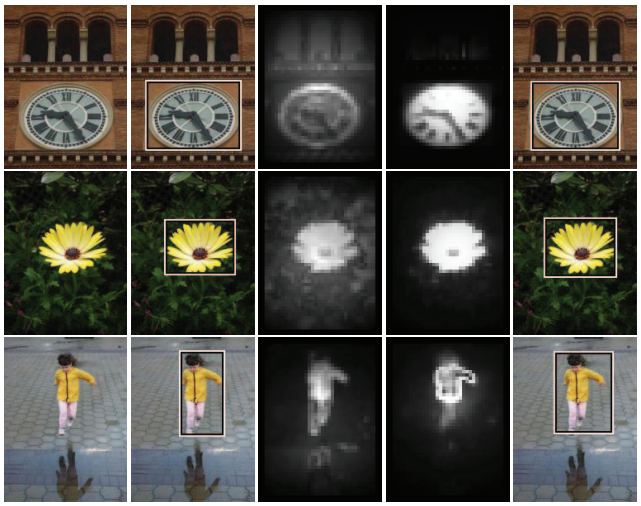


**Fig. 1**. Process of MSD and comparison with ground truth. (a) Original image. (b) Ground truth. (c) Our baseline map. (d) Our refined map. (e) Our MSD.

2 some failure cases to emphasize that MSD remains a challenging task which may require contextual and/or higher-level analyses.

## 5. CONCLUSION

In this paper, we presented an algorithm which adaptively uses low-level features for main subject detection. Our results demonstrate that relatively simple low-level features can be effective for MSD if these features are combined in an adaptive and interactive fashion (i.e., using adaptive weights and multiple stages of refinement). We believe that such adaptive feature selection can be a useful strategy for a variety of image processing applications.

## 6. APPENDIX

To measure local sharpness, we employ a block-based method which uses the slope of the image's local power spectrum and the local kurtosis of a whitened version of the image. The following steps summarize this sharpness metric.

For each $16 \times 16$ block:

1. Let $\Delta =$ slope of power spectrum.
2. Let $b_1 = 1 - \dfrac{1}{1 + e^{2.3\Delta + 5.8}}$.
3. Let $b_2 =$ kurtosis of the block in the whitened image. (the whitened image is obtained by filtering the original image with a radially symmetric filter whose magnitude spectrum increases proportionally with log frequency).
4. The relative sharpness value of the block is then given by $\sqrt{b_1^2 + b_2^2}$.

Note that following Step 3 the set of $\{b_2\}$ is normalized to span the range $[0, 1]$. Similarly the sharpness values computed in Step 4 are normalized to span the range $[0, 1]$.

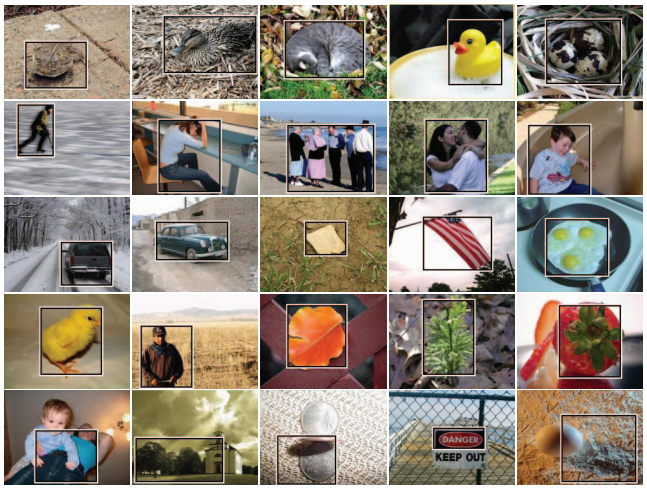


**Fig. 2**. Our detection results on some other images. Some failure cases are shown in the last row.

## 7. REFERENCES

[1] Jiebo Luo, "Subject content-based intelligent cropping of digital photos," in *IEEE International Conference on Multimedia and Expo 2007*, 2007.

[2] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?," *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–37–II–44 Vol.2, June-2 July 2004.

[3] Jiebo Luo, Amit Singhal, Stephen P. Etz, and Robert T. Gray, "A computational approach to determination of main subject regions in photographic images," *Image and Vision Computing*, vol. 22, no. 3, pp. 227 – 241, 2004.

[4] Yu-Fei Ma and Hong-Jiang Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, New York, NY, USA, 2003, pp. 374–381, ACM.

[5] T. Liu, J. Sun, N. N. Zheng, X. Tang, and H. Y. Shum, "Learning to detect a salient object," *Computer Vision and Pattern Recognition, CVPR '07. IEEE Conference on*, pp. 1–8, June 2007.

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

[7] O. L. Meur, P. L. Callet, D. Barba, and D. Thoreau, "A coherent computational approach to model bottom-up visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 802–817, 2006.

[8] L.G. Roberts, "Machine perception of three-dimensional solids," *Optical and Electrooptical Information Processing*, MIT Press, Cambridge, MA, 1965.

[9] Srivani Pinneli and Damon M. Chandler, "A Bayesian approach to predicting the perceived interest of objects," *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 2584–2587, Oct. 2008.