Hierarchical Saliency Detection
Supplementary Material

Qiong Yan Li Xu Jianping Shi Jiaya Jia
The Chinese University of Hong Kong
{qyan,xuli,jpshi,leojia}@cse.cuhk.edu.hk
http://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/

A. Technical details

A.1. Scale Estimation for Layer Extraction

We introduced a new metric in Sec. 3.1 of our paper, which is defined by the *encompassment* relation, to measure region scale. The metric is used to merge small-size regions. Since direct computing *encompassment* is relatively costly, we resort to a fast method by spatial convolution. Given a map M with each pixel labeled by its region index in the region list $\mathcal{R}$, we apply a box filter k_t of size $t \times t$, which produces a blurred map $k_t \circ M$ ($\circ$ denotes 2D convolution).

With computation of absolute difference $D_t = |M - k_t \circ M|$, we screen out regions in $\mathcal{R}$ with their scales smaller than t. The scale for a region R_i is smaller than t if and only if

$$\left(\min_y \{D_t(y) | y \in R_i\} \right) > 0, \tag{1}$$

where y indexes pixels in the image. It is based the observation that if all the label values for region R_i in M are altered after the convolution, R_i cannot encompass k_t. Thus, the scale of the region is smaller than t.

We present the scale estimation process in Algorithm 1. After obtaining regions whose scales are smaller than t, we merge each of them to its closest neighboring region in CIELUV color space. The merging process is shown in Algorithm 2.

Algorithm 1 Scale Estimation

```
input: Region list $\mathcal{R}$, scale threshold $t$
Create a map $M$ with each pixel labeled by its region index in $\mathcal{R}$;
Create a box filter $k_t$ of size $t \times t$;
$D_t \leftarrow |M - k_t \circ M|$;
$\mathcal{R}_t \leftarrow \emptyset$;
for each region $R_i$ in $\mathcal{R}$ do
    $x \leftarrow \min_y \{D_t(y) | y \in R_i\}$;
    If $x > 0$ then $\mathcal{R}_t \leftarrow \mathcal{R}_t \bigcup \{R_i\}$;
end for
output: Region list $\mathcal{R}_t$
```

A.2. Optimization in the Hierarchical Inference

The hierarchical inference model defined in our paper is

$$E(\mathcal{S}) = \sum_l \sum_i E_D(s_i^l) + \sum_l \sum_{i, R_i^l \subseteq R_j^{l+1}} E_S(s_i^l, s_j^{l+1}), \tag{2}$$

Algorithm 2 Region Merge

```
input: Region list $\mathcal{R}$, scale threshold $t$
repeat
    Get region list $\mathcal{R}_t$ by Algorithm 1;
    for each region $R_i$ in $\mathcal{R}_t$ do
        Find the neighboring region $R_j \in \mathcal{R}$ with the minimum Euclidian distance to $R_i$ in CIELUV color space;
        Merge $R_i$ to $R_j$;
        Set the color of $R_j$ to the average of $R_i$ and $R_j$;
    end for
until $\mathcal{R}_t = \emptyset$
output: Region list $\mathcal{R}$
```

where $\mathcal{S}$ is the set of all saliency variables $\{s_i^l\}$ we aim to estimate. Variable s_i^l denotes the saliency value for region i in layer $\mathcal{L}^l$. The data term $E_D(s_i^l)$ is defined as

$$E_D(s_i^l) = \beta^l ||s_i^l - \bar{s}_i^l||_2^2 \tag{3}$$

for each region R_i^l. The hierarchical term $E_S(s_i^l, s_j^{l+1})$ is

$$E_S(s_i^l, s_j^{l+1}) = \lambda^l ||s_i^l - s_j^{l+1}||_2^2 \tag{4}$$

for each pair of corresponding regions R_i^l, R_j^{l+1} in layers $\mathcal{L}^l$ and $\mathcal{L}^{l+1}$ respectively. They satisfy $R_i^l \subseteq R_j^{l+1}$.

By definition, energy function $E(\mathcal{S})$ forms a tree structure, whose nodes store unary energy $E_D(s_i^l)$ and edges are with the hierarchical energy $E_S(s_i^l, s_j^{l+1})$. Both of them are convex functions, thus the objective function can be efficiently solved using belief propagation. Two steps, i.e., bottom-up energy update and top-down optimization, are involved.

Bottom-up energy update In this step, we propagate local energies between connected nodes in the tree in a bottom-up way. For each node j in layer $\mathcal{L}^{l+1}$, energy is updated via

$$\widetilde{E}(s_j^{l+1}) = E_D(s_j^{l+1}) + \sum_{i \in Ch_j} \min_{s_i^l} [E_S(s_i^l, s_j^{l+1}) + \widetilde{E}(s_i^l)], \tag{5}$$

where Ch_j denotes all children of node j in the tree. For nodes in the bottom layer, we have $\widetilde{E}(s_j^1) = E_D(s_j^1)$. In Eq. (5), $E_D(s_j^{l+1})$ is pre-computed according to its definition in Eq. (3). Term $\min_{s_i^l}[E_S(s_i^l, s_j^{l+1}) + \widetilde{E}(s_i^l)]$ is easy to solve since $E_S(s_i^l, s_j^{l+1})$ is a quadratic term according to Eq. (4) and $\widetilde{E}(s_i^l)$ is also quadratic according to its definition. Solving the minimum expression plays the role of optimizing the energy configuration for child node s_i^l and obtaining its optimal expression w.r.t. its parent node s_j^{l+1}. By denoting the expression as $f_i^l(s_j^{l+1})$, Eq. (5) becomes

$$\widetilde{E}(s_j^{l+1}) = E_D(s_j^{l+1}) + \sum_{i \in Ch_j} [E_S(f_i^l(s_j^{l+1}), s_j^{l+1}) + \widetilde{E}(f_i^l(s_j^{l+1}))]. \tag{6}$$

The whole energy update in this step is therefore simple and efficient. After all energies are updated, we have the total energy represented using only saliency variables in the top layer.

Top-down optimization In this step, we propagate energies from top layers to the lower ones. First, for each node s_i^3 in the top layer, because $\widetilde{E}(s_i^3)$ solely depends on s_i^3 only, we directly optimize it to get optimal s_i^3. Denote the optimal solution as $(s_i^3)^*$. Then for each node i in layer $\mathcal{L}^l$, given the optimum calculated in the upper layer $\mathcal{L}^{l+1}$, we need to solve

$$\min_{s_i^l} [E_S(s_i^l, (s_j^{l+1})^*) + \widetilde{E}(s_i^l)], \tag{7}$$

where the first term is to bring down energy from upper layers, and the second term expresses the self-energy. Observing that it shares the similar form as the second term in Eq. (5), we directly use the optimal expression of s_i^l with respect to s_j^{l+1} derived in the previous step to calculate the optimal solution, i.e.

$$(s_i^l)^* = f_i^l((s_j^{l+1})^*). \tag{8}$$

After this step, we can obtain the global optimal value s_i^l for all nodes in the tree, guaranteed by the tree structure of our inference model.

B. More Results

We show in Fig. 1 more comparisons on the MSRA-1000 dataset [1] with several recent methods, including MZ [6], LC [8], GB [4], RC [2] and SF [7]. Abbreviations are the same as those in the paper. Our method shows advantages on handling objects with small structures or background containing fine texture patterns.

In Fig. 2, we show more comparisons on our new CSSD dataset with several recent methods, of which the implementations are public available. They include IT [5], FT [1], CA [3], HC [2] and RC [2]. Abbreviations follow those in the paper.

References

[1] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.

[2] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416, 2011.

[3] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, pages 2376–2383, 2010.

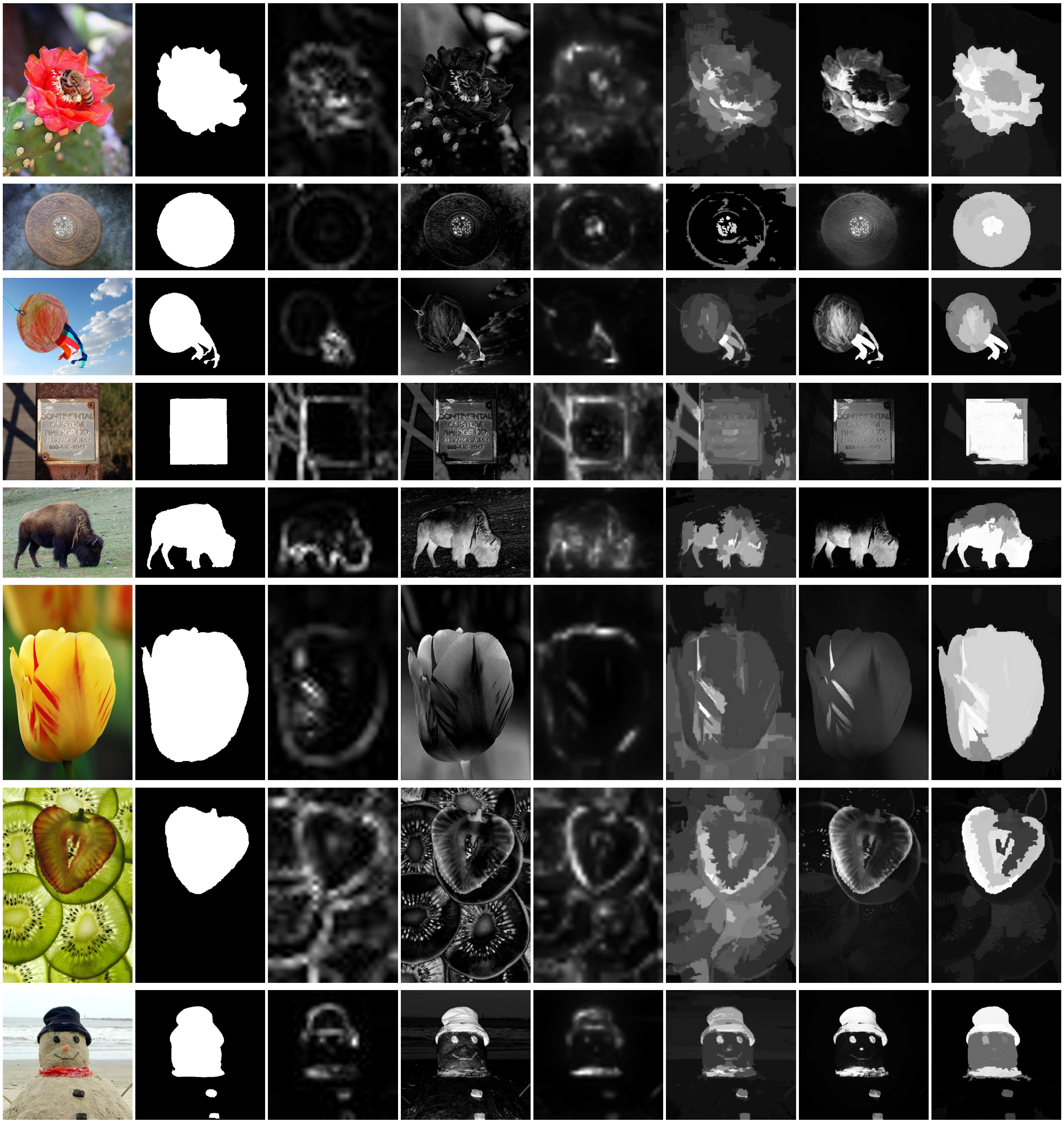
[4] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, pages 545–552, 2006.

[5] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, 1998.

[6] Y.-F. Ma and H. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *ACM Multimedia*, pages 374–381, 2003.

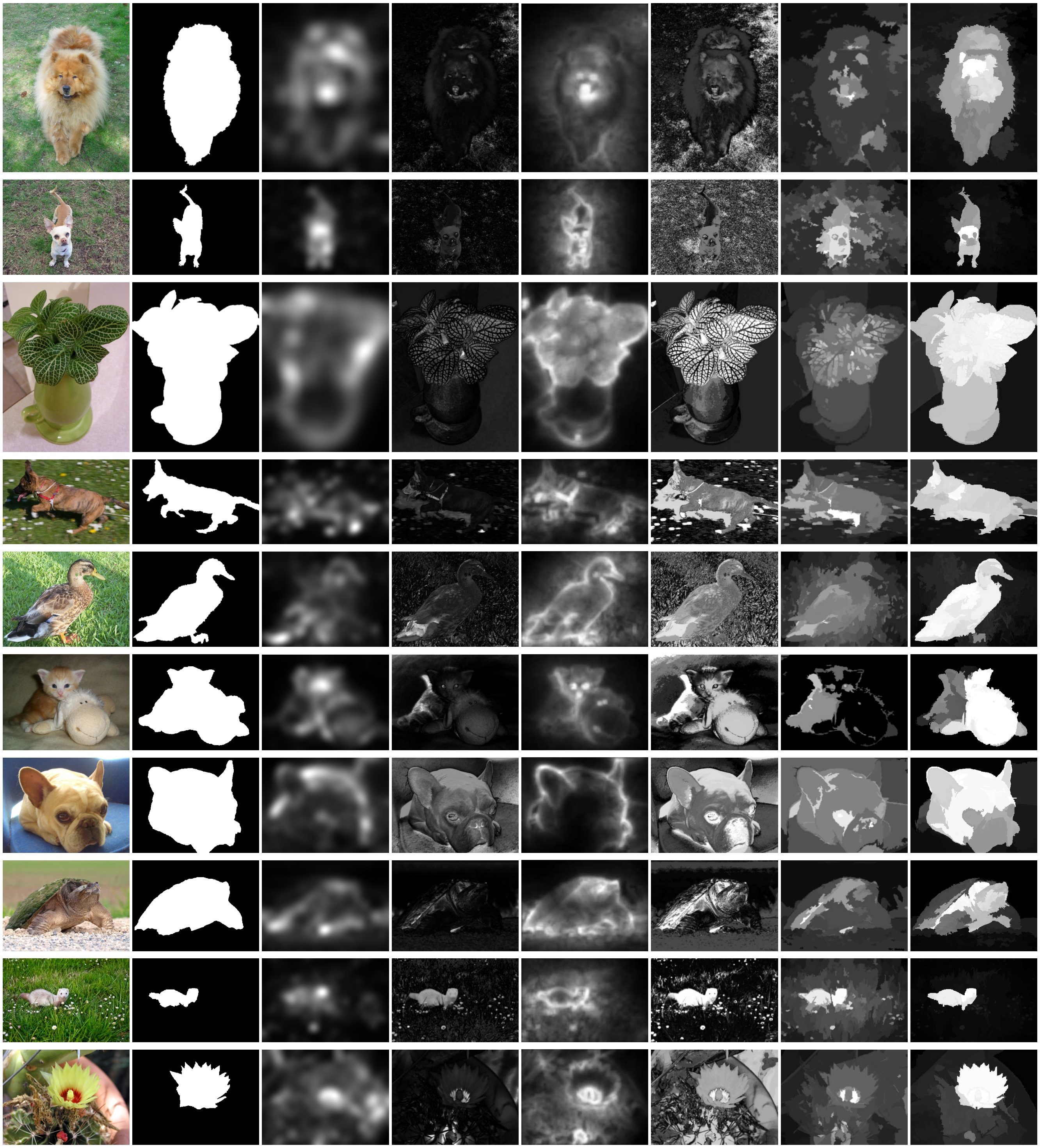
[7] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, 2012.

[8] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues. In *ACM Multimedia*, pages 815–824, 2006.



(a) Input (b) GT (c) MZ (d) LC (e) GB (f) RC (g) SF (h) Ours

Figure 1. More visual comparisons on MSRA-1000 dataset.



(a) Input (b) GT (c) IT (d) FT (e) CA (f) HC (g) RC (h) Ours

Figure 2. More visual comparisons on CSSD dataset.