Submodular Salient Region Detection

Zhuolin Jiang, Larry S. Davis
University of Maryland, College Park, MD, 20742
{zhuolin, lsd}@umiacs.umd.edu

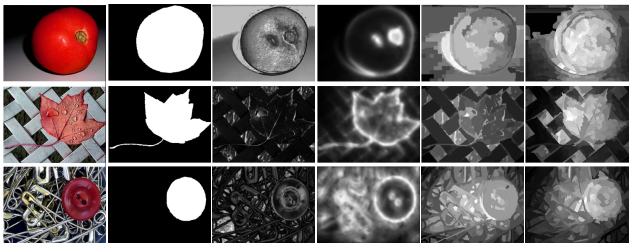
Abstract

The problem of salient region detection is formulated as the well-studied facility location problem from operations research. High-level priors are combined with low-level features to detect salient regions. Salient region detection is achieved by maximizing a submodular objective function, which maximizes the total similarities (i.e., total profits) between the hypothesized salient region centers (i.e., facility locations) and their region elements (i.e., clients), and penalizes the number of potential salient regions (i.e., the number of open facilities). The similarities are efficiently computed by finding a closed-form harmonic solution on the constructed graph for an input image. The saliency of a selected region is modeled in terms of appearance and spatial location. By exploiting the submodularity properties of the objective function, a highly efficient greedy-based optimization algorithm can be employed. This algorithm is guaranteed to be at least a $(e-1)/e \approx 0.632$*-approximation to the optimum. Experimental results demonstrate that our approach outperforms several recently proposed saliency detection approaches.*

1. Introduction

Visual saliency modeling is relevant to a variety of computer vision problems including object detection and recognition [29, 26], image editing [13, 4, 6] and image segmentation [16]. Most saliency models [2, 20, 4, 6, 8] are based on a *contrast prior* between salient objects and backgrounds. Saliency models map natural images into saliency maps, in which each image element (pixel, superpixel, region) is assigned a saliency strength or probability. These maps can then be converted into crisp segmentations using a variety of methods (*e.g.*, simple thresholding).

These approaches work well in images which have simple backgrounds or high contrast between foreground and background, but can fail in more complex images. For example, Figure 1 illustrates saliency detection results using four state-of-art algorithms [2, 6, 4, 26]. The three input images have increasingly complex background but all have high color contrast between objects and background. However, given the ground truth salient regions in Figure 1(b), even for the first simple example, these approaches either fail to separate the object from the background, as in Figures 1(c) and 1(e), or mostly outline the object but miss the interior as in Figure 1(d).



(a) Inputs (b) GT (c) FT [2] (d) CA [6] (e) RC [4] (f) LR [26]

Figure 1. Saliency detection results using four state-of-the-art approaches on three examples of increasing background complexity. (a) Input images; (b) Ground truth salient regions; (c)∼(e): Saliency maps using [2, 6, 4] with contrast priors; (f) Saliency map using [26] with a low-rank prior.

Using only a contrast prior has shortcomings. For example, a small region with high contrast might be considered to be noise by humans. Hence some approaches, such as [26, 28, 27] propose background priors to address this problem. [26, 28] represent an image as a low-rank matrix plus sparse noise, where the background is modeled by the low-rank matrix and the salient regions are indicated by the sparse noise (*i.e.*, *low-rank prior*). Natural images usually exhibit cluttered backgrounds, so models that make simplifying assumptions, such that the background lies in a low-dimensional space, might not perform well. For example, the poor saliency detection results in Figure 1(f) using the low-rank prior are due to the cluttered background.

We present a submodular objective function for efficiently creating saliency maps from natural images; these maps can then be used to detect multiple salient regions within a single image. The diminishing return property of submodularity has been successfully applied in various applications including sensor placement [18], facility location [24] and image segmentations [15]. Our objective function consists of two terms: a similarity term (between the selected centers of salient regions and image elements (superpixels) assigned to that center), and the 'facility' costs for the selected region centers. The first term encourages the selected centers to represent the region elements well. Hence it favors the extraction of high-quality potential salient regions. The second term penalizes the number of selected potential salient region centers, so it avoids oversegmentation of salient regions. It reduces the redundancy among selected salient region centers because the small gain obtained by splitting a region through the introduction of an extrane-

ous region center is offset by the facility cost. This high level prior is integrated with low level feature information into a unified objective function to identify salient regions. This is in contrast to previous approaches based on low level features [2, 4] or high level information only [29, 5], or heuristic integration approaches [13, 6] based on weighted averages on the saliency maps from low level features and high level priors. In contrast to some approaches [28, 27] which use uniform image patches to represent an image, our representation is based on super-pixels, which are less likely to cross object boundaries and lead to more accurately segmented salient regions. Unlike approaches that identify only one salient region in an image [7], our approach identifies multiple salient regions simultaneously without any strong assumptions about the statistics of the backgrounds [28]. The main contributions of our paper are:

- Salient region selection is modeled as the *facility location* problem, which is solved by maximizing a submodular objective function. This provides a new perspective using submodularity for salient region detection, and it achieves state-of-art performance on two public saliency detection benchmarks.
- The similarities between hypothesized region centers and their region elements are formulated as a labeling problem on the vertices of a graph. It is solved by finding a harmonic function on the graph, which has a closed-form solution.
- We present an efficient greedy algorithm by using the submodularity property of the objective function.
- We naturally integrate high-level priors with low-level saliency into a unified framework for salient region detection.

1.1. Related Work

Existing salient region detection approaches can be roughly divided into two categories: bottom-up and top-down approaches. Bottom-up approaches are data-driven based on low level features (*e.g.*, oriented filter responses and color), and usually are based on a *contrast prior*. Both local [20, 12] and global [4, 6, 2, 8, 11, 10] contrast priors have been investigated. Recently, [26, 28] decompose an image into a low-rank matrix representing the background (*low-rank prior*) and a sparse noise matrix indicating the salient regions by low-rank matrix recovery. [27] proposes to use the boundary prior, which assumes the image boundary is mostly background for saliency detection.

Top-down approaches make use of high level knowledge about 'interesting' objects to identify salient regions [29, 5, 14]. [29] learns interesting region features by dictionary learning and then generates the saliency map by modeling spatial consistency via a CRF model. [5] proposes a top-down saliency algorithm by selecting discriminant features from a pre-defined filter bank.

In addition, some approaches integrate multiple saliency maps generated from different features or priors to detect salient regions. The saliency maps are combined by weighted averaging, where the weights are predefined [6, 8], learned by a SVM [13] or estimated by a CRF [20]. Unlike previous approaches that are purely top-down or bottom-up, we integrate high level priors with low level information into a unified framework, which is graph-based and is optimized in a submodular framework.

2. Preliminaries

Facility Location: [17, 22] We solve a facility location problem to generate candidate regions for saliency-based segmentation. The formulation of the uncapacitated *facility location* problem is:

$$\begin{array}{ll} \max & \sum_{i\in I}\sum_{j\in J} c_{ij}\tilde{x}_{ij} - \sum_{j\in J} f_j\tilde{y}_j \\ s.t. & \sum_{j\in J}\tilde{x}_{ij} = 1, \tilde{x}_{ij} \le \tilde{y}_j, \forall i \in I, \forall j \in J \end{array} \quad (1)$$

where I is the locations of a set of clients and J denotes the potential sites for locating facilities. f_j is the cost of opening a facility at location j and c_{ij} denotes the profit made by satisfying the demand of client i by facility j. $\tilde{x}_{ij}$ and $\tilde{y}_j$ are binary variables. $\tilde{y}_j = 1$ if facility j is open and $\tilde{y}_j = 0$ otherwise; $\tilde{x}_{ij} = 1$ if the demand of client i is satisfied from facility j and $\tilde{x}_{ij} = 0$ otherwise. The *combinatorial* formulation of (1) is $\max_{A\subseteq J} Z(A)$, where $Z(A) = \sum_{i\in I}\max_{j\in A} c_{ij} - \sum_{j\in A} f_j$. Given I, J, c_{ij} and f_j, the goal is to find a subset A of facility locations from J and an allocation of each client to an open facility to maximize the overall profit.

Harmonic Function on a Graph: [9, 32] Suppose we have n ($n = l + u$) data points comprised of labeled data points $(x_1, y_1), ..., (x_l, y_l)$ with m class labels $y \in \{1...m\}$ and unlabeled data points $x_{l+1}, ..., x_{l+u}$. Graph-based semi-supervised learning can be modeled by constructing a graph $G = (V, E)$ with nodes V represent the n data points, with $L = \{1...l\}$ being labeled data points, and $U = \{l + 1...l + u\}$ being unlabeled data points and edges E represent similarities between them. These similarities are given by a weight matrix $W = [w_{ij}]$: w_{ij} is nonzero if edge $e_{i,j} \in E$. The task of assigning labels to U is solved by constructing a real-valued function: $h : V \to R$. The optimal h minimizes the quadratic energy function $\mathcal{D}(h) = \frac{1}{2}h^t \triangle h = \frac{1}{2}\sum_{e_{i,j}\in E} w_{ij}(h(i) - h(j))^2$, which is the combinatorial formulation of the Dirichlet integral. It is not difficult to show that the quadratic energy is minimized when $\triangle h = 0$, where $\triangle \equiv D - W$ is the combinatorial Laplacian matrix. D is the diagonal degree matrix, where $D_{ii} = \sum_j w_{i,j}$ is the degree of vertex i. A function that solves the Dirichlet problem is called a harmonic function and satisfies $\triangle h = 0$. The probability that a random walker first reaches a labeled node exactly equals the solution to the Dirichlet problem with boundary condi-

tions at the locations of the labeled nodes: the labeled node in question fixed to unity while the others are set to zero [9].

Let Y_L denote a label matrix for L of size $l \times m$, where m is the number of classes and $Y_L(i,j) = \delta(y_i, k)$. Given labeled nodes L and unlabeled nodes U, W is divided into 4 blocks: $W = \begin{bmatrix} W_{LL} & W_{LU} \\ W_{UL} & W_{UU} \end{bmatrix}$ (and similarly D, and the transition matrix $P = D^{-1}W$). The harmonic solution $h = \begin{pmatrix} h_L \\ h_U \end{pmatrix}$ satisfying $\triangle h = 0$ subject to $h_L = Y_L$ is:

$$h_U = (D_{UU} - W_{UU})^{-1} W_{UL} h_L = (I_{UU} - P_{UU})^{-1} P_{UL} h_L \quad (2)$$

where h_U is a $u \times m$ matrix of label values for U. Note that this is a closed-form solution that can be efficiently computed using matrix operations.

3. Submodular Saliency

There are three main steps in our approach: First, a set of potential region centers are extracted from an image. They serve as a set of potential facility locations (denoted by J). Second, given that set of potential region centers, we identify the final region centers and cluster superpixels into regions by solving the facility location problem. This provides a set of potential salient regions. We combine the high-level top-down priors with the low-level information in the optimization process. Finally, the saliencies of the potential salient regions and their constituent superpixels are computed from color and spatial location information.

3.1. Graph Construction

We represent an image as an undirected k-nearest-neighbor graph [1] $G = (V, E)$, where the vertices $v \in V$ are superpixels while the edges $e \in E$ model the pairwise relations between vertices. Figure 3(b) shows an example of superpixel segmentation for an input image. We extract a 3-D CIE Lab color feature descriptor for each superpixel: $X = [x_1, x_2 ... x_N]$, where N is the number of superpixels. Let v_i denote the i-th vertex and $e_{i,j}$ be the edge that connects v_i and v_j. The weight $w_{i,j}$ assigned to the edge $e_{i,j}$ is computed as: $w_{i,j} = \exp(-\beta d^2(x_i, x_j))$ if $e_{i,j} \in E$, otherwise $w_{i,j} = 0$. The normalization factor β is set to $\beta = 1/\sigma_i\sigma_j$. σ_i and σ_j are local scaling parameters for v_i and v_j respectively. σ_i is selected by using the local statistics of the neighborhood of v_i. A simple choice for σ_i in our experiments is $\sigma_i = d(x_i, x_k)$ as in [30], where x_k is the feature descriptor of the k-th neighbor of v_i.

3.2. Identifying A Set of Potential Region Centers

It is computationally too expensive to use the whole set V as the set, J, of potential region centers to identify the final region centers, denoted by A. For example, there are many 'wall' superpixels in Figure 3(b); no matter which is chosen as a region center, the region extracted is more or less the same. Thus we employ agglomerative clustering on G to obtain the hypothesis set J. J is generally less than 100 in our experiments. Then we evaluate the marginal gain of elements in J to iteratively construct the subset A. In Figure 3(c), the candidate set J is marked in blue.

3.3. Extraction of Potential Salient Regions

We model the problem of identifying high quality potential salient regions as selecting a subset, A, of J as the final region centers. A is regarded as the set of locations for opening facilities, and the similarities between elements of A and superpixels eventually assigned to elements of A as the profits made by satisfying the demand of clients by facilities from A. As discussed previously, this problem can be modeled as the *facility location* problem [22]. Let N_A denote the number of open facilities. With the constraint $N_A = |A| \leq K$, the combinatorial formulation of the facility location problem in [22] can be applied to our problem:

$$\max_A \mathcal{H}(A) = \sum_{i \in V} \max_{j \in A} c_{ij} - \sum_{j \in A} f_j$$
$$s.t. \quad A \subseteq J \subseteq V, \; N_A \leq K \quad (3)$$

where c_{ij} denotes the similarity between a vertex v_i (considered as clients) and its potential region center v_j (considered as facilities), and the cost f_j of facility opening is fixed to λ. The overall profit $\mathcal{H} : 2^J \rightarrow R$ on the graph G is a submodular function [24, 22].

The first term encourages the similarity between v_i and its assigned region center to have the greatest value. The optimization favors region centers that are visually similar to their 'clients'. The second term is the penalty for extraneous facilities. It mitigates against fragmentation of visually homogenous regions, since the small gain in visual similarity to marginally 'productive' region centers is more than offset by the cost of opening such a facility. This makes A both representative (*i.e.*, centrality) and compact (*i.e.*, diversity).

K is the maximum number of salient regions that the algorithm might identify, and is a parameter specified by the user. Generally, fewer than K locations are chosen because the marginal gain does not outweigh the facility cost.

3.3.1 Computation of c_{ij}

c_{ij} serves as the profit made by satisfying the demand of client i from a facility at location $j \in J$. It should be computed before the optimization of (3) since it is an input variable for the facility location problem. Given a $j \in J$, we discuss how to compute the similarities c_{ij}. The following is performed for each j in J. Since not all nodes in G should be assigned to any j, we add a background node v_g to G with label 0 so $v_i \in U$ can also be assigned to background. v_g is fully connected to all the nodes in G. The weight $w_{i,g}$ for the edge $e_{i,g}$ is a constant z [2]. z can be

[1] We select k nearest neighbors for each superpixel from a set of spatially proximate candidates based on feature similarity.

[2] $z = 0.1$ is used for all our experiments.

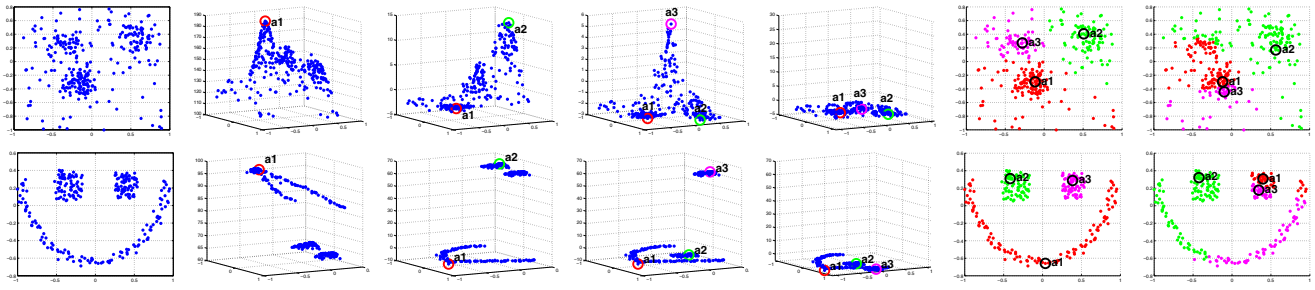


Figure 2. Examples of facility location and facility assignment results (clustering) on two synthetic datasets. The selected region centers (facility locations) are marked as circles. Our approach successfully captures the structure of the data. (a) Input datasets. (b)∼(e): a_1, a_2, a_3 are selected based on their marginal gains in $\mathcal{H}(A \cup \{a\}) - \mathcal{H}(A)$ in three iterations. The selected A is representative and compact. (f) Facility assignment results by using harmonic solution to compute c_{ij}. Different colors denote different clusters. (g) Facility assignment results by simply using naive weight w_{ij} as c_{ij}.

Figure 4. Examples of high-level prior maps. (a) Input image; (b) Center prior map; (c) Face prior map; (d) Color prior map; (e) Final combined and smoothed prior map.

viewed as a parameter to control the trade-off between centrality and diversity of A. If z is large, the number of nodes assigned to the background becomes larger, and only neighbors within a small distance of j can have high similarity to j. Hence the possibility of a potential region center close to A being selected increases during the subsequent iteration of the optimization.

c_{ij} can be computed by finding the harmonic function on the graph G with the labeled nodes L set to node j with label 1 and the background node with label 0, while the other nodes in G are the unlabeled nodes U. This is a two-class labeling problem and (2) can be used to compute h_U for U. c_{ij} is the probability that a random walker starting from v_i, will reach j before reaching the background node [9, 32]. For a two-class problem, we have $h_L = (1, 0)^t$ as in [32]. With $c_{gj} = 0$ and $c_{jj} = 1$, we can also obtain $c_{ij} = h_U \in R^{u \times 1}$ for $\forall i \in U$. The computation is conducted $|J|$ times, each time taking one node from J and the background node. We can obtain c_{ij} for $\forall i \in V$ and $\forall j \in J$. c_{ij} is fixed during the subsequent optimization of (3).

3.3.2 High-level Prior Integration

We can incorporate high level priors into the computation of c_{ij}, so that solving (3) will tend to make A focus on high probability areas indicated by the high-level prior map. The high level priors we used are the following:

Face Prior. People often pay attention to objects such as faces [26, 13, 6]. Here, the detected face regions Λ are assigned higher probabilities to generate the face prior map $p_f(x) = \sigma_1$, for $x \in \Lambda$; otherwise $p_f(x) = 0$. σ_1 is a constant obtained by simple thresholding the output of a face detector.

Center Prior. People taking photographs generally frame an object of interest near the image center. Hence, we generate a prior map based on the distance of a pixel to the image center $\hat{c}$ using: $p_l(x) = \exp(-d^2(x, \hat{c})/\sigma_2)$, where σ_2 is set to $(2\langle d^2(x, \hat{c})\rangle)^{-1}$, where $\langle\cdot\rangle$ denotes expectation over all pairwise distances, as in [25].

Color Prior. The warm colors such as red and yellow are more attractive to people [13, 26]. We generate a pair of 2-D histograms, H_s and H_b, in the normalized rg space ($r = \frac{R}{R+G+B}$, $g = \frac{G}{R+G+B}$) for the labeled foregrounds and backgrounds from the training data respectively. The color prior map for each pixel x with color x_c is generated by: $p_c(x) = \exp((H_s(x_c) - H_b(x_c))/\sigma_3)$, where $\sigma_3 = 0.02$ in our experiments.

These prior maps $p_l(x)$, $p_f(x)$ and $p_c(x)$ are normalized to $[0, 1]$ by using $p(x) = (p(x) - \min_x(p(x)))/(\max_x(p(x)) - \min_x(p(x)))$. Then they are simply averaged and spatially smoothed [3] to generate the final combined high-level prior map $P_H = [p^{(1)}, p^{(2)}...p^{(N)}]$, where $p^{(i)} \in [0, 1]$. Figure 4 provides some examples of high-level prior maps. For each superpixel, we use the mean of the prior values of its pixels, as its high-level prior value.

We introduce an 'assignment cost' $1 - P_H$ for each superpixel and incorporate it into the computation of c_{ij} as follows. Given a labeled set L (comprised of a region center node v_j and the background node v_g), we augment the graph G to include a set of labeled nodes, by attaching a labeled node v_q^i to each unlabeled node v_i ($i \in U$) as its prior. Note that only v_i is connected to v_q^i. Let G' denote the augmented graph and Q be the set of augmented nodes. Y_Q is a label matrix for Q of size $u \times 1$, where $Y_Q(i) = 1 - p^{(i)}$ for $i \in U$. We again use the harmonic solution on G' to compute the label values for U. More

[3] It is achieved by weighted averaging, where the weights are computed by pair-wise distances between superpixels using a Gaussian kernel.

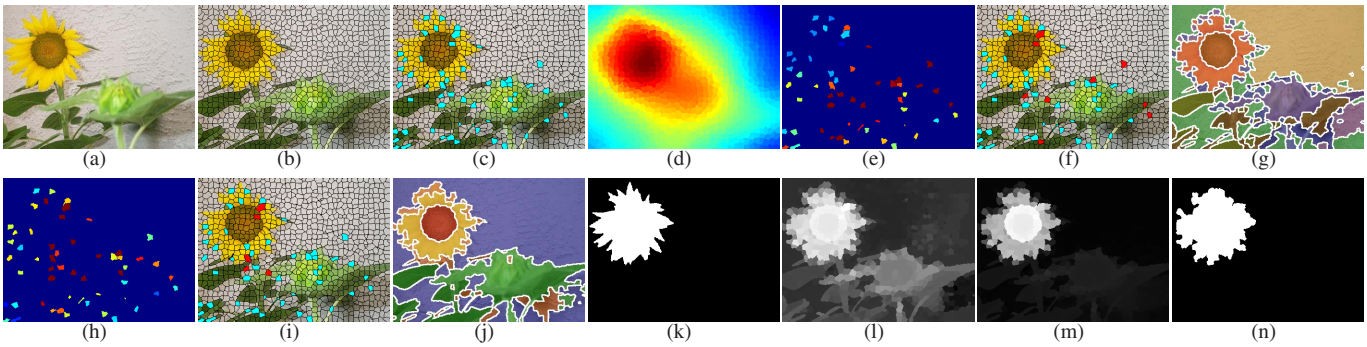


Figure 3. An example of detecting salient regions with different components. (a) Input image; (b) Superpixel segmentation; (c) Hypothesized set J marked as blue; (d) Combined high-level prior map; (e) Marginal gain of each point in J: $\mathcal{H}(A\cup\{a\})-\mathcal{H}(A)$ without prior for the 1-st iteration; (f) Final selected region centers (facility locations) without prior. $|A|=9$ locations marked as red are iteratively selected; (g) Potential salient region extraction. Nine regions generated; (h) Marginal gain of each point in J with prior for the 1-st iteration. Most data points having high gains are aggregated in the area with high prior values; (i) Final selected facility locations with prior. $|A|=5$ locations marked as red are selected; (j) Potential salient region extraction with prior. Five regions generated; (k) Ground truth salient region; (l) Saliency map without prior; (m) Saliency map with prior; (n) Salient region mask based on the saliency map in (m).

specifically, the weight matrix W' for G' (and similarly D', and the transition matrix $P'=(D')^{-1}W'$) is divided into 9 blocks:$W'=\begin{bmatrix} W_{LL} & W_{LQ} & W_{LU} \\ W_{QL} & W_{QQ} & W_{QU} \\ W_{UL} & W_{UQ} & W_{UU} \end{bmatrix}$. $h=\begin{pmatrix} h_L \\ h_Q \\ h_U \end{pmatrix}$, is the harmonic solution satisfying $\triangle h=0$ on G' subject to $h_L=Y_L$ and $h_Q=Y_Q$ is:

$$\begin{aligned} h_U &= (D'_{UU}-W_{UU})^{-1}(W_{UL}h_L+W_{UQ}h_Q) \\ &= (I_{UU}-(D'_{UU})^{-1}W_{UU})^{-1} \\ &\quad ((D'_{UU})^{-1}W_{UL}h_L+(D'_{UU})^{-1}W_{UQ}h_Q) \quad (4) \end{aligned}$$

Assume the transition probability P_{iq} from each v_i ($i\in U$) to its attached node v_q^i ($q\in Q$) be a constant θ, we have $(D'_{UU})^{-1}W_{UQ}=\theta\mathbf{I}$ ($\mathbf{I}$ is the identity matrix). Given $D'_{UU}=W_{UL}+W_{UQ}+W_{UU}$, we obtain $D'_{UU}=\frac{1}{1-\theta}(W_{UL}+W_{UU})$. (4) can be rewritten as:

$$h_U=(I_{UU}-(1-\theta)P_{UU})^{-1}((1-\theta)P_{UL}h_L+\theta h_Q) \quad (5)$$

where P_{UU} and P_{UL} are the transition probabilities on the 'original' graph G as in (2). We have $h_L=(1,0)^t$ for this problem. We can compute $c_{ij}=h_U\in R^{u\times 1}$ while $c_{jj}=1$ and $c_{gj}=0$. At one extreme, when $\theta=1.0$, h_U is purely based on high level prior information and $h_U=h_Q$. The region center with the largest marginal gain is the location with the lowest assignment cost. At another extreme, when $\theta=0$, h_U is purely data driven and can be computed using (2). Hence this computation of c_{ij} encourages the selected facility locations to be close to low cost areas (*i.e.*, high probability area indicated by high-level prior map). We use $\theta=0.05$ in all of our experiments.

Figures 3(e) and 3(h) show the marginal gain for each point in J in the first iteration of the facility location optimization without and with the high level prior. After high-level prior integration, the points with large marginal gains are more concentrated in the perceptually important areas (indicated by high-level prior map) such as the flower and the flower leaf. Compared to the selected region centers A without the prior in Figure 3(f), our approach with priors will select most of the potential region centers for A from the high prior areas as shown in Figure 3(i).

3.3.3 Potential Salient Region Extraction

Given a set of selected facility locations A, let the current maximal profit from v_i be $\rho_i^{cur}=\max_{j\in A}c_{ij}$, and the facility assignment for v_i be $x_i^{cur}=\arg\max_{j\in A}c_{ij}$. At each iteration during the optimization (discussed in Sec. 3.3.4), given the newly selected a^*, if $\rho_i^{cur}<c_{ia^*}$, we update ρ_i^{cur} and x_i^{cur} for v_i to be c_{ia^*} and a^* respectively. This corresponds to steps $10-14$ in Algorithm 1. Hence, we cluster the image elements that share the same facility location as the most profitable to obtain potential salient regions $\{r_i\}_{i=1\ldots|A|}$.

Figure 2 show two examples of facility location and facility assignment results (*i.e.*, clustering results) on two synthetic datasets. The results in Figure 2(f) using a harmonic function to compute c_{ij} are better than the results in Figure 2(g) that simply uses the edge weight w_{ij} as c_{ij}. The reason is that harmonic solution for c_{ij} enforces that nearby points have similar harmonic function values; this better models the geometry of the data induced by the graph structure (edges and weights W). For these two examples, we construct fully-connected graphs. We compute the marginal gain for every point a in V (*i.e.*, $J=V$) and add the point with the maximum gain to A at each iteration.

Figures 3(g) and 3(j) show the region extraction results for the two sets of selected region centers A shown in Figure 3(f) and Figure 3(i), respectively.

3.3.4 Optimization

Direct maximization of (3) is a NP-hard problem [22]. However, one simple solution can be obtained by a greedy

algorithm from [24, 22]. The algorithm starts from an empty set $A = \emptyset$, and iteratively adds to A an element $a \in J \backslash A$ that provides the largest marginal gain for $\mathcal{H}$ and updates the facility assignment of v_i whose current profit ρ_i^{cur} is small than the profit c_{ia^*} from the newly selected a^*. The iteration stops when the desired number of regions (open facilities) is reached or $\mathcal{H}$ decreases.

The constraint on the number of open facilities induces a simple uniform matriod $\mathcal{M} = (J, \mathcal{I})$. $\mathcal{I}$ is the collection of subsets $A \subseteq J$ which satisfies: the number of open facilities N_A is less than K. Maximization of a submodular function with a uniform matroid constraint yields a $(1 - 1/e)$-approximation [24]. Hence the greedy algorithm provides a performance-guarantee solution.

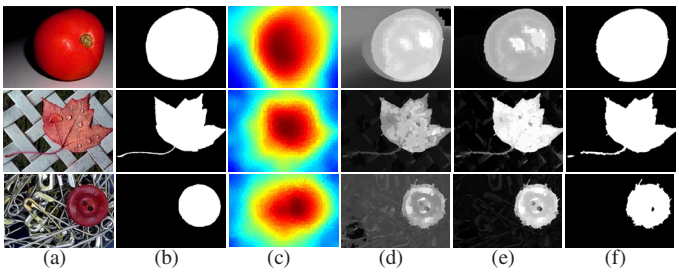
Instead of recomputing the gain for every location $a \in J \backslash A$ after adding a new element into A, which requires $|J| - |A|$ evaluations for the gain of $\mathcal{H}$, we use lazy evaluation from [18] to speed up the optimization process, by using the submodularity property of the objective function. In our experiment, the lazy greedy approach achieves up to $3 \sim 20$ times speedup while having the same accuracy as the naive greedy algorithm. Algorithm 1 presents the pseudocode of our algorithm.

3.4. Saliency Map Construction

After extracting $\{r_i\}_{i=1...|A|}$, we next compute the saliency of r_i in terms of its color and spatial information. A region which has a high color contrast with respect to other regions should have a high saliency [2, 4]. The color saliency of r_i is defined as: $f_c(r_i) = \sum_k \tau(r_k) D_c(r_i, r_k)$, where $\tau(r_k)$ is the number of superpixels in r_k; this gives more weight to contrast with larger regions. D_c is the average of all feature distances between pairs of superpixels from r_i and r_k.

A region which has a wider spatial distribution is typically less salient than regions which have small spatial spread [20, 8]. The spatial saliency of r_i is computed as $f_s(r_i) = 1 - \frac{V(r_i)}{\max_i V(r_i)}$. $V(r_i) = \sum_k D^{(i)}(\mu_k)$, where $D^{(i)}(\mu_k)$ is the average of all the distances of superpixels from r_i to the spatial mean μ_k of region r_k. This favors regions with small spatial variance and eliminates the background color of large variance. After f_c and f_s are maximum normalized to $[0, 1]$, the saliency of r_i is computed as: $f(r_i) = f_c(r_i) f_s(r_i)$. We generate the final saliency map $\mathcal{S}$ by weighted averaging over superpixels, where the weights are computed by pair-wise feature distances between superpixels using a Gaussian kernel to enforce that similar superpixels should have similar visual saliency.

Figures 3(l) and 3(m) present the saliency maps using our approach without and with high-level priors, respectively. Compared to the ground truth region in Figure 3(k), the saliency maps with priors are better than the result without priors. More results are shown in Figure 5. As shown in Figure 5(e), the saliency maps using high-level priors are better than the results without priors in Figure 5(d).



(a) (b) (c) (d) (e) (f)

Figure 5. Examples of extracting salient regions. (a) Input images; (b) Ground truth salient regions; (c) High-level prior map; (d) Saliency map without high level prior; (e) Saliency map with high level prior; (f) Salient Region extraction based on (e) by simple thresholding.

Algorithm 1 Submodular Salient Region Detection

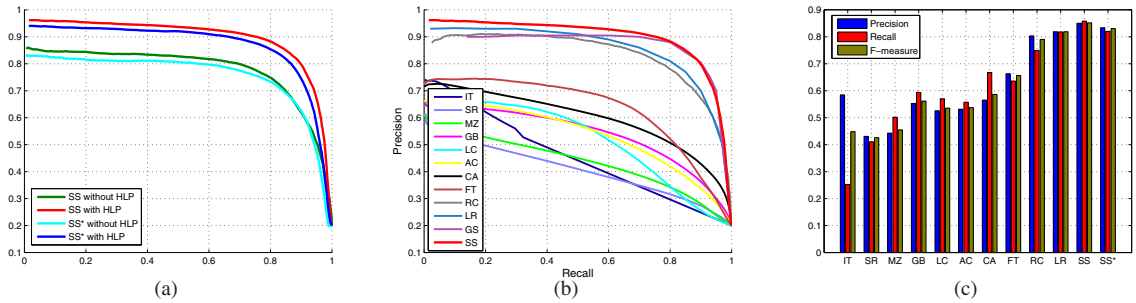
Input: I, $G = (V, E)$, c_{ij}, K and λ.
Output: A, x_i^{cur}, $\mathcal{S}$
Initialization: $A \leftarrow \emptyset$, $\rho_i^{cur} \leftarrow 0$, $x_i^{cur} \leftarrow 0$
loop
  $a^* = \underset{\{A \cup a\} \in \mathcal{I}}{\operatorname{argmax}} \mathcal{H}(A \cup \{a\}) - \mathcal{H}(A)$
  if $\mathcal{H}(A \cup \{a^*\}) \leq \mathcal{H}(A)$ or $N_A > K$ **then**
    break;
  end if
  $A \leftarrow A \cup \{a^*\}$, $\rho_{a^*}^{cur} \leftarrow 1$
  for $\forall i \in V \backslash A$ **do**
    if $\rho_i^{cur} < c_{ia^*}$ **then**
      $\rho_i^{cur} = c_{ia^*}$, $x_i^{cur} = a^*$
    end if
  end for
end loop
Construct the saliency map $\mathcal{S}$ for I;

4. Experiments

We evaluate our approach on two popular saliency databases: MSRA-1000 database [2] and Berkeley-300 database [23]. The MSRA-1000 database is a 1000-image subset of the MSRA database [20]. These 1000 images are excluded when learning the color prior, which is trained on other images from the MSRA dataset [20] and evaluated on the both MSRA-1000 database and Berkeley-300 database. We refer to our approach using TurboPixels [19] and SLIC [3] for superpixel extraction as 'SS' and 'SS*', respectively, in the following.

For the first evaluation, a fixed threshold within $[0, 255]$ is used to construct a binary foreground mask from the saliency map. Then, the binary mask is compared with the ground truth mask to obtain a precision-recall (PR) pair. We vary the threshold over its entire range to obtain the PR curve for one image. The average precision-recall curve is obtained by averaging the results from all testing images.

For the second evaluation, we follow [2, 4, 26] to segment a saliency map by adaptive thresholding. The saliency mean is first computed over the entire image. If the saliency of a superpixel is larger than two times the saliency mean, the superpixel is considered as foreground. Then the preci-



(a) (b) (c)

Figure 6. Experimental results on the MSRA-1000 database. 'SS*' means that superpixels are extracted using SLIC [3], SS with TurboPixels [19]. (a) Average precision-recall curves using our approach with different superpixels and with/without priors; (b) Average precision-recall curves using different approaches. The curves for LR [26] and GS [27] are copied from the papers; (c) Average precision, recall and F-measure using different approaches with adaptive thresholding. The bars for LR [26] are copied from the paper.

sion ($\tilde{P}$), recall ($\tilde{R}$) and F-Measure ($\tilde{F}$) values can be computed over the ground truth maps, where the F-Measure is defined as $F = ((1+\alpha)\tilde{P}\tilde{R})/(\alpha\tilde{P}+\tilde{R})$. α is set to be 0.3 as in [2, 4, 26].

4.1. MSRA-1000 Database

The MSRA-1000 database provides the human labeled object segmentation masks. We first compare the performances of our approach with priors and without priors. The average precision-recall curves are shown in Figure 6(a). By combining the high-level priors, the saliency detection performance is improved. λ is set to 5 in our experiments.

In Figure 6(b), we compare our precision-recall curve with IT [12], MZ [21], GB [10], CA [6], RC [4], SR [11], AC [1], LC [31] and FT [2] and two recently proposed approaches LR [26] and GS [27]. Our result is comparable to GS and outperforms the other approaches. The average precision, recall and F-Measure using different approaches with adaptive thresholding are shown in Figure 6(c). Among all approaches, our approach (SS) achieves highest precision, recall and F-Measure values.

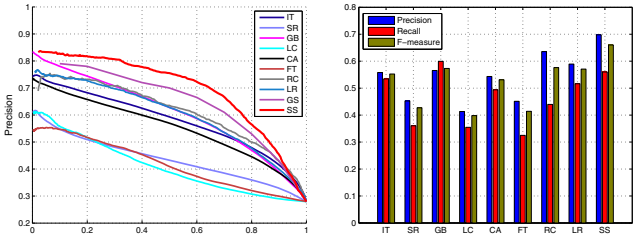
Table 1 compares the average running time of different approaches. Here we only list the approaches which use Matlab implementation for fair comparison. Our approach (using either SS or SS*) is faster than CA [6] and LR [26]. Because our approach needs superpixel segmentation, it is slower than IT [12] and GB [10] but produces superior quality saliency maps as shown in Figure 7. For SS, superpixel extraction by [19] takes about 5 seconds per frame. We use [3] to extract SLIC superpixels based on their more efficient algorithm and reevaluate the precision-recall performances. As shown in Figure 6(a) and 6(c), SS* is comparable to SS and outperforms most other approaches. Figure 7 shows some examples of saliency map construction using our approach and IT, FT, GB, LC, CA, RC and LR.

4.2. Berkeley-300 database

The Berkeley-300 database is a more challenging database introduced in [23]. Images typically contain multiple foreground objects of different sizes and positions. The foreground masks are provided by [27] as the ground truth

Table 1. Computation time per image for saliency detection, measured on an Intel 2.40GHZ CPU machine with 4GB RAM. All approaches use Matlab implementations. For SS, it takes about 5 seconds per image for superpixel extraction by [19]. 'SS*', that superpixels extracted by SLIC [3] is more efficient.

Method	IT [12]	GB [10]	CA [6]	LR [26]	SS	SS*
Time (s)	0.45	1.61	58.8	41.1	6.6	2.1



(a) (b)

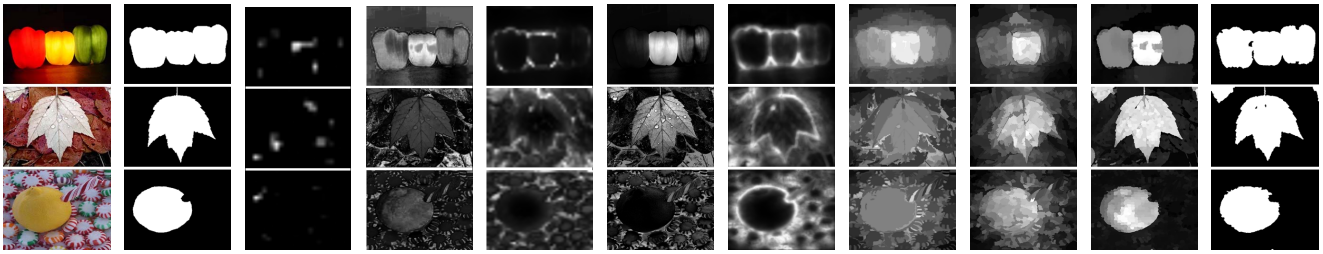
Figure 8. Experimental results on the Berkeley-300 database. (a) Average precision-recall curves using different approaches. The curves for GS [27] are copied from the papers; (b) Average precision, recall and F-measure with adaptive thresholding.

salient regions. The color prior trained from the MSRA dataset is used. We evaluate the precision-recall curves using SS on this database.

We compare the precision-recall curve of our approach with IT [12], FT [2], GB [10], CA [6], RC [4], SR [11], LR [26], LC [31] and GS [27]. We rerun their implementations for evaluation except GS. We copied the precision-recall curve of GS from [27]. As shown in Figure 8(a), our approach achieves the overall best performance. We also evaluate average precision, recall and F-Measure using adaptive thresholding, and compare with other approaches. The result is shown in Figure 8(b). Our approach achieves better performance on the average precision and F-measure, and is comparable to GB on the recall measure. Figure 9 shows some saliency maps using different approaches.

5. Conclusion

We presented a greedy-based salient region detection approach by maximizing a submodular function, which can be viewed as the *facility location* problem. By combing high level prior information with low level feature information



(a) Inputs (b) GT (c) IT [12] (d) FT [2] (e) GB [10] (f) LC [31] (g) CA [6] (h) RC [4] (i) LR [26] (j) SS (ours) (k) Detection

Figure 7. Examples of saliency map construction using different approaches on the MSRA-1000 database. The saliency maps in (j) are used to segment the salient regions by simple thresholding. The results are shown in (k).

(a) Inputs (b) GT (c) IT [12] (d) FT [2] (e) GB [10] (f) LC [31] (g) CA [6] (h) RC [4] (i) LR [26] (j) SS (ours) (k) Detection

Figure 9. Examples of saliency map construction using different approaches on the Berkeley-300 database. The saliency maps in (j) are used to segment the salient regions by simple thresholding. The results are shown in (k).

into the objective function, the saliency of detected regions is improved and more consistent with human visual perception. The objective function is optimized by a highly efficient greedy algorithm. The similarities between a region center and its region elements can be modeled as a labeling problem on the constructed graph and solved by constructing the harmonic function of the graph. Experimental results indicate that the algorithm outperforms recently proposed saliency detection techniques including FT [2], CA [6], RC [4] and LR [26] and is comparable to GS [27].

Acknowledgement

This work was supported by the Army Research Office MURI Grant W911NF-09-1-0383.

References

[1] R. Achanta, F. Estrada, P. Wils, and S. Susstrunk. Salient region detection and segmentation, 2008. *ICVS*.

[2] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection, 2009. *CVPR*.

[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 2012.

[4] M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu. Global contrast based salient region detection, 2011. *CVPR*.

[5] D. Gao and N. Vasconcelos. Discriminant saliency for visual recognition from cluttered scenes, 2004. *NIPS*.

[6] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection, 2010. *CVPR*.

[7] V. Gopalakrishnan, Y. Hu, and D. Rajan. Random walks on graphs to model saliency in images, 2009. *CVPR*.

[8] V. Gopalakrishnan, Y. Hu, and D. Rajan. Salient region detection by modeling distributions of color and orientation. *IEEE Trans. on Multimedia*, 2009.

[9] L. Grady. Random walks for image segmentation. *TPAMI*, 2006.

[10] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency, 2007. *NIPS*.

[11] X. Hou and L. Zhang. Saliency detection: A spectral residual approach, 2007. *CVPR*.

[12] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *TPAMI*, pages 1254–1259, 1998.

[13] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look, 2009. *ICCV*.

[14] C. Kanan, M. Tong, L. Zhang, and G. Cottrell. Sun: Top-down saliency using natural statistics. *TPAMI*, 2009.

[15] G. Kim, E. Xing, F. Li, and T. Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion, 2011. *ICCV*.

[16] B. Ko and J. Nam. Object-of-interest image segentation based on human attention and semantic region clustering. *Journal of Opt. Soc. Am.*, 2006.

[17] N. Lazic, I. Givoni, and B. Frey. Floss: Facility location for subspace segmentation, 2009. *ICCV*.

[18] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks, 2007. *KDD*.

[19] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *TPAMI*, 2009.

[20] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum. Learning to detect a salient object, 2007. *CVPR*.

[21] Y. Ma and H. Zhang. Contrast-based image attention analysis by using fuzzy growing, 2003. *ACM Multimedia*.

[22] P. Mirchandani and R. Francis. The uncapacitated facility location problem. *Discrete Location Theory*, 1990.

[23] V. Movahedi and J. Elder. Design and perceptual validation of performance measures for salient object segmentation, 2010. *IEEE Workshop on Perceptual Organization in Computer Vision*.

[24] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions-i. *Mathematical Programming*, pages 265–294, 1978.

[25] C. Rother, V. Kolmogorov, and A. Blake. Interactive foreground extraction using iterated graph cuts, 2004. *SIGGRAPH*.

[26] X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery, 2012. *CVPR*.

[27] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors, 2012. *ECCV*.

[28] J. Yan, M. Zhu, H. Liu, and Y. Liu. Visual saliency detection via sparsity pursuit. *IEEE Signal Processing Letters*, pages 739–742, 2010.

[29] J. Yang and M. Yang. Top-down visual saliency via joint crf and dictionary learning, 2012. *CVPR*.

[30] L. Zelnik-Manor and P.Perona. Self-tuning spectral clustering, 2004. *NIPS*.

[31] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues, 2006. *ACM Multimedia*.

[32] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions, 2003. *ICML*.