2013 IEEE Conference on Computer Vision and Pattern Recognition

Maximum Cohesive Grid of Superpixels for Fast Object Localization

Liang Li[1,2], Wei Feng[1,2,*], Liang Wan[3], Jiawan Zhang[3]
[1] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin, China
[2] School of Computer Science and Technology, Tianjin University, Tianjin, China
[3] School of Computer Software, Tianjin University, Tianjin, China
{liangli,wfeng,lwan,jwzhang}@tju.edu.cn

Abstract

This paper addresses a challenging problem of regularizing arbitrary superpixels into an optimal grid structure, which may significantly extend current low-level vision algorithms by allowing them to use superpixels (SPs) conveniently as using pixels. For this purpose, we aim at constructing maximum cohesive SP-grid, which is composed of real nodes, i.e. *SPs, and dummy nodes that are meaningless in the image with only position-taking function in the grid. For a given formation of image SPs and proper number of dummy nodes, we first dynamically align them into a grid based on the centroid localities of SPs. We then define the SP-grid coherence as the sum of edge weights, with SP locality and appearance encoded, along all direct paths connecting any two nearest neighboring real nodes in the grid. We finally maximize the SP-grid coherence via cascade dynamic programming. Our approach can take the regional objectness as an optional constraint to produce more semantically reliable SP-grids. Experiments on object localization show that our approach outperforms state-of-the-art methods in terms of both detection accuracy and speed. We also find that with the same searching strategy and features, object localization at SP-level is about* 100-500 *times faster than pixel-level, with usually better detection accuracy.*

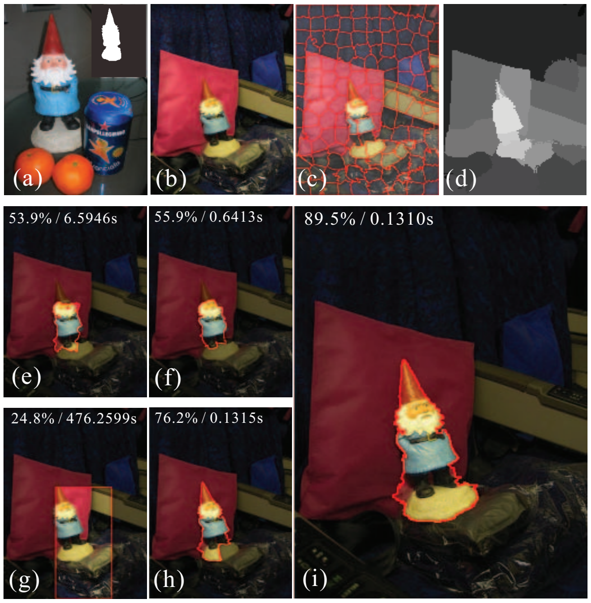

Figure 1: Grid regularized SPs for object localization. (a) Query image (with query mask in the corner). (b) Target image. (c) SLIC SPs (with irregular structure) [1]. (d) SP-level objectness. (e)-(g) Object detection results of TurboPixel [14], SuperLattice [17] and pixel-level RC with finer searching step [23]. Due to the apparent scale variation in query and target images, pixel-level RC needs finer step to search the foreground box, which may become very slow. (h) and (i) are the results of our approach by regularizing SLIC SPs (c) without/with the guidance of SP objectness (d), respectively. In (e)-(i), the left-up corner shows the accuracy and running time.

1. Introduction

To pursue efficiency, accuracy and scalability in large-scale image analysis, many recent algorithms in computer vision are now based on superpixels. From the angle of MRF [8], superpixels (SPs), generated by grouping similar pixels into perceptually meaningful atomic regions [18], can dramatically reduce the number of variables to be optimized, thus leading to significant speed-up and allowing the analysis of long-range correlations. At the same time, since SPs capture most meaningful image structures, they are usually well-aligned to image edges. As a result, some algorithms, *e.g.* [9, 24, 25], if running at SP-level, can achieve higher accuracy than running at pixel-level.

At the very beginning, superpixels were simply treated as fast over-segmentations to the image [15]. As shown in Fig. 1(c), image over-segmentations usually tend to generate SPs with variant sizes, shapes and irregular spatial dis-

[*] is the corresponding author. Tel: (+86)-22-27406538.

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.408

tributions. As the increasing usage of SPs in image parsing [22], segmentation [18, 24], co-segmentation [10, 20], and object localization [13], people start to realize the importance of structural regularities in SPs [1, 16, 21, 25].

Recently, some regular or near-regular SP algorithms have been proposed, *e.g*. SuperLattice [17], LatticeCut [16], TurboPixel [14] and min-energy based SPs [25]. They generally follow a similar strategy to generate SPs, *i.e*. seeking proper tradeoff between the structural regularity and the boundary accuracy of superpixels. As validated by our experiments, for some kinds of images, current regular SP methods can produce feasible results. However, due to the complexity of natural images, *e.g*. the existence of both homogeneous and multi-scale texture regions, the influence of changing lighting, shadows and occlusions, no particular method can guarantee satisfactory segmentations for all types of images. Thus, compared to particular regular SP algorithms, it is more desirable to find a way rectifying arbitrary segmentations into a regular structure. Besides, another notable weakness of current regular SP methods is that their performance may highly depend on the pre-computed edge map [16, 21]. This paper, to the best of our knowledge, for the first time proposes a generic approach to optimally regularizing arbitrary SPs into a regular grid. By this, we can both fully take advantage of the strength of various image segmentation/SP methods [1, 6, 5, 8], and enjoy the desirable properties of grid at the same time.

To this end, we define *cohesive SP-grid*, which is composed of (1) *real nodes*, *i.e*. real SPs generated by any appropriate superpixel or segmentation algorithms, and (2) *dummy nodes* that are meaningless in the image with only position-taking function in the grid. We aim at constructing maximum cohesive SP-grid that regularizes all pairwise SP connections into a lattice, while preserving the most important image structures. We propose a two-step approach for this purpose. First, we unevenly assign all real nodes into a grid by minimizing the overall *locality discrepancy cost*. The initial cohesive SP-grid is obtained by appending proper number of dummy nodes at the end of each grid column. We then iteratively refine the cohesive SP-grid by optimizing each grid column within its contemporary context configurations. We call this process cascade dynamic programming (DP) that converges very fast in practice. As an optional compensation, the regional objectness score [2] can also be used as an extra constraint to refine the SP coherence measurement, thus leading to a more semantically feasible SP-grid. Experiments on object localization show that our approach outperforms state-of-the-art ones in terms of both detection accuracy and speed. With the same strategy and features [23], object localization via our SP-grid is 100-500 times faster (including grid regularization and matching time) than pixel-level matching, and usually produces better detection accuracy.

2. Related Work

Superpixels. The concept of superpixels stems from the homogeneous subregions generated by a fast over-segmentation to the image, *e.g*. MeanShift [5] and EGS [6]. This kind of SPs usually form an irregular graph, with SP boundaries well-aligned to image edges. They were widely used in image segmentation [18, 15]. Recently, people start to realize the advantages of regular structured SPs. Typical methods include SLIC [1], TurboPixel [14], SuperLattice [17] and LatticeCut [16]. To maintain the structural regularity, both SLIC and TurboPixel start from a set of uniformly placed seeds, and respectively use k-means clustering and geometric-flow to generate final SPs. In contrast to the near-grid property of SLIC and TurboPixel, SuperLattice and LatticeCut are able to produce exact grid structured SPs. For instance, based on a pre-computed reliable edge map, SuperLattice adopts a greedy strategy to generate the optimal paths of SP-grid by following the input edge map and satisfying the grid structural constraints [17].

Fast object localization. Recent useful object localization routines includes Region Covariance (RC) [23] and Efficient Subwindow Search (ESS) [12]. In [23], the R-C descriptor encoding color, gradient and locality features has been proposed for robust object detection in different images. The searching efficiency using RC descriptor is guaranteed by a $O(1)$-time integral-image based rectangle mean/covariance computation. In contrast, ESS uses a branch-and-bound searching strategy, but also relies on the integral-image computation as a core step. The ESS strategy has been further refined by a linear-time Kadane's algorithm by [4] and been extended to detect objects within large-volume image database in [11]. Note that, integral-image acceleration cannot be directly applied to irregular graphs with arbitrary structures.

Dynamic programming. Our approach utilizes DP to obtain the optimal locality-based SP-grid initialization and to maximize the overall SP-grid coherence. As an effective method for seeking globally optimal solution to discrete optimization problems, DP has a long history in computer vision and is still widely used in many recent algorithms [3, 7]. However, there is no general way to apply DP to any kind of irregular graphs of SPs.

With the proposed approach, most successful algorithms for both object localization and DP-based applications can be directly applied to SP-level, via any suitable type of SPs.

3. Overview

To regularize arbitrary SPs with any kind of irregular structure, we consider optimally allocating SPs within a virtual grid. By constructing such grid, only the most important pairwise SP connections can be explicitly preserved. In the regularization process, we take all possible SP pairs

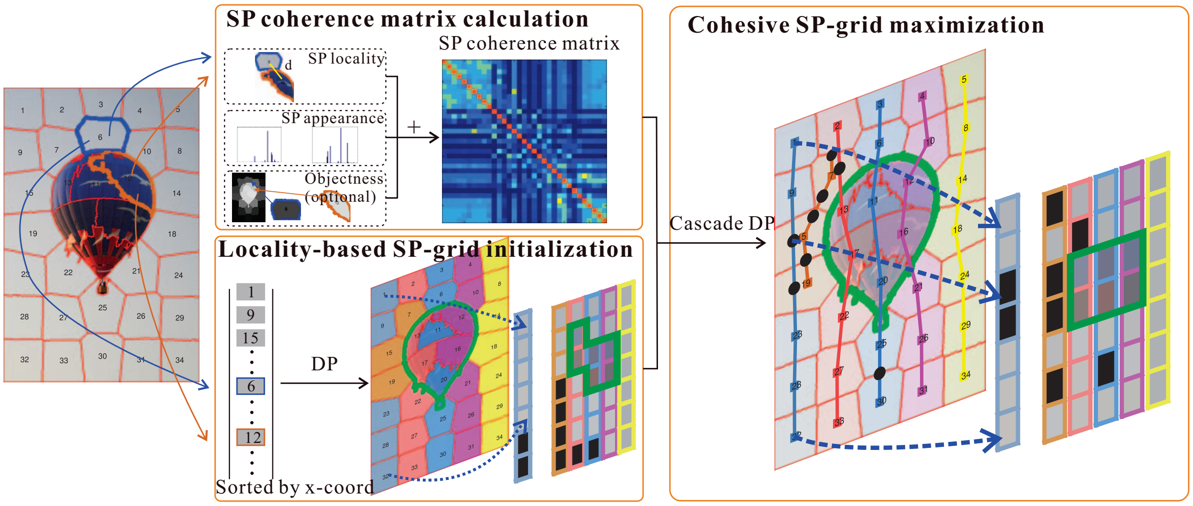

Figure 2: Algorithm flow of maximum cohesive grid regularization of arbitrary superpixels. See text for more details.

into consideration to capture both local and global (*i.e.* long-range) image structures. Since the SP pair coherence defined as their centroid locality and regional appearance closeness in Eq. (10) captures image SP-level structures, an optimal SP-grid should preserve maximum overall coherence. Besides, another issue is that all SPs may not necessarily occupy all positions in the virtual grid. Thus, we need also to incorporate position-taking dummy nodes.

Definition 1 (Cohesive SP-Grid) *Composed of* real nodes *(*i.e. *the input SPs) and position-taking* dummy nodes*, a* cohesive SP-grid *needs to satisfy three conditions: (1) The edge weight of any two neighboring real nodes equals to their coherence; (2) The edge weight of any two neighboring dummy nodes is* 0*; and (3) For any two real nodes p and q, if they lie in the same row/column and there are no in-between real nodes in that row/column, the weight of the* direct path *from p to q equals to their coherence* $\mathrm{Coh}(p,q)$*.* [1]

Note that, condition (2) and (3) in the Definition 1 ensure the only positive-taking function of dummy nodes. Hence, for a given set of real nodes $\mathcal{P}$ and dummy nodes $\mathcal{D}$, our objective of generic SP grid regularization can be formally expressed as constructing an optimal cohesive SP-grid $\mathcal{G} = \langle \mathcal{P} \cup \mathcal{D}, \mathcal{E} \rangle$ with maximum overall coherence:

$$\mathrm{Coh}(\mathcal{G}) = \sum_{e \in \mathcal{E}} \mathrm{Coh}(e). \tag{1}$$

4. Maximum Cohesive Grid of Superpixels

For a given set of superpixels $\mathcal{P}$, seeking global maximum cohesive SP-grid is generally intractable, *e.g.* let $r \times c$ be the size of target DP-grid, the solution space is of size $(|\mathcal{P}|+1)^{rc}$.[2] Moreover, recalling condition (3) in Definition 1, the *state* of a grid position, *i.e.* either some real node or the dummy node, highly correlates to its nearest neighboring positions, if they together form a direction path. This may induce many high-order terms and make the energy function very hard to be solved [19].

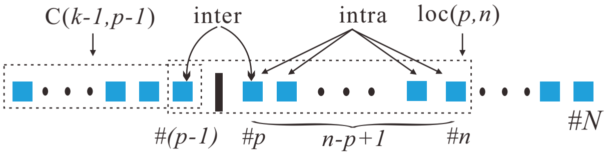

Figure 3: Locality-based dynamic SP-grid initialization. The black line indicates the position of a column-cut. See text for details.

As shown in Fig. 2, this paper proposes a two-step near-otimum approach to (1) cohesive SP-grid initialization, and (2) cohesive SP-grid maximization by cascade DP.

4.1. Locality-based SP-grid initialization

For a good cohesive SP-grid, the relative localities of SPs in the image should be respected. So, our first step is to optimally assign all SPs of $\mathcal{P}$ into c columns according to their normalized localities. Note that, in this paper, both of our initialization and maximization are conducted in columns only, which is empirically proven to be comparable with optimizing in rows or in both rows and columns for object localization by our experiments.

To do so, we first sort all SPs into a 1D sequence $\mathcal{P}_{\mathrm{xsort}}$ with increasing x-coordinates of their centroids. The c columns can be determined by $c-1$ column-cuts, with all

[1] Direct path from p to q is the sequence of edges connecting them and passing only dummy nodes in the same row/column of p and q. The weight of a direct path is the sum of all edge weights in the path.

[2] This assumes an SP can be assigned to more than one grid positions.

Algorithm 1 Locality-based SP-grid initialization

```
Require: The set of input SPs $\mathcal{P}$
Ensure: Optimal $\hat{\mathcal{B}} = \{\hat{b}_i\}_{i=1}^{c-1}$
/* Stage 1: compute $\mathrm{C}(k,n)$ and $\mathrm{B_L}(k,n)$ */
for $n = 1$ to $|\mathcal{P}|$ do
    for $k = 2$ to $c$ do
        Compute $\mathrm{C}(k,n)$ and $\mathrm{B_L}(k,n)$ using Eqs. (4)–(6);
    end for
end for
/* Stage 2: back retrieving $\hat{\mathcal{B}}$ */
Set $n := |\mathcal{P}|$ and $\hat{b}_1 = 1$;
for $k = c$ to 2 do
    Set $\hat{b}_{k-1} := \mathrm{B_L}(k,n)$;
    Set $n := \hat{b}_{k-1} - 1$;
end for
```

SPs between two consecutive cuts forming a particular column. Note, the SPs within a column are re-arranged in an ascending order of the y-coordinates of their centroids. Hence, a set of column-cuts $\mathcal{B} = \{b_i\}_{i=1}^{c-1}$ defines a particular way of partitioning all SPs of $\mathcal{P}$ into c columns, where $b_i = p$ means the #i column-cut lies right between the #$(p-1)$ and #p SPs of $\mathcal{P}_{\mathrm{xsort}}$. Then, we measure its goodness by the following locality discrepancy score, and seek an optimal $\hat{\mathcal{B}} = \arg\min_{\mathcal{B}} \mathrm{Loc}(\mathcal{B})$:

$$\mathrm{Loc}(\mathcal{B}) = \sum_{i=1}^{c} \mathrm{loc}(b_i, b_{i+1}), \tag{2}$$

where $\mathrm{Loc}(\mathcal{B})$ is the sum of discrepancy values of all c columns. For the ease of expression, we set $b_c = |\mathcal{P}| + 1$.

$$\mathrm{loc}(b_i, b_{i+1}) = \frac{\omega_{\mathrm{sep}} \cdot \mathrm{intra}_i}{\max(\mathrm{inter}_i, \epsilon)} + \omega_{\mathrm{len}} \cdot (L_i - \bar{L}). \tag{3}$$

As shown in Fig. 3, intra_i is the intra-column discrepancy measured by the average centroid L_2-distance of all consecutive SP pairs in the #i column; inter_i is the inter-column discrepancy measured by the x-coordinates difference between two neighboring SPs of the #i column-cut; constant $\epsilon > 0$ avoids dividing by zero; $L_i = b_{i+1} - b_i + 1$ is the length of the #i column and $\bar{L} = \frac{|\mathcal{P}|}{c}$ is the average column length. Note that, $\mathrm{Loc}(\mathcal{B})$ encourages nearly equal-sized columns with minimum intra-discrepancies and maximum inter-differences. ω_{sep} and ω_{len} are the weights of column separability and size regularity in Eq. (2), respectively.

We can efficiently obtain the global optimum $\hat{\mathcal{B}}$ using the following DP formulation:

$$\mathrm{loc}(p,k,n) = \mathrm{C}(k-1,p-1) + \mathrm{loc}(p,n), \tag{4}$$

$$\mathrm{C}(k,n) = \min_{k \le p \le n} \mathrm{loc}(p,k,n), \tag{5}$$

$$\mathrm{B_L}(k,n) = \arg\min_{k \le p \le n} \mathrm{loc}(p,k,n). \tag{6}$$

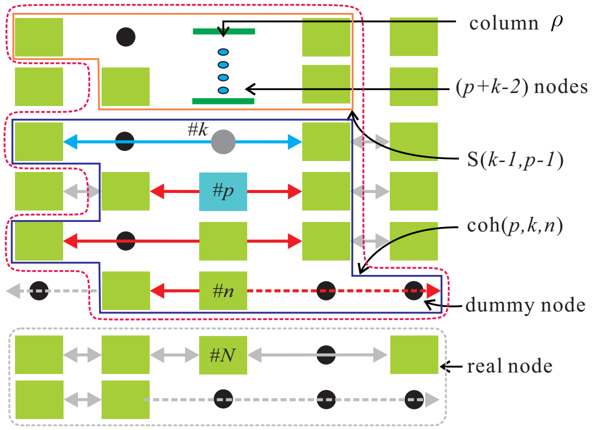

Figure 4: Maximizing cohesive SP-grid using cascade DP. The blue solid-lined region is the correlated subgraph used to calculate $\mathrm{coh}(p,k,n)$; the orange solid-lined region is the correlated subgraph of $\mathrm{S}(k-1,p-1)$; while the red dash-lined region is the correlated subgraph corresponding to $\mathrm{S}(k,n)$. The red arrow-lines starting from the #p position in column ρ denote the direct path connecting two nearest neighboring left and right real nodes of #p. See text for more details.

Note that, $\mathrm{loc}(p,k,n)$ is the intermediate discrepancy value of partitioning the first n SPs of $\mathcal{P}_{\mathrm{xsort}}$ into k columns with the last column-cut lying right between the #$(p-1)$ and #p SPs; $\mathrm{C}(k,n)$ is the minimum discrepancy value of partitioning the first n SPs of $\mathcal{P}_{\mathrm{xsort}}$ into k columns;[3] $\mathrm{B_L}(k,n)$ is the back-retrieval table that records the #$(k-1)$ position of optimal column-cuts partitioning the first n SPs into k columns. Then, we can efficiently obtain $\hat{\mathcal{B}}$ with minimum discrepancy using Algorithm 1 under the following boundary conditions: $2 \le k \le c$, $1 \le n \le |\mathcal{P}|$, $\mathrm{C}(1,n) = \mathrm{loc}(1,n)$ and $\mathrm{B_L}(1,n) = 1$. Based on $\hat{\mathcal{B}}$, we construct the initial cohesive DP-grid $\mathcal{G}^{(0)}$ by simply padding proper number of dummy nodes at the end of each column (see Fig. 2).

4.2. Dynamic maximization of cohesive SP-grid

Starting from $\mathcal{G}^{(0)}$, our near-optimal maximum cohesive SP-grid $\mathcal{G}^*$ is progressively refined by optimizing its every column under current configurations of other columns. Since we in turn repeatedly update every column of $\mathcal{G}^*$ via DP to maximize the overall SP-grid coherence, we name the method cascade DP.

For a particular column ρ of $\mathcal{G}^*$, we use the following DP process to update it under the context of current configurations of other columns in $\mathcal{G}^*$, while satisfying the three conditions of Definition 1:

$$\mathrm{S}(k,n) = \max_{k \le p \le n+1} [\mathrm{S}(k-1,p-1) + \mathrm{coh}(p,k,n)], \tag{7}$$

[3] Clearly, $\mathrm{Loc}(\hat{\mathcal{B}}) = \mathrm{C}(c, |\mathcal{P}|)$.

$$\mathrm{B_S}(k,n) = \arg\max_{k \le p \le n+1} [\mathrm{S}(k{-}1, p{-}1) + \mathrm{coh}(p,k,n)]. \quad (8)$$

Note that, $\mathrm{S}(k,n)$ is the maximum overall coherence of the *correlated subgraph*, asserting the maximum increments caused by allocating k dummy nodes in the first n real nodes of the current column ρ. The correlated subgraph corresponding to such change is composed of all the nodes in the first $n+k$ rows of $\mathcal{G}^*$ whose states (*i.e.* either some particular real node or the dummy node) jointly contribute to increasing $\mathrm{Coh}(\mathcal{G}^*)$. As shown in Fig. 4, $\mathrm{S}(k,n)$ is actually the changed component of $\mathrm{Coh}(\mathcal{G}^*)$.

Clearly, we need only maximizing $\mathrm{S}(k,n)$, which can be dynamically expressed as the maximum sum of $\mathrm{S}(k-1, p-1)$ and $\mathrm{coh}(p,k,n)$, where $\mathrm{coh}(p,k,n)$ denotes the extra coherence increments by assigning the #k dummy node right in front of the #p real node of ρ. If $p = n+1$, this dummy node will be assigned right after the #n real node. $\mathrm{B_S}(k,n)$ is the back-retrieval table recording the best position of #k dummy node that forms the best configuration of adding k dummy nodes in the first n real nodes of ρ. Note, to meet condition (3) of Definition 1, calculating $\mathrm{coh}(p,k,n)$ needs to find the nearest horizontal real node neighbors for a particular position p (see Fig. 4).[4] We repeat the above process column by column till convergence.

For a particular column ρ, the whole SP-grid can be divided into two correlated subgraphs, as shown in Fig. 4 surrounded by orange and blue solid-lines. The overall coherence of these two correlated subgraphs is $\mathrm{S}(k-1, p-1)$ and $\mathrm{coh}(p,k,n)$, respectively. It is clear that maximizing the coherence of the first $(n+k)$ rows of $\mathcal{G}^*$ is equivalent to maximizing the sum of coherence of the two correlated subgraphs, *i.e.* $\mathrm{S}(k-1, p-1) + \mathrm{coh}(p,k,n)$, since other edge weights are irrelevant to the state of column ρ. In practice, we can easily calculate $\mathrm{coh}(p,k,n)$ as:

$$\mathrm{coh}(p,k,n) = \sum_{i=p}^{n} \mathrm{Coh}(i, i_r) + \mathrm{Coh}(i, i_l) + \mathrm{Coh}(k_l, k_r), \quad (9)$$

where i_l and i_r is the left-first real node and right-first real node for #i real node of column ρ, k_l and k_r is the left-first real node and right-first real node for #k dummy node of column ρ. In Fig. 4, $\mathrm{Coh}(i, i_r)$ and $\mathrm{Coh}(i, i_l)$ are shown as red arrow-lines (solid and dash ones), $\mathrm{Coh}(k_l, k_r)$ is shown as the blue arrow-line, respectively.

4.3. Superpixel coherence metric

We calculate the coherence of two SPs according to both their localities and appearances. For the #p SP of $\mathcal{P}$, we represent its locality as the normalized coordinates $o(p)$ of its centroid. To derive the appearance model of SPs, we uniformly quantize each channel of the RGB color space into 16 levels and then calculate the appearance histogram of each SP in the space of 16^3 bins. Then, we define the coherence of any two SPs p and q as

$$\mathrm{Coh}(p,q) = \exp\Big(-\frac{\|o_p - o_q\|_2}{\omega_{\mathrm{pos}}} + \frac{\mathrm{Ba}(H_p, H_q)}{\omega_{\mathrm{app}}}\Big), \quad (10)$$

where o_p and o_q are the normalized centroids of superpixel p and q, while H_p and H_q are the quantized color histograms of p and q, respectively. The Bhattacharyya coefficient $\mathrm{Ba}(H_p, H_q) = \sum_{b=1}^{16^3} \sqrt{H_p(b) \cdot H_q(b)}$ is used to measure the similarity between two histograms H_p and H_q.

4.4. Other issues

Objectness guidance. As shown in Fig. 1, the accuracy of our cohesive SP-grid for object detection may be further refined by incorporating the SP-level objectness that is defined as the mean objectness of all inclusive pixels. We compute the objectness of each pixel by averaging the objectness scores of a number of randomly-sampled subwindows in the image using the method of [2]. We impose the guidance of thresholded SP objectness in the following way: for any two neighboring SPs p and q in the image, if they both survive the objectness thresholding, we amplify their original coherence $\mathrm{Coh}(p,q)$ by a constant factor $F > 1$.

Convergence and complexity. Since our cohesive SP-grid initialization and maximization strictly satisfy Definition 1, the resultant $\mathcal{G}^*$ is certainly a cohesive SP-grid. Besides, the cascade DP guarantees strictly increasing coherence of $\mathcal{G}^*$ in each iteration. The complexity of SP-grid initialization is $\mathrm{O}(|\mathcal{P}|^2 c)$, where $|\mathcal{P}|$ is the number of input superpixels and c is the column number of target SP-grid. Due to the computation of correlated subgraphs, the complexity of maximizing column ρ is $\mathrm{O}(|\rho|^3 m)$, where $|\rho|$ is the number of real nodes and m is the number of dummy nodes to be added in the column.

5. Experimental Results

In this paper, we evaluate our approach and two state-of-the-art regular (or near-regular) SP methods, *i.e.* SuperLattice [17] and TurboPixel [14], by the task of object localization on benchmark datasets. Note, for TurboPixel, we simply assign the grid coordinates for each SP as the grid coordinates of its corresponding seed [14]. We also use two versions of pixel-level region covariance (RC) searching [23] as comparative baselines, *i.e.* a fast version with 5-pixel searching step and $[0.9, 1.1]$ size ratio, and a slower version with 2-pixel searching step and $[0.8, 1.2]$ size ratio. To demonstrate the generality of our approach, we have extended three widely-used irregular SP methods, SLIC [1],

[4] As we only update columns of $\mathcal{G}^*$, the overall coherence along vertical direction of any column is fully determined by $\mathcal{G}^{(0)}$ and is invariant to any configurations of that column. So, we can treat the vertical coherence component of $\mathrm{coh}(p,k,n)$ for the #i column as a constant V_i, which can be pre-computed using $\mathcal{G}^{(0)}$.

Method	Accuracy (exact query)	Accuracy (bndbox query)	SP Time	Matching Time
SP-Grid (SLIC) [1]	[37.0%, 67.2%, 97.6%]	[29.4%, 59.6%, 88.6%]	0.06s	0.03s
SP-Grid (EGS) [6]	[23.0%, 66.7%, 87.6%]	[23.0%, 62.5%, 87.6%]	0.05s	0.03s
SP-Grid (MS) [5]	[35.7%, 68.3%, 94.1%]	[30.6%, 61.5%, 87.7%]	1.60s	0.04s
SuperLattice [17]	[14.2%, 55.1%, 79.3%]	[21.0%, 52.5%, 80.5%]	0.07s	0.09s
TurboPixel [14]	[26.1%, 54.7%, 91.8%]	[10.3%, 44.4%, 77.3%]	1.72s	0.14s
Pixel-level (fast) [23]	[1.4%, 47.6%, 71.1%]	[1.3%, 42.6%, 70.2%]	–	4.09s
Pixel-level (slower) [23]	[13.8%, 48.8%, 79.3%]	[1.3%, 43.8%, 77.3%]	–	18.34s

Table 1: Average accuracy and speed of object localization using region covariance [23] on 50 benchmark image pairs. The accuracy rates are shown in the order of [min, mean, max] values. Note, SuperLattice [17] still needs additional time to pre-compute edge map, which is not counted in this table.

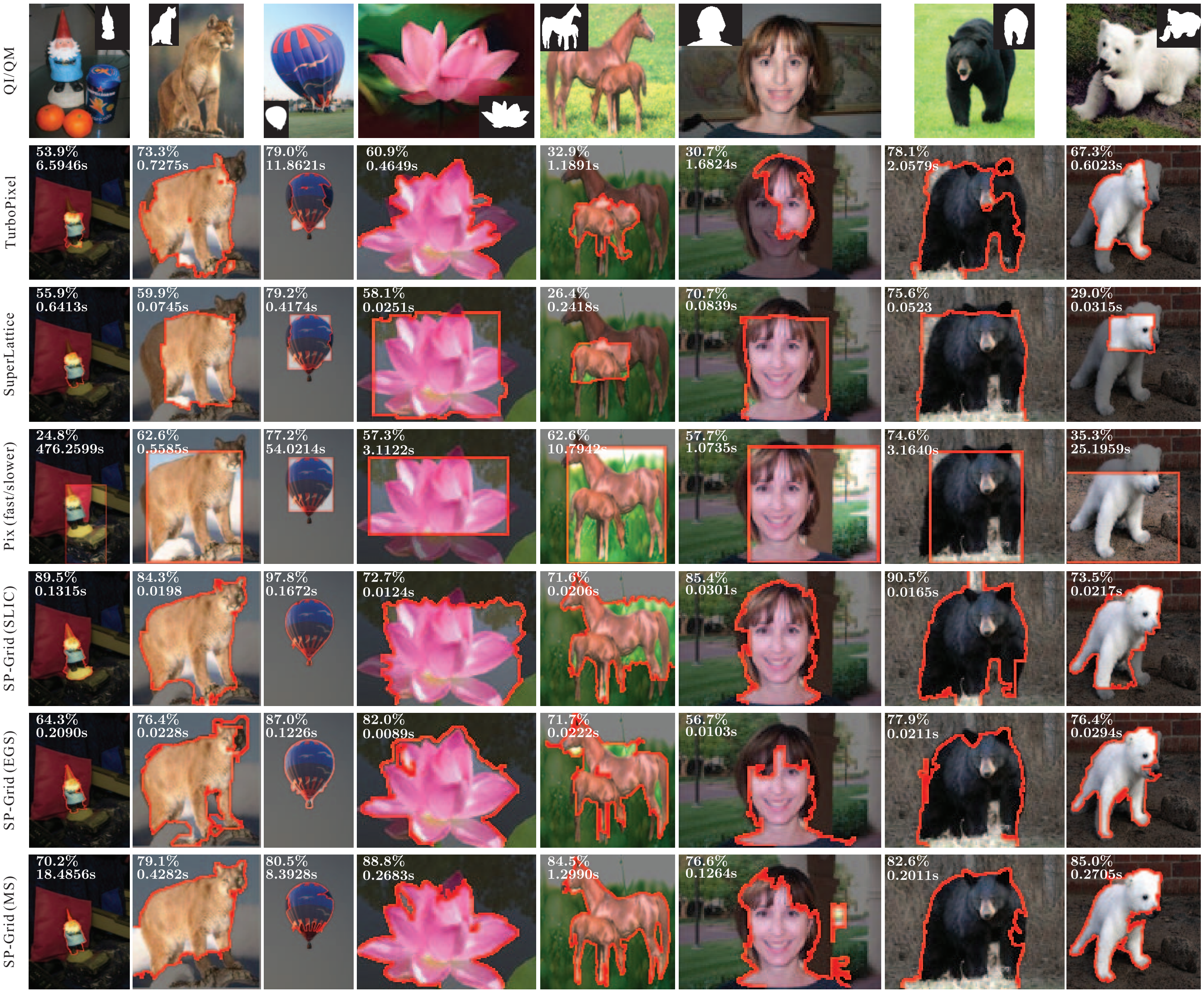

Figure 5: Performance comparison of object localization in benchmark co-segmentation image pairs. Note, Pix (fast/slower) means for each image we show the better detection results generated by Pix (fast) and Pix (slower).

EGS [6] and MeanShift (MS) [5], in the experiments. For the fairness of comparison, all SP-level methods used the same integral-image based searching strategy and RC features [23]. Besides, the experimental results were obtained using the original implementations of all tested methods. [5]

[5] The source code of our approach will be released soon.

For any object localization result R, in this paper, we measure its accuracy rate compared to ground truth GT as the cardinality ratio of their intersection and union sets $\mathrm{Acc}(R) = \frac{|R \cap \mathrm{GT}|}{|R \cup \mathrm{GT}|}$.

We first selected 50 image pairs from the benchmark image co-segmentation dataset [20], and compared the average performance of all seven tested methods. To show the best performance of all methods, for each image, every method produced multiple detection results using a group of reasonable SP-generation parameters, and used the best result for comparison. Table 1 shows the comparative results. We can clearly see that for all methods, exact query constantly leads to higher accuracy than query by bounding box. For pixel-level RC, searching by finer steps may lead to better results than using large steps, but with the cost of rapidly increased running time. The grid regularity of TurboPixel and SuperLattice help to quickly produce the detection results, with comparable (or better) accuracy to pixel-level methods. Among all test methods, the proposed SP-grid helps SLIC, EGS and MS to generate the highest detection accuracy using exact queries. Note that, except for MS, SP-grid based object localization is 100-500 times faster (including SP generation, SP-grid regularization and matching time) than pixel-level matching, while producing more than 15% accuracy improvements. All results of our approach reported in Table 1 were generated without objectness guidance.

Fig. 5 shows some real object detection results. We can see that compared to SuperLattice and TurboPixel, our approach tends to produce more accurate detection results better aligned to the real object boundaries. This is because, by maximizing the overall coherence, our approach preserves the most important image structures in the grid. For the first three images in Fig. 5, our approach used objectness guidance to refine the SP pair coherence matrix using $F = 10$.

In Fig. 6, we compare our approach and state-of-the-art methods on the tolerance of rotation and scaling in object localization, *i.e*. the object regions have increasing rotation and scaling variations, which may bring more difficulties to pixel-level detection. We can clearly see that the proposed approach is quite robust to both rotation and scaling variances. In contrast, the performance of baseline pixel-level RC degrades quickly for increasing scaling factors.

As shown in Fig. 7, we also tested object detection on multiple images. Similarly, we observe that our approach produces the best detection results. It seems that the spatial regularity term in TurboPixel and SuperLattice makes them prone to generate near-rectangle regions, which may lower their accuracy for detecting articulated objects, such as the cow and swan in Fig. 7, horse and bear in Fig. 5.

6. Conclusion

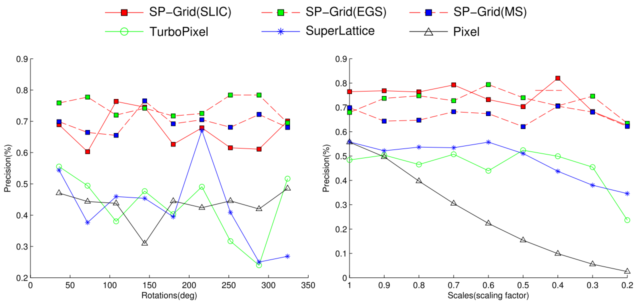

Figure 6: The robustness of our approach to rotation and scale variances in object localization.

We have proposed an efficient approach to regularize arbitrary superpixels into a regular grid by adding dummy nodes and maximizing the overall coherence. To the best of our knowledge, it is the first method served for this generic purpose. We also show how to incorporate regional objectness as an extra (optional) constraint to produce semantically more feasible SP-grids. As demonstrated by extensive experiments in object localization, our approach outperforms state-of-the-art methods in terms of both detection accuracy and speed. Besides, the proposed approach can be readily applied to a lot of other kinds of vision tasks, such as object co-segmentation and recognition etc.

Acknowledgements We appreciate anonymous reviewers and the area chair for their valuable comments. This work is supported by the Program for New Century Excellent Talents in University (NCET-11-0365), in part by the National Basic Research Program of China (2013CB329301), NSFC (61100121, 61100122 and 61272266), and the National Science and Technology Support Project (2013BAK01B01, 2013BAK01B05 and 2013BAH62F02).

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 34(11):2274–2282, 2012.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE TPAMI*, 34(11):2189–2202, 2012.

[3] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE TPAMI*, 12(9):855–867, 1990.

[4] S. An, P. Peursum, W. Liu, and S. Venkatesh. Efficient algorithms for subwindow search in object detection and localization. In *CVPR*, 2009.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI*, 24(5):1–18, 2002.

[6] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

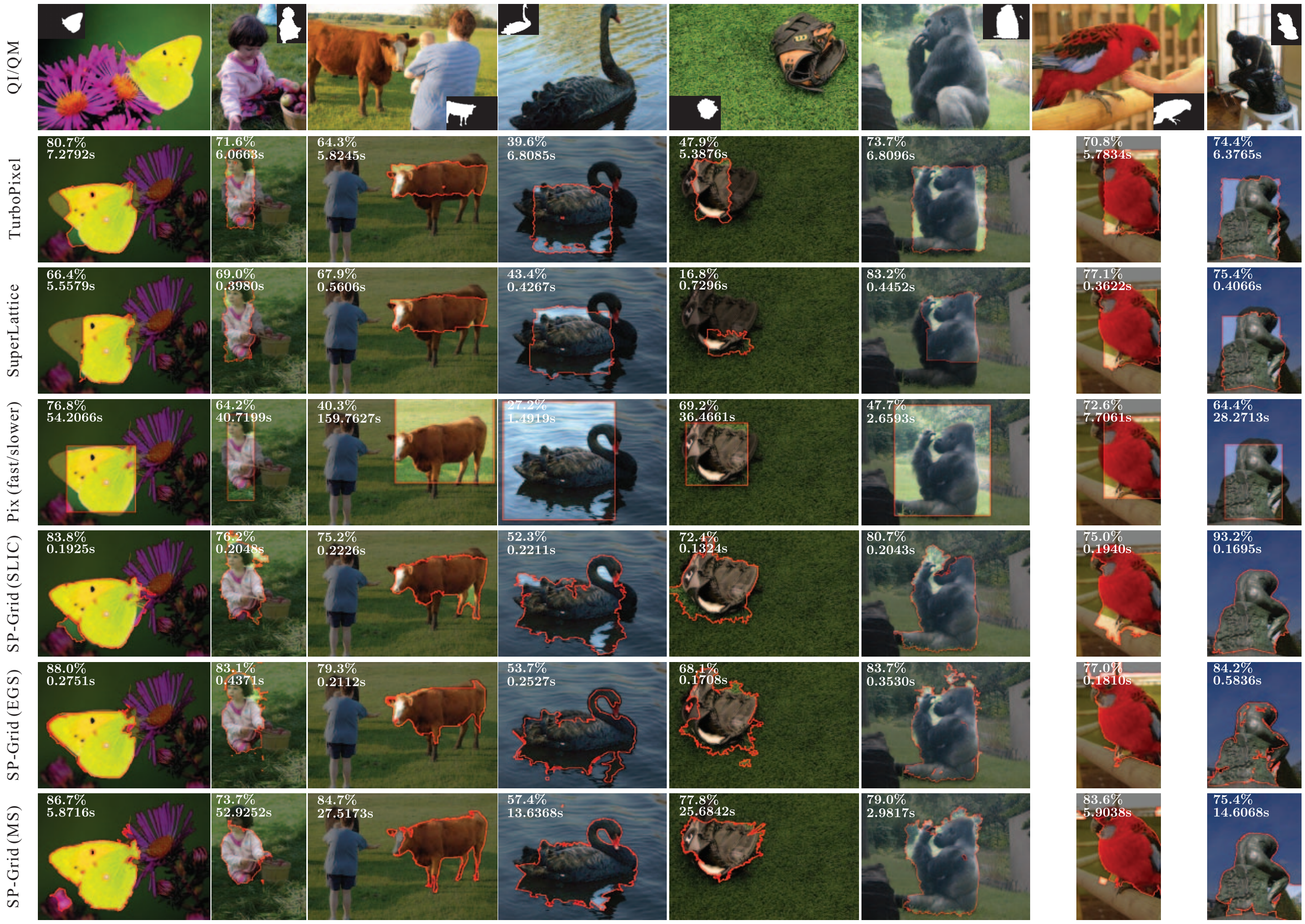

Figure 7: Performance comparison of object localization in multiple images on the MFC dataset [10].

[7] P. Felzenszwalb and R. Zabih. Dynamic programming and graph algorithms in computer vision. *IEEE TPAMI*, 33(4):721–740, 2011.

[8] W. Feng, J. Jia, and Z.-Q. Liu. Self-validated labeling of Markov random fields for image segmentation. *IEEE TPAMI*, 32(10):1871–1887, 2010.

[9] W. Feng and Z.-Q. Liu. Region-level image authentication using Bayesian structural content abstraction. *IEEE TIP*, 17(12):2413–2424, 2008.

[10] G. Kim and E. Xing. On multiple foreground cosegmentation. In *CVPR*, 2012.

[11] C. Lampert. Detecting objects in large image collections and videos by efficient subimage retrieval. In *ICCV*, 2009.

[12] C. Lampert, M. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE TPAMI*, 31(12):2129–2142, 2009.

[13] A. Levinshtein, C. Sminchisescu, and S. Dickinson. Optimal image and video closure by superpixel grouping. *IJCV*, 2012.

[14] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. TurboPixels: Fast superpixels using geometric flows. *IEEE TPAMI*, 31(12):2290–2297, 2009.

[15] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. In *ACM SIGGRAPH*, 2004.

[16] A. Moore, S. Prince, and J. Warrell. “Lattice cut” - constructing superpixels using layer constraints. In *CVPR*, 2010.

[17] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. In *CVPR*, 2008.

[18] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003.

[19] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, 2009.

[20] J. Rubio, J. Serrat, A. López, and N. Paragios. Unsupervised co-segmentation through region matching. In *CVPR*, 2012.

[21] M. Sargin, L. Bertelli, B. Manjunath, and K. Rose. Probabilistic occlusion boundary detection on spatio-temporal lattices. In *ICCV*, 2009.

[22] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

[23] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, 2006.

[24] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, 2010.

[25] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels in an energy optimization framework. In *ECCV*, 2010.