Foreground segmentation via co-saliency

Feba Grace George\(^1\), Nivy G S\(^2\)

\(^1\)PG Scholar, Department of Computer Science and Engineering, Musaliar College of Engineering and Technology, Kerala, India. febagsam@gmail.com

\(^2\)Assistant Professor, Department of Computer Science and Engineering, Musaliar College of Engineering and Technology, Kerala, India. nivy.g.s@gmail.com

**ABSTRACT**

Co-segmentation means segmenting the foreground objects from co-saliency map. Here, foreground objects are identified via cosaliency map and segment it using thresholding. Co-saliency identifies the saliency on the multiple images, which is a relatively underexplored area. Our method, using cluster-based algorithm for co-saliency detection. During the clustering process global correspondence between the multiple images is implicitly learned. To effectively compute the cluster saliency the three visual attention cues: contrast, spatial, and corresponding are calculated. By fusing the single image saliency and multimage saliency the final co-saliency map are generated. From co-saliency map obtain a binary image. And finally, segment the foreground objects using binary image from original image. The advantage of our method compared to existing is mostly bottom-up without heavy learning, simple and efficient.

**Key Words:** Clustering, Co-saliency map, Co-segmentation

1 **INTRODUCTION**

Visual saliency is an efficient way of capturing the most noticeable part in a scene, and can give the most usable cues. Saliency detection plays important roles in many image processing applications, such as regions of interest extraction and image resizing.

A co-salient region usually exhibits the following properties, i.e., 1) a salient region in an image should be prominent or noticeable with respect to its surroundings. 2) high similarity can be observed for such regions with respect to certain features (e.g., intensity, color, texture or shape).

Most of existing saliency algorithms formulate on detecting the salient object from an individual image. Recently, the multiple image correspondence based on a small image set has become one of the popular and challenging problems, and meanwhile the co-saliency is proposed. Co-saliency detection is firstly defined as discovering the salient objects in a group of similar images. The extracted co-saliency map is more useful in various applications, such as co-segmentation, common pattern discovery object co-recognition image retrieval and image summaries. Single image algorithm extracts salient objects in each image, but at the cost of having confusions with the complex backgrounds. Single image saliency method also ignores the relevance cues on the multiple images. In contrast, co-saliency utilizes repetitiveness property as additional constraint, and discovers the salient object on the multiple images.
A directly related application is motivated by the recent trend of co-segmentation. Most co-segmentation tasks are formulated as an energy optimization problem, including a within-image energy term and a global constraint term. These complex energy functions often cost significantly. Another thing is that most existing segmentation methods either need user interaction, i.e., use manually input strokes or bounding boxes.

Thanks to co-saliency detection method, its simplicity and efficiency automatically discriminates the salient objects. Nevertheless, we evaluate the proposed co-saliency method on a some of co-segmentation dataset, finding out that, despite the lack of complex learning, the performance of our co-saliency is rather competitive with recent co-segmentation methods.

So, the goal of our work is to segment the foreground object via a novel cluster-based algorithm of co-saliency detection for detecting the saliency on the multiple images. Our method employs the clustering to preserve the global correspondence among multiple images, and generates final co-saliency maps by fusing three effective bottom-up cues. A nice thing about our method is mostly bottom-up without heavy learning, simple and efficient.

## 2 PROPOSED SOLUTION

![Diagram of proposed system](image)

The proposed solution focuses efficient foreground segmentation of images. In this paper, we using a two-layer cluster-based method to detect co-saliency on the multiple images. Fig. 1 shows the work flow of proposed system. Given a set of images, our method starts by two-layer clustering. One layer groups the pixels on each image (single image), and the other layer associates the pixels on all images (multi-image). We then compute the saliency cues for each cluster, and measure the cluster-level saliency. The measured features include the uniqueness (on single/multi-image), the distance from the image center (on single/multi-image) and the repetitiveness (on multi-image). We call them contrast, spatial, and corresponding cues, respectively. Based on these cluster level cues, our method computes the saliency value for each pixel, which is used to generate the final saliency map.

Then generate a binary map of co-saliency map using threshold value. Finally, segment the foreground objects using binary map from original images.
2.1 Algorithm

**Input**: Input images

1. Single image saliency detection

   for each image do
     Clustering image into $K_1$ clusters;
     for each cluster do
       Computing contrast cue and spatial cue using Eq. (1) and Eq. (2) resp.;
     end
     Obtain final single image saliency map by combining two saliency cues.
   end

2. Multiple image co-saliency detection

   Clustering images into $K_2$ clusters;
   for each cluster do
     Computing contrast cue, spatial cue and corresponding cue using Eq. (1), Eq. (2) and Eq. (3) resp.;
   end
   Obtain multi image saliency map by combine two saliency cues

3. Co-saliency map

   Obtain by combining single and multi image saliency map.

4. Cosegmentation

   //----------Part 1: Binary map Generation----------
   for each image in co-saliency map do
     if pixel value > threshold value then
       pixel value = 1;
     end
   //----------Part 2: Extract foreground ----------
   for each image in binary map do
     if pixel value = 1 then
       extract that region from original image;
   end

**Output**: Foreground objects

2.2 Proposed Method Working

In our implementation, CIE Lab color and Gabor filter are employed to extract the feature vector. We compute the magnitude map of Gabor filter by combining 8 orientations as the texture feature. K-means++ is used in twolayer clustering. The cluster numbers are set to $K_1 = 6$ for intra image (single image), and $K_2 = \min(\max(2*\text{M},10),30)$ for inter image (multiple images), where $\text{M}$ denotes the image number.

Steps of Clustering algorithm: Kmeans++ are as follows:

1. Choose one center uniformly at random from among the data points.
2. For each data point $x$, compute $\text{D}(x)$, the distance between $x$ and the nearest center that has already been chosen.
3. Choose one new data point at random as a new center, where a point $x$ is chosen with probability proportional to $\text{D}(x)^2$.
4. Repeat Steps 2 and 3 until $k$ centers have been chosen.
5. Now that the initial centers have been chosen, proceed using standard k-means.
Next, compute the Contrast cue (on single/multi-image), Spatial cue (on single/multi-image) and Corresponding cue (on multi-image).

- **Contrast Cue**: Computes the visual feature uniqueness on the single or multiple images. The contrast cue, \( w^c(k) \) of cluster \( C_k \) is defined

\[
w^c(K) = \sum_{i=1,i\neq k}^{k} \left( \frac{n_i}{N} \| \mu^k - \mu^i \|^2 \right) \tag{1}
\]

where a L2 norm is used to compute the distance on the feature space, \( n_i \) represents the pixel number of cluster \( C_i \), and \( N \) denotes the pixel number of all images.

- **Spatial Cue**: Computes the distance between the object and the image center, which measures a global spatial distribution of the cluster. The spatial cue, \( w^s(k) \) of cluster \( C_k \) is defined as:

\[
w^s(K) = \sum_{j=1}^{M} \mathcal{N}(\| z_j^i - o_j^i \|^2 | 0, \sigma^2) \tag{2}
\]

Gaussian kernel \( \mathcal{N} \) computes the Euclidean distance between pixel \( z_j^i \) and the image center \( o_j^i \), the variance \( \sigma^2 \) is the normalized radius of images.

- **Corresponding Cue**: Computes how the cluster distributes on the multiple images, describing how frequent the object recurs. Then, our corresponding cue, \( w^d(k) \) is defined as:

\[
w^d(K) = \frac{1}{\text{var}(q^k) + 1} \tag{3}
\]

where,

\[
q^k = \frac{1}{M} \sum_{j=1}^{N_j} \delta[|b(p_j) - c^k|], j = 1 \ldots M
\]

where \( \text{var}(q^k) \) denotes the variance of histogram \( q^k \) of the cluster \( C_k \), which enforces condition that the sum of \( q^k \) = 1. The cluster with the high corresponding cue represents that the pixels of this cluster evenly distribute in each image. Then fusion of cues and saliency map are done by multiplication. The multiplication is better in depressing the noises than summation. Finally, generate a binary map based on the threshold value, \( t = 10 \). In the generated binary map, pixel value for foreground is one (white) and for background is zero (black). Then segment the foreground part by extracting the pixels from the original image which have value one in the binary map.
RESULTS AND PERFORMANCE EVALUATION

The quality comparison of these methods was performed by calculating the accuracy value.

\[
\text{Accuracy} = \left( \frac{\text{sum of foreground pixels detected in binary image}}{\text{sum of foreground pixels in ground truth}} \right) \times 100
\]

Table. 1 Segmentation accuracy (%)

<table>
<thead>
<tr>
<th>Method</th>
<th>Plane</th>
<th>Cow</th>
<th>Cat</th>
<th>Face</th>
<th>Bike</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our</td>
<td>88.1</td>
<td>92.1</td>
<td>87.8</td>
<td>88.5</td>
<td>77.6</td>
<td>86.8</td>
</tr>
<tr>
<td>[1]</td>
<td>86.8</td>
<td>90.4</td>
<td>83.5</td>
<td>84.8</td>
<td>72.3</td>
<td>83.6</td>
</tr>
</tbody>
</table>

Fig. 3  Output by proposed method
The accuracy results show that the proposed method for foreground segmentation is 3% better than the method mentioned in [1].

4 CONCLUSION

Here, propose an efficient and accurate method to segment the foreground object via a novel cluster-based algorithm of co-saliency detection. Segmentation is to pop potential objects out from the background. This task is made difficult by the wide variability of the object's shape, appearance, and its surrounding complex scene. Another thing is that most existing segmentation methods either need user interaction or have high computation cost. Typically, values on a saliency map can serve as beliefs of pixels' labels and thus are highly useful for segmentation. So in the proposed method, segment objects from visual scenes using such saliency priors.

In our method employs Kmeans++ clustering to preserve the global correspondence among multiple images and single image, and generates final co-saliency maps by fusing three effective bottom-up cues. A nice thing about this method is mostly bottom-up without heavy learning, efficient and no user interaction is required.

REFERENCES


