Causal Video Segmentation
Using Superseeds and Graph Matching

Vijay N. Gangapure1, Susmit Nanda1, Ananda S. Chowdhury1(✉), and Xiaoyi Jiang2

1 Department of Electronics and Telecommunication Engg,
Jadavpur University, Kolkata, India
aschowdhury@etce.jdvu.ac.in
2 Department of Mathematics and Computer Science,
University of Münster, Münster, Germany

Abstract. The goal of video segmentation is to group pixels into meaningful spatiotemporal regions that exhibit coherence in appearance and motion. Causal video segmentation methods use only past video frames to achieve the final segmentation. The problem of causal video segmentation becomes extremely challenging due to size of the input, camera motion, occlusions, non-rigid object motion, and uneven illumination. In this paper, we propose a novel framework for semantic segmentation of causal video using superseeds and graph matching. We first employ SLIC for the extraction of superpixels in a causal video frame. A set of superseeds is chosen from the superpixels in each frame using color and texture based spatial affinity measure. Temporal coherence is ensured through propagation of labels of the superseeds across each pair of adjacent frames. A graph matching procedure based on comparison of the eigenvalues of graph Laplacians is employed for label propagation. Watershed algorithm is applied finally to label the remaining pixels to achieve final segmentation. Experimental results clearly indicate the advantage of the proposed approach over some recently reported works.

Keywords: Causal video segmentation · Superseeds · Spatial affinity · Graph matching

1 Introduction

Video segmentation [1, 2, 7] aims at grouping pixels into meaningful spatiotemporal regions that exhibit coherence in appearance and motion. The problem of video segmentation [8–10] becomes extremely challenging due to size of the input, camera motion, occlusions, non-rigid object motion, and uneven illumination. Video segmentation techniques can be classified into non-causal (off-line) and causal (on-line) categories. While non-causal segmentation techniques make use of both the past and future video frames, causal segmentation approaches rely only on the past frames. For some recently reported causal video segmentation works, please see [3–6]. Some of these algorithms employ superpixels to
reduce computational complexity and to achieve powerful within-frame representation [3,6]. The method in [5] does not guarantee temporal consistency. Miksik et al. [4] performs semantic segmentation using optical flow to ensure temporal consistency. But, complexity of pixel-level optical flow computation poses a serious constraint for its use in real-time applications. Couprie et al. [3] proposed an efficient causal graph-based video segmentation method using minimum spanning tree. However, the method uses some heuristics in both the pre and post processing stages. In this paper, we propose a novel framework for semantic segmentation of causal video using superseeds and local graph matching [21]. The major contribution of the work is to propose a novel method of label propagation based on graph matching. Secondly, we have used superseeds for achieving better segmentation. Thirdly, unlike some of the existing approaches [3], we do not use any post-processing steps to achieve superior segmentation performance. Experimental results clearly indicate the advantage of the proposed approach over some of the recently published works [3–5]. The rest of the paper is organized in the following manner: in Section 2, we describe the proposed method. In Section 3, we present the experimental results along with necessary comparisons. We conclude the paper in Section 4 with an outline for directions of future research.

2 Proposed Method

The proposed framework is illustrated in Fig. 1 as shown below. SLIC [11] is applied for the generation of superpixels in each frame of a causal video. As a part of the initialization step, we apply the DBSCAN [13] method with some modifications resulting from our spatial consistency measure to achieve the final segmentation of the first frame. Some representative superpixels are then chosen using the above spatial affinity measure. We deem the centers of such superpixels as superseeds. Labels of these superseeds are propagated to the current frame from the previous frame by using local graph matching. Entries and exits are also handled efficiently to achieve temporal consistency. Watershed is applied

Fig. 1. Schematic of the proposed method
to label the remaining pixels (other than the superseeds) to achieve complete segmentation of the current frame.

2.1 Superpixel Extraction

Superpixel extraction significantly reduces computational complexity in video segmentation algorithms [3,6]. We use the SLIC algorithm [11] for the extraction of superpixels in each frame of a causal video. So, we can write:

$$I_{t,SLIC} = \text{SLIC}(I_t, k)$$  \hspace{1cm} (1)

where $I_t$ is the current frame and $I_{t,SLIC}$ is the frame with extracted superpixels. The inputs to SLIC are the current frame $I_t$ and the desired number of superpixels $k$. The CIELAB color space is used for clustering color images. In an initialization step, $k$ initial cluster centers $C_i, i = 1, ..., k$, are sampled on a regular grid with spacing $S$ pixels. Hence, we can write:

$$C_i = [l_i, a_i, b_i, x_i, y_i]^T$$  \hspace{1cm} (2)

$$S = \sqrt{\frac{N}{k}}$$  \hspace{1cm} (3)

where $N$ is the number of pixels in the image. The seed centers $C_i$ are moved to locations with lowest gradient position in $3 \times 3$ neighborhood. Then, each pixel $i$ is associated with the nearest cluster center. Limiting the size of search region to $2S \times 2S$ around the center significantly reduces the computation compared to the $k$-means clustering. A new distance measure $D$ which is a combination of color distance ($d_c$) in CIELAB space and spatial distance ($d_s$) is used for that purpose. The update step then adjusts each cluster center to be the mean $[l, a, b, x, y]^T$ vector of all the pixels of that cluster. For our work, we find 10 iterations to be sufficient to reach the convergence.

2.2 Spatial Consistency Measure

A hexagonal neighborhood graph $G = (V,E)$ is constructed with the extracted superpixels as the nodes using hexagonal grid as suggested by [20]. This is shown in Fig. 2. The spatial affinity between two superpixels $S_i$ and $S_j$ is captured by the edge weights $\omega_{ij}$. Color and texture information are used to compute these edge weights. For the color information, intersection (minimum) between cumulative color histograms of two superpixels under consideration is employed as a measure. This is given by:

$$c_{ij} = N \left[ \text{Hist}(S_i) \cap \text{Hist}(S_j) \right]$$  \hspace{1cm} (4)

Here, $\text{Hist}(\cdot)$ represents the cumulative color histogram of a superpixel. $N$ is the normalization constant, set equal to $1/\max(c_{ij})$. The larger the value of $c_{ij}$, the
higher is the color affinity between the superpixels \( S_i \) and \( S_j \). For the texture information measure, we use a gray-scale local binary pattern (LBP) \([12]\) based measure. The \( LBP_{P,R} \) number characterizes the local image structure and can be computed as follows:

\[
LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p
\]  

(5)

where \( p \) denotes a pixel having intensity \( g_p \) within a circular neighborhood of radius \( R \) centering the pixel \( c \) with intensity \( g_c \). We have chosen \( P=8 \) and \( R=1 \) for our problem. The function \( s \) is given by:

\[
s(x) = \begin{cases} 
1 & \text{if } x \geq 0 \\
0 & \text{if } x < 0 
\end{cases}
\]  

(6)

In fact, we compute the \( LBP \) binary vector corresponding to the above \( LBP \) number for every pixel in a superpixel. For a superpixel \( S_i \) of size \( n \), the texture measure is given by the ordered collection of \( n \) such individual vectors:

\[
\overline{ST}_i = \{LBP_{P,R,1}, LBP_{P,R,2}, \ldots, LBP_{P,R,n}\}
\]  

(7)

The normalized texture affinity measure \( t_{ij} \) between two superpixels \( S_i \) and \( S_j \) is given by:

\[
t_{ij} = 1 - \frac{W_H(\overline{ST}_i \oplus \overline{ST}_j)}{\max_{i,j}[W_H(\overline{ST}_i \oplus \overline{ST}_j)]}
\]  

(8)

where \( \overline{ST}_i \) is truncated to the length of \( \overline{ST}_j \) (assuming without loss of generality \( |\overline{ST}_j| < |\overline{ST}_i| \)), \( \oplus \) denotes the bitwise XOR operation, and \( W_H \) is the Hamming weight function on binary vectors. Larger value of \( t_{ij} \) indicates higher texture affinity. Finally, we present the proposed spatial affinity measure between the superpixels \( S_i \) and \( S_j \) as:

\[
\omega_{ij} = c_{ij} \times t_{ij}
\]  

(9)

Note that \( \omega_{ij} \in [0, 1] \).

### 2.3 Label Propagation Using Graph Similarity

We now mention the various steps linked with propagation of labels from the previous frame to the current frame. These steps are discussed below:

**Selection of Superseeds.** In the initialization step, only the first frame is segmented by the modified DBSCAN \([13]\) using the above spatial affinity measure. Each segment consists of multiple superpixels and we discard those segments which have less than two superpixels. The geometric centers of the remaining segments are extracted and treated as superseeds.
Local Graph Matching. Local region graphs are constructed surrounding each superseed in the previous frame and surrounding corresponding pixels (having same spatial locations as that of the superseeds in the previous frame) in the current frame. This is illustrated in Fig. 3. These two graphs are compared to propagate the label from the previous frame to the current frame. Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ respectively represent the local region graph surrounding a superseed in the previous frame and the local region graph surrounding the pixel with same spatial location (as that of the superseed in the previous frame) in the current frame. We use graph Laplacian’s eigenvalue-based score for matching [15]. Let $A_1$ and $A_2$ be the adjacency matrices, $D_1$ and $D_2$ be the diagonal matrices and $L_1$ and $L_2$ be the Laplacian matrices of the graphs $G_1$ and $G_2$ respectively. Then, we can write:

$$L_1 = D_1 - A_1 \tag{10}$$

$$L_2 = D_2 - A_2 \tag{11}$$
We use the similarity matching score $\text{Sim}_{G_1, G_2}$ between $G_1$ and $G_2$ by computing the top $k$ eigenvalues of Laplacians $L_1$ and $L_2$, that contain 90% of energy, as given by:

$$
\text{Sim}_{G_1, G_2} = \sum_{i=1}^{k} (\lambda_{1i} - \lambda_{2i})^2
$$

(12)

where $k$ is chosen as shown below:

$$
\min_j \left( \frac{\sum_{i=1}^{k} \lambda_{ji}}{\sum_{i=1}^{n} \lambda_{ji}} > 0.9 \right)
$$

(13)

Low values of $\text{Sim}_{G_1, G_2}$ indicate that the graphs are very similar and vice-versa.

**Temporal Consistency and Label Propagation.** If the matching score (see equation (12)) is less than an experimentally chosen threshold ($T_1$), then the two co-located regions under consideration have temporal coherence. So, we simply copy the label of the superseed of the previous frame to the next frame. If this score is higher, then there is no such temporal consistency between the two corresponding regions. This may occur due to an exit or a new entry in the current frame. To further differentiate between these two situations, we check the spatial affinity ($\omega_{ij}$) of the superpixel in the current frame with its neighbors in the local region graph. If the spatial affinity is more than an experimentally chosen threshold ($T_2$), it signifies an exit and no new label is required in that case. If the spatial affinity is less, it signifies an entry and we assign a new label to the superpixel in the current frame. In this manner, we ensure temporal coherence between each successive pair of frames under different situations (with or without entry and/or exit).

### 2.4 Watershed for Final Segmentation

We next employ the sequential unordered watershed algorithm with respect to topographical distance function [16], derived from the shortest path algorithm, to label the remaining pixels in the current frame to achieve the final segmentation. The basics of watershed transform following [16, 19] is included for the sake of completeness. Let $f$ be a gray value of the morphologically processed input frame (image). The lower slope $LS(p)$ at pixel $p$ is defined as the maximal slope linking $p$ to any of its neighbors of lower altitude. Thus,

$$
LS(p) = \max_{q \in N_G(p) \cup \{q\}} \left( \frac{f(p) - f(q)}{d(p, q)} \right)
$$

(14)

where $N_G(p)$ is the set of neighbors of pixel $p$ on the grid graph $G = (V, E)$ built on $f$ and $d(p, q)$ is the distance associated with the edge $(p, q)$. The cost of walking from a pixel $p$ to its neighboring pixel $q$ is defined as:

$$
cost(p, q) = \begin{cases} 
LS(p) \cdot d(p, q) & \text{if } f(p) > f(q) \\
LS(q) \cdot d(p, q) & \text{if } f(p) < f(q) \\
\frac{1}{2} (LS(p) + LS(q)) \cdot d(p, q) & \text{if } f(p) = f(q)
\end{cases}
$$

(15)
The topographical distance along a path $\pi$ between $p$ and $q$ is defined as:

$$ T_{\pi}^f(p, q) = \sum_{i=0}^{l-1} d(p_i, p_{i+1}) \cdot \text{cost}(p_i, p_{i+1}) $$

(16)

The topographical distance between $p$ and $q$ is the minimum of the topographical distances along all paths between $p$ and $q$ and is defined as:

$$ T_f(p, q) = \min_{\pi \in [p \rightarrow q]} T_{\pi}^f(p, q) $$

(17)

Let $(m_i)_{i \in I}$ be the collection of minima (markers) of $f$. The catchment basins $CB(m_i)$ of $f$ correspond to a minimum $m_i$ is defined as the basin of the lower completion of $f$:

$$ CB(m_i) = \{ p \in D \mid \forall j \in I \setminus \{i\} : f^*(m_i) + T_f^*(p, m_i) < f^*(m_j) + T_f^*(p, m_j) \} $$

(18)

where $f^*$ is the lower completion of $f$. The watershed of $f$ with 2D grid $D$ are the points which do not belong to any catchment basin and is defined in the following manner:

$$ Wshed(f) = D \cap (\cup_{i \in I} \cdot CB(m_i))^c $$

(19)

The superseeds generated in the earlier stage of our solution pipeline act as the markers (regional minima). Thus construction of the catchment basins (segments) of the frame becomes a problem of finding a path of minimal cost between each pixel and a marker (regional minima). Note that for the second frame onwards, the watershed-based final segmentation provides the labels of the superpixels in the current frame. We then propagate the labels of the superseeds in the current frame to the next frame using the graph matching technique.

3 Experimental Results

Experiments are carried out over two different types of datasets, one acquired with a static camera (NYU depth dataset) [14] and the other acquired with a moving camera (NYU Scene Dataset) [3,4]. To evaluate the performance, we use the overall pixel accuracy (OP) [18] metric. We have implemented the proposed method in MATLAB R2013b environment on a desktop PC with 3.4GHz Intel Core i7 CPU with 8GB RAM. SLIC for superpixels extraction is used from [11] and DBSCAN from [13]. The average execution time of the proposed method is 3.5 sec. out of which SLIC itself takes 3 sec. [20]. The values of the thresholds $T_1$ and $T_2$ are experimentally chosen as 0.45 and 0.50. To demonstrate the robustness of our method in terms of spatial consistency we compare our results with that of [3] and [5] in Fig. 4. For our experiment, we use 500 superpixels
Causal Video Segmentation Using Superseeds and Graph Matching

(an experimentally chosen value) for each frame. In case of the NYU scene dataset, the results are shown in Table 1. In this table, we compare our method with the results of frame by frame method, [4] and [3]. Table 1 clearly demonstrates the OP of our method (85.63) is superior as compared to that of the frame by frame (71.11), [4] (75.31), and [3] (76.27). We also show in Table 1 that the modified DBSCAN (OP: 85.63) yield better results than the standard DBSCAN (OP: 78.26). In Fig. 5, we present the comparison of our semantic segmentation with the ground truth and with that of [3] for five intermediate frames 55 - 59 of the NYU Scene dataset. The labeled images are overlaid on the original frames for better representation. The results clearly show that our output frames resemble the ground truth much better as compared to that of [3]. The quantitative results in terms of overall pixel accuracy (OP) for the NYU Depth dataset are presented in Table 2. We experiment with four videos from the NYU Depth dataset, namely, Dining room, Living room, Classroom and Office. Our proposed method (using modified DBSCAN) with an average OP of 72.32 surpasses both the frame-by-frame approach with an OP of 60.5 and that of [3] with an average OP of 61.6.

Fig. 4. Comparison of spatially consistent segments on different frames of Two women dataset [5] with independent segmentation

Table 1. OP values for the semantic segmentation task on the NYU Scene dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame by frame</td>
<td>71.11</td>
</tr>
<tr>
<td>Miksik et al. [4]</td>
<td>75.31</td>
</tr>
<tr>
<td>Couprie et al. [3]</td>
<td>76.27</td>
</tr>
<tr>
<td>Proposed method</td>
<td></td>
</tr>
<tr>
<td>DBSCAN [13] for initial frame</td>
<td>78.26</td>
</tr>
<tr>
<td>Modified DBSCAN for initial frame</td>
<td>85.63</td>
</tr>
</tbody>
</table>
Fig. 5. Comparison of temporally consistent semantic video segmentation on frames 55 - 59 of NYU Scene dataset

Table 2. OP for the semantic segmentation task on the NYU Depth dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Frame by Frame</th>
<th>Couprie et al. [3]</th>
<th>Proposed Method With Modified DBSCAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dining room</td>
<td>63.8</td>
<td>58.5</td>
<td>78.80</td>
</tr>
<tr>
<td>Living room</td>
<td>65.4</td>
<td>72.1</td>
<td>83.28</td>
</tr>
<tr>
<td>Classroom</td>
<td>56.5</td>
<td>58.3</td>
<td>65.55</td>
</tr>
<tr>
<td>Office</td>
<td>56.3</td>
<td>57.4</td>
<td>61.63</td>
</tr>
<tr>
<td>Mean</td>
<td>60.5</td>
<td>61.6</td>
<td>72.32</td>
</tr>
</tbody>
</table>

4 Conclusions

In this paper, we have presented a solution for the problem of causal video segmentation using superseeds and local graph matching. The superseeds are selected from the superpixels extracted using the SLIC algorithm. The labels of the superseeds are propagated using local graph matching. Finally, watershed algorithm is used to obtain the complete segmentation. In future, we will work on improving the execution time of our method. We will also explore how the segmentation accuracy can be further improved.

References