Multi-View Object Segmentation in Space and Time
Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, Francois Le Clerc, Patrick Pérez

To cite this version:
Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, Francois Le Clerc, Patrick Pérez. Multi-View Object Segmentation in Space and Time. ICCV 2013 - IEEE International Conference on Computer Vision, Dec 2013, Sydney, Australia. pp.2640-2647, 10.1109/ICCV.2013.328 . hal-00873544

HAL Id: hal-00873544
https://inria.hal.science/hal-00873544
Submitted on 26 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License
Multi-View Object Segmentation in Space and Time

Abdelaziz Djelouah 1,2 Jean-Sébastien Franco1 Edmond Boyer1
1LJK - INRIA Rhône-Alpes, France
François Le Clerc2 Patrick Pérez2
2Technicolor, Cesson Sevigne, France

Abstract

In this paper, we address the problem of object segmentation in multiple views or videos when two or more viewpoints of the same scene are available. We propose a new approach that propagates segmentation coherence information in both space and time, hence allowing evidences in one image to be shared over the complete set. To this aim the segmentation is cast as a single efficient labeling problem over space and time with graph cuts. In contrast to most existing multi-view segmentation methods that rely on some form of dense reconstruction, ours only requires a sparse 3D sampling to propagate information between viewpoints. The approach is thoroughly evaluated on standard multi-view datasets, as well as on videos. With static views, results compete with state of the art methods but they are achieved with significantly fewer viewpoints. With multiple videos, we report results that demonstrate the benefit of segmentation propagation through temporal cues.

1. Introduction

Segmenting objects of interest in images is a key prerequisite task for many applications of computer vision, e.g., matting and compositing in post-production, image indexing, video compression, and 3D reconstruction. Segmentation from multiple images of the same object has gained interest in recent years, as a means to remove the need for shape or appearance priors, or user interaction [20] in monocular approaches. This paper addresses the task of unsupervised multiple image segmentation of a single physical object, possibly moving, as seen from two or more calibrated cameras, which we refer to as multi-view object segmentation (MVOS), see Fig. 1 for a first example. As noted by [25], this is an intrinsically challenging problem, especially when the number of views is small, and viewpoints far apart. Indeed, it then becomes difficult to rely

...
and computational requirements, using only sparse inter-view links. Second, the graph structure intrinsically produces conservative and inclusive segmentations of the object of interest. Third, the ability to handle few viewpoints, much further apart than most state of the art approaches require. A situation that naturally arises in practice and for which none of the previous works is giving results below 8 viewpoints. Fourth, the framework straightforwardly extends to use of temporal links for multiple video sequences to propagate momentarily reliable segmentation evidences across time in multi-view setups. To the best of our knowledge, this is the first approach to leverage temporal cues for multiple video segmentation, with significant future applications.

2. Related work

2.1. Multi-View Object Segmentation

Zeng et al. [29] coined the problem, and proposed an initial rudimentary silhouette-based algorithm for building segmentations consistent with a single 3D object. Many methods follow this initial trend by building explicit 3D object reconstructions and alternating with image segmentations of the views based on foreground/background appearance models [7, 18, 11, 19]. Different object representations and cues are used, most often silhouette-based and volumetric [7], depth-based [10, 11], or stereo-based [16], and a range of techniques are used to regularize occupancy, enforcing smoothness criteria with graphcuts [7, 11], or global joint optimization of both [15]. A significant portion of existing works require user guidance and interaction [15, 28]. While generally a successful strategy, there is undeniable motivation to take dense 3D reconstruction out of the loop when processing a small number of viewpoints: image-based 3D models only achieve acceptable quality for a dozen views or more. Some of the most successful MVOS approaches to date [16] strongly rely on large number of viewpoints and small baseline. Our goal is to achieve equivalent quality with only a few, possibly widespread viewpoints. Our focus is therefore on how to propagate information between views and across time for consistent pixel labeling and not precise 3D modeling.

Propagating geometric consistency information from one view to another has proven surprisingly difficult. Indeed, the simple 3D definition of geometric consistency given above often leads to a complex counterpart in images with regions carved if no compound occupancy from other views is observed on epipolar lines of a pixel, e.g. [17, 22]. In [6] a graphcut/superpixel framework is used with constraints derived from epipolar geometry jointly with soft stereo and depth binning. This requires semi-circular setups with short baseline (as in [16]) and using specific heuristics to sparsify the superpixel interaction matrix with unclear complexity outcome.

We draw inspiration from a method that uses only a sparse 3D occupancy sampling of the scene [9], which proves to be a successful and efficient alternative to 3D reconstruction. While 3D samples embody spatial consistency between views, a specific construct is nonetheless still required to properly model information transfer between images and across time. In this paper we investigate graph representations for that purpose.

2.2. Cosegmentation Approaches

Cosegmentation was first coined in the work of Rother et al. [21] as the simultaneous binary segmentation of image parts in an image pair and by extension to more images [3, 14, 25]. The key assumptions of these methods is the observation of a common foreground region, or objects sharing appearance properties, versus a background with higher variability across images. As noted by [25], cosegmentation increasingly refers to diverse scenarios, ranging from user-guided segmentation to segmentation of classes of objects rather than instances. MVOS differs by only considering geometric cues for inter-view propagation of segmentations, and focuses on single object instances. Interestingly, some cosegmentation methods [13] have created tools to link segmentations across views based on appearance, formulating segmentation as a joint graph cut on the views. Similarly, we introduce a graph structure specifically relevant to propagate geometric cues for MVOS, rather than photometric cues.

2.3. Monocular Video Segmentation

Recent trends examine the use of temporal cues for monocular video segmentation. Such cues may be used to propagate manually specified segmentation information [26, 2, 27], or completely automated [8, 5]. Cues are propagated either deterministically based on e.g. optic flow [2], probabilistically by weighing different flow or link hypotheses [5, 27] or by learning low level variation statistics [8]. Interestingly, some approaches construct a graph over the full 2D+t volume to link segmentations in time [26], which we propose to unify in a single graph-based framework to include intra-view, inter-view and temporal links. To the best of our reading, our method is the first to propose such unification and temporal treatment of the MVOS problem.

3. Overview

We adopt the same definition of foreground as in [17]. That is, an object of interest should satisfy two constraints: be fully visible in all considered views, and its general appearance should be different from the background’s general appearance. To this end, we cast the MVOS problem as a joint labeling problem among the $n$ input views, and $t$ time steps if available, governed by a single MRF energy
4. Formulation

We are given a set of input images \( I^t = \{ I_1^t, ..., I_n^t \} \) at instant \( t \). For each image \( i \) at \( t \) we have the set \( P_i^t \) of its superpixels \( p \). We use superscript \( t \) for time for all terms, generally keeping it implicit for concision unless terms from different instants are involved. Segmenting the object in all the views consists in finding for every superpixel \( p \in P_i^t \) its label \( x_p \) with \( x_p \in \{ f, b \} \), the foreground and background labels. We denote \( S^t \) the set of 3D samples used to model dependencies between the views at instant \( t \). These points are uniformly sampled in the common visibility volume.

4.1. MRF Energy Principles

Given the superpixel decomposition and 3D samples (shown Fig. 1), we wish the MRF energy to reward a given labeling of all superpixels as follows, each principle leading to MRF energy terms described in the next subsections.

**Individual appearance.** The appearance of a superpixel should comply with image-wide foreground or background models, depending on its label.

**Appearance continuity.** Neighboring superpixels likely have the same labels if they have similar appearance.

**Appearance similarity.** Two superpixels with similar color/texture are more likely to be part of the same object and thus, more likely to have the same label. These superpixels may not be neighbors due to occluding objects, etc.

**Multi-view coherence.** 3D samples are considered object-consistent if they project to foreground regions with high likelihood.

**Projection constraint.** Assuming sufficient 3D sampling of the scene, a superpixel should be foreground if it sees at least one object-consistent sample in the scene. Conversely, a superpixel should be background if it sees no object-consistent 3D sample.

**Time consistency.** In the case of video data, superpixels in a sequence likely have the same label when they share similar appearance and are temporally linked through an observed flow field (e.g. optic flow, SIFT flow).

### 4.2. Intra-view appearance terms

We use the classic unary data and binary spatial smoothness terms on superpixels, to which we add non-local appearance similarity terms on superpixel pairs for broader information propagation and a finer appearance criterion.

**Individual appearance term.** We denote \( E_c \) the unary data-term related to each superpixel appearance. We characterize appearance by the sum of pixel-wise log-probabilities of being predicted by an image-wide foreground or background appearance distribution:

\[
E_c(x_p) = \left\{ \begin{array}{ll}
\sum_{r \in R_p} - \log H_i^B(I_r^t) & \text{if } x_p = b, \\
\sum_{r \in R_p} - \log H_i^F(I_r^t) & \text{if } x_p = f,
\end{array} \right.
\]

with \( R_p \) the set of pixels contained in superpixel \( p \). To model appearance we use a combination of color and texture histograms. In our case, \( I_r^t \) is an 11-dimension vector that includes both color and texture information. Appearance histograms are assumed to be shared for all frames of a given viewpoint for video sequences. Texture is defined as gradient magnitude response for 4 scales and Laplacian for 2 scales. As an initialization step, a k-means is run separately on color and texture values. This clustering is used to create texture and color vocabulary on which foreground and background histograms \( (H_i^F \) and \( H_i^B) \) are computed.
**Appearance continuity term.** This binary term, denoted $E_n$, discourages the assignment of different labels to neighboring superpixels that exhibit similar appearance. It is of the form of a contrast sensitive Potts model [4]. To model this similarity we use the previously defined texture and appearance descriptor of a given superpixel $p$ is noted $A_p$.

Let $N_{n,t}^i$ define the set of adjacent superpixel pairs in view $i$ at time $t$. For $(p, q) \in N_{n,t}^i$, the proposed $E_n$ is inversely proportional to the distance between the two superpixel descriptors, as follows:

$$E_n(x_p, x_q) = \begin{cases} \exp(-d(A_p, A_q)^2) & \text{if } x_p \neq x_q, \\ 0 & \text{otherwise.} \end{cases}$$

(2)

The distance $d(\cdot, \cdot)$ here is the $\chi^2$ distance between the superpixel descriptors. $< d(A_p, A_q) >$ indicates expectation over all neighboring superpixels.

**Appearance similarity term.** To favor consistent labels and efficient propagation among similar superpixels, we introduce a second binary term $E_a$ of the same form as $E_n$ but defined non-locally. Retrieving for each superpixel its $k$-nearest neighbors for $\chi^2$ distance, we define the set $N_{a,t}^i$ of similar superpixel pairs and for each of these pairs:

$$E_a(x_p, x_q) = \begin{cases} \exp(-d(A_p, A_q)^2) & \text{if } x_p \neq x_q, \\ 0 & \text{otherwise.} \end{cases}$$

(3)

### 4.3. Inter-view geometric consistency terms

To propagate inter-view information, we use a graph structure connecting a 3D sample to pixels it projects on. While this leads to a structure similar to [13], the latter builds inter-pixel hard links that are always active based on common histogram binning of pixels. A key difference we have to cope with is that geometric consistency of samples may change during iteration because of evolving segmentations. We thus evaluate before each iteration an “objectness” probability measuring consistency with current segmentation, and use it to reweigh the propagation strength of the sample, using a per-sample unary term as follows.

**Sample objectness term.** Let $P_s^t$ be the coherence probability of a sample $s \in S^t$. $P_s^t$ is computed using a conservative probability of common foreground coherence based on the view’s histogram sets, as in [9]. We associate a unary term and a label $x_s$ to sample $s$, allowing the cut algorithm the flexibility of deciding on the fly whether to include $s$ in the object segmentation, based on all MRF terms:

$$E_s(x_s) = \begin{cases} -\log(1 - P_s^t) & \text{if } x_s = b, \\ -\log P_s^t & \text{if } x_s = f. \end{cases}$$

(4)

**Sample-pixel junction term.** To ensure projection consistency, we connect each sample $s$ to the superpixels $p$ it projects onto in all views, which defines a neighborhood $N_s$. We define a simple binary term $E_j$ as follows:

$$E_j(x_s, x_p) = \begin{cases} \infty & \text{if } x_s = f \text{ and } x_p = b, \\ 0 & \text{otherwise.} \end{cases}$$

(5)

The key property of this energy is that, as shown in Fig.4, no cut of the corresponding graph may assign simultaneously to background a superpixel $p$ and to foreground a sample $s$ that projects on $p$. Thus it enforces the following desirable projection consistency property: labeling a superpixel $p$ as background is only possible if it is coherent to label all the samples $s$ projecting on it as background.
The converse property, inclusion of segmentations in the sample’s projected set, cannot be ensured: a superpixel can be labeled foreground even though it sees no foreground sample. This would require enforcing a foreground superpixel to see at least one foreground sample, which can only be expressed with higher order MRF terms. We opt to keep a first order MRF by modeling this behavior through an iteratively reweighed unary term, computed as follows:

\[
E_p(x_p) = -\log P(x_p|V_p) \quad \text{where} \quad V_p = \max_{s \in V_p} P_s^f
\]

### 4.4. Time consistency terms

In the case of video segmentation, the idea is to benefit from information at different instants and to propagate consistent foreground/background labeling for the frames of the same viewpoint. A set \( N_f \) of related superpixels between frames can be estimated by matching interest points or using optical flow. The propagation is done through the energy term \( E_f \) that enforces consistent labeling of linked superpixels \((p^t, q^{t+1}) \in N_f \) as follows:

\[
E_f(x_{p^t}, x_{q^{t+1}}) = \begin{cases} 
\theta_f \exp \left( -\frac{\bar{d}(A_{p^t}, A_{q^{t+1}})^2}{2} \right) & \text{if } x_{p^t} \neq x_{q^{t+1}}, \\ 
0 & \text{otherwise}. 
\end{cases}
\]

In this equation, \( \theta_f \) will depend on the considered links: in the case of SIFT based links, \( \theta_f \) is inversely proportional to the descriptor distance between the matched points. Thus, a good matching will constrain the two linked superpixels to have the same label. In the case of optical flow it will be proportional to the estimated flow quality.

### 4.5. MRF energy and graph construction

Let \( X \) be the conjunction of all possible sample and superpixel labels. Our MRF energy can thus be written with the three groups of terms: the intra-view group, the inter-view group with its own multi-view binary and unary terms, and finally the time consistency group with only binary terms between successive instants \( t \) and \( t+1 \). \( \lambda_1, \lambda_2, \lambda_3 \) are relative weighing constant parameters. Finding a multi-view segmentation for our set of images, given the set of histograms \( H^B_i \) and \( H^F_i \), and the probabilities \( P_s^f \), consists in finding the labeling \( X \) minimizing:

\[
E(X) = \sum_{t,i} \sum_{(p^t, q^{t+1}) \in N_f} E_f(x_{p^t}, x_{q^{t+1}}) + \\
\sum_{t,i} \sum_{p \in P_i^t} E_c(x_p) + \lambda_1 \sum_{(p,q) \in A_{i,t}^f} E_a(x_p, x_q) + \lambda_2 \sum_{(p,q) \in A_{i,t}^g} E_a(x_p, x_q) + \\
\sum_{t} \sum_{s \in S_i} \lambda_3 E_s(x_s) + \sum_{(s,p) \in N_s^t} E_p(x_p) + \sum_{i} \sum_{p \in P_i^t} E_p(x_p) .
\]

The submodularity constraint being satisfied in our model, we can build an \( s-t \) graph \( G \) where the min-cut will provide the solution to our energy minimization problem. This graph contains the two terminal nodes source and sink, one node for each superpixel and one node for each 3D sample \( s \). Edges are added between superpixels and samples according to the energy terms previously defined. Fig. 3 shows the resulting graph.

### 5. Computational approach

Similar to most of state of the art segmentation methods, we adopt an iterative scheme where we alternate between the previous graph cut optimization, and an update of the color models. The common visibility constraint can be used to initialize color models as in [17].

Fig. 5 gives an overview of the whole method. The extraction, description and linking of superpixels is done once, at initialization time. In the iterative process, the unary terms (objectness, superpixel sample projection and silhouette labeling probabilities) computed using the appearance models of the previous iteration. The algorithm converges when no more superpixels are re-labeled from an iteration to another. Superpixel labeling at convergence is used to estimate foreground/background appearance models which are used in a standard graphcut segmentation at pixel level, with unary terms based on appearance and smoothing binary terms using color dissimilarity.

In the case of video segmentation, the same scheme is applied over a sliding window of 5-10 frames. In this situation additional cues can be used, such as considering non-moving regions as background.
method, which project inside all views. We randomly generate 100k ing only 3D samples in the common visibility domain, i.e. 150 “words”. The region of interest is computed by keep-
ances models, we run K-means on texture and color val-
perpixel sizes of 30-50 pixels to ensure oversegmentation, 
Kolmogorov’s s-t mincut implementation [4]. We use su-
software for superpixel segmentation (SLIC [1]) and using 6.1. Experimental protocol

6. Experimental Results

6.1. Experimental protocol

We implemented our approach using publicly available software for superpixel segmentation (SLIC [1]) and using Kolmogorov’s s-t mincut implementation [4]. We use superpixel sizes of 30-50 pixels to ensure oversegmentation, obtaining around 2000 superpixels per image. For appearance models, we run K-means on texture and color values, to quantize texture and color into respectively 60 and 150 “words”. The region of interest is computed by keeping only 3D samples in the common visibility domain, i.e. which project inside all views. We randomly generate 100k 3D samples for all tests. The only free parameters in the method, $\lambda_1$, $\lambda_2$ and $\lambda_3$ were respectively set to 2.0, 4.0 and 0.05 for all datasets. No particular sensitivity was observed to these settings. Initialization of the algorithm was very weak, by setting $H_F^t$ to the statistics of the projection region of the common visibility domain of all views, which is quite large on all datasets, only eliminating about 25% of pixels on outer regions of the image. Background histograms $H_B^t$ are set to the statistics of the known background (outside the projection of visibility domain). Computation time depends on the number of viewpoints and the number of frames. In a static case with 10 viewpoints, each iteration of the algorithm takes less than 10s with our C++ implementation and convergence is reached in less than 10 iterations. Tests were run on a 2.3 GHz Intel i7 pc with 4GB memory.

6.2. Qualitative results

To validate our approach, we run our implementation on a dozen challenging datasets. Note that among the ex-
isting literature on the subject, few or no MVOS datasets are made publicly available, making comparisons difficult. We obtained datasets from two state of the art approaches: COUCH, BEAR, CAR, CHAIR1 from [16] which we use for qualitative and quantitative evaluation, BUSTE from [17] and PLANT which we use for qualitative evaluation.

The figures from 6 to 8 show the results for our methods on the various datasets. We show the graph cut result on superpixels at convergence and the final segmentation at pixel level. We illustrate the resilience of the algorithm in particular with low numbers of viewpoints on all the datasets. Very good results are obtained with only 3 widespread viewpoints (such as Fig[1]). This corresponds to a scenario where approaches that need numerous viewpoints, e.g. [16], are likely to fail.

Figure 5. Algorithm overview.

<table>
<thead>
<tr>
<th>Input images</th>
<th>Superpixels</th>
<th>Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Input images](Image 315x241 to 363x305)</td>
<td>![Superpixels](Image 315x461 to 382x511)</td>
<td>![Segmentation](Image 315x515 to 382x565)</td>
</tr>
</tbody>
</table>

Figure 6. Results on BEAR (3 views), COUCH (3 views), CAR (5 views) and CHAIR (9 views) datasets. First column corresponds to one of the input images. Second and third columns contain respectively superpixel and pixel level segmentation results.

<table>
<thead>
<tr>
<th>Input</th>
<th>8 views</th>
<th>13 views</th>
<th>26 views</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Input](Image 394x358 to 462x409)</td>
<td>![8 views](Image 394x461 to 461x511)</td>
<td>![13 views](Image 394x515 to 461x565)</td>
<td>![26 views](Image 394x413 to 461x457)</td>
</tr>
</tbody>
</table>

Figure 7. Results on BUSTE dataset with different numbers of views. For the 8 views result, the table is seen by all the views. With 13 views some cameras eliminate parts of the table and it is thus classified as background. Finally with all the views, the black elements in background appear close and similar to the black base. Thus only the head is identified as foreground in this case.

In complex scenarios, such as in Fig. 8 approaches relying only on color [9] fail to segment foreground objects, where our approach benefits from a more complex appearance model. Fig. 7 shows that what is considered as foreground object depends on the viewpoints. For the first ex-

---

1 from http://vision.in.tum.de/data/datasets/rgbd-dataset
ample with 8 viewpoints the table is seen by all the views and it is identified as part of the foreground. When adding more viewpoints, the table is no longer entirely seen by all the cameras, and thereby it is segmented as background. Using all the views, many cameras see the black elements in the background very close to the black base. They are then cut out from the foreground and only the statue is left.

6.3. Quantitative and Comparative results

To illustrate the strength of the approach and for the purpose of comparison, we use the same protocol as [16], computing accuracy as the proportion of correctly labeled pixels (Fig. 9). We evaluate here the sensitivity of our approach to the number of viewpoints and the quality of the segmentation result compared to state of the art approaches [9, 16, 25], by randomly picking 10 viewpoint subsets for a given tested number of viewpoints and averaging results.

Clearly Fig. 9 shows that our approach exhibits very little sensitivity to the number of viewpoints and achieves excellent segmentation results even with only 3 widespread viewpoints. Let us emphasize the excellent performance of the algorithm on CAR and CHAIR datasets, despite the very low number of viewpoints used and the challenging nature of color ambiguities in the datasets.

The difference of segmentation precision between approaches is mainly due to some difficult color ambiguities in the model, such as shadows that appear consistent both with hypothesis of geometric and photometric cosegmentation methods. In [16], it should be noted that depth information and plane detection significantly help, especially through the identification of the ground plane, which eliminates some ambiguities at the price of requiring more viewpoints for the purpose of obtaining the stereo.

6.4. Video segmentation results

In the case of video sequences, our framework has the ability to propagate multi-view segmentation evidences over time. It also enables to propagate temporal evidences from a given viewpoint to other viewpoints, e.g. static background or moving foreground. They can help resolve local segmentation ambiguities in few views in time or space. In order to demonstrate these principles, we evaluated the approach with two datasets DANCERS and HALF-PIPE from [11] and [12] respectively (more available as supplemen-

![Figure 8. Results on PLANT dataset (3 views) with qualitative comparison with [9]. Our method benefits from a richer appearance model and also from intra-image consistency constraints.](image)

![Figure 10. Multi-video segmentation results using space and time propagation (middle) vs. space only propagation (right) of information. row 1: DANCERS dataset; rows 2&3: HALF-PIPE dataset. User inputs are shown in blue (fg. region) and red (bg. region)](image)

2http://hal.inria.fr/hal-00873544

7. Conclusion

We have presented a new approach to solve the MVOS problem based on iterated joint graph cuts. To our knowledge we propose the first unified solution dealing with intra-view, inter-view, and temporal cues in a multi-view image and video segmentation context, into a single consistent MRF model. The approach is shown to cope with a low number of widespread viewpoints, many times achieving state of the art quality with only three wide baseline views. The algorithm has been demonstrated on very challenging datasets, including MVOS segmentation with videos from four moving handheld cameras. We believe that the framework is a solid basis to explore more complex multi-view
motion models, which we suspect may even further improve segmentation quality in the video MVOS problem context.

References


