Cotemporal Multi-View Video Segmentation

Abdelaziz Djelouah[1], Jean-Sébastien Franco[1], Edmond Boyer[1], Patrick Pérez[2], George Drettakis[1]

[1]Inria [2]Technicolor

Abstract

We address the problem of multi-view video segmentation of dynamic scenes in general and outdoor environments with possibly moving cameras. Multi-view methods for dynamic scenes usually rely on geometric calibration to impose spatial shape constraints between viewpoints. In this paper, we show that the calibration constraint can be relaxed while still getting competitive segmentation results using multi-view constraints. We introduce new multi-view cotemporality constraints through motion correlation cues, in addition to common appearance features used by co-segmentation methods to identify co-instances of objects. We also take advantage of learning based segmentation strategies by casting the problem as the selection of monocular proposals that satisfy multi-view constraints. This yields a fully automated method that can segment subjects of interest without any particular pre-processing stage. Results on several challenging outdoor datasets demonstrate the feasibility and robustness of our approach.

1. Introduction

Multi-view video segmentation is the process of jointly extracting foreground regions in multiple videos of the same dynamic scene. Video segmentation is of interest for many applications that need to identify objects in temporal image sequences for further processing such as tracking, editing, reconstruction or recognition. Recent work in this field has demonstrated the benefit of multi-view over monocular strategies when several viewpoints of the same objects are available. Such multi-view contexts are becoming more common, especially with the advent of cheap commodity cameras that make simultaneous recordings easy. However, in such situations and especially with unconstrained outdoor environments and moving devices, geometric relationships between viewpoints, *i.e.*, calibration, can be difficult to obtain. In this paper, we propose a new solution to multi-view segmentation in this *uncalibrated* situation.

Multi-view segmentation strategies can be roughly divided in two categories in the literature. *Co-segmentation* approaches aim at segmenting several instances of an object or a class of objects in different views or different videos using appearance similarities. This object-oriented strategy has demonstrated its efficiency over monocular strategies, and without calibration. Here we consider a more specific situation where simultaneous videos of the *same* scene are available. In this case, cotemporality provides additional multi-view constraints that successfully complement the appearance similarity constraint. On the other hand, *multi-view object segmentation* approaches have been proposed, relying on geometric consistency between views. As mentioned earlier, they require camera calibration that can be challenging in unconstrained environments. Our approach relaxes this constraint by exploiting dynamic coherence of foreground regions in different views in addition to appearance and structural coherence.

We propose a new solution to the multi-video segmentation problem in a general context, through the use of structural, appearance and motion information and without the need for calibration. First, one can observe that a dynamic scene should exhibit similar motion patterns over different viewpoints, *i.e.*, temporal coherence, in addition to similar appearance patterns. We exploit this intuition by introducing movement pattern histograms [5] as a key feature for the multi-video segmentation problem, in particular for cross-view matching. Second, we enforce region-wide matching of these features in image space by embedding the feature responses in a spatial graph structure and matching these graphs across views [30]. Third, recent segmentation methods that build on object-like properties ("objectness") have demonstrated their ability to detect objects in videos. We leverage this ability by casting multi-view segmentation as a selection process over monocular proposals and under multi-view constraints, as provided by the matched graphs. We show that this simple yet efficient strategy yields high quality results, in particular outperforming state-of-the-art co-segmentation approaches, and yielding results competitive with multi-view segmentation approaches on data where calibration was possible, even though our method does not use it. To summarize, our main contributions are:

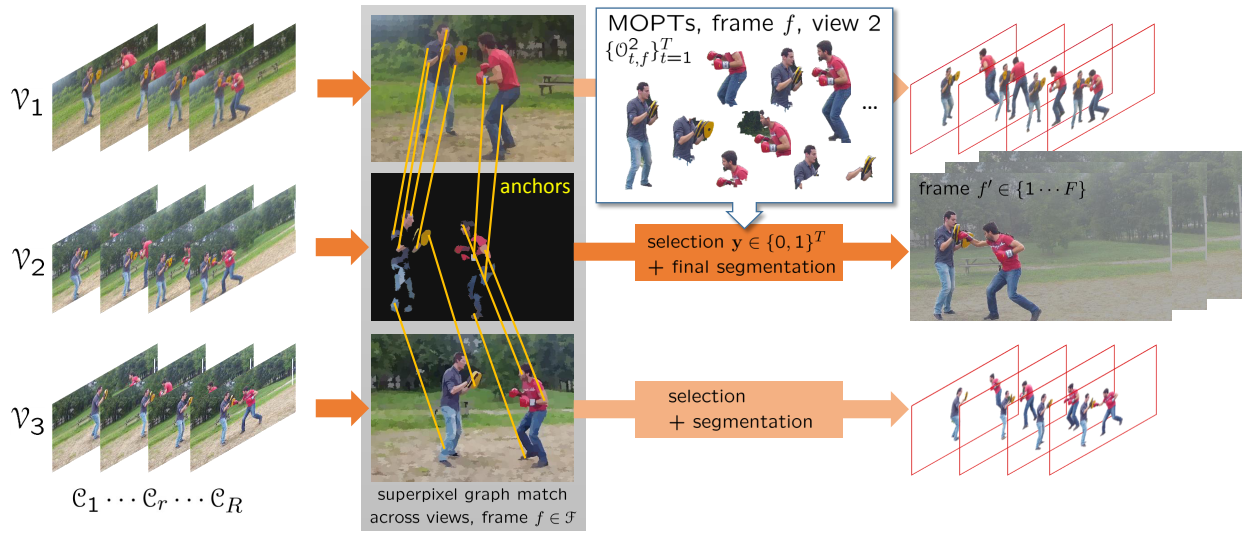

Figure 1: **Overview of the cotemporal multi-view video segmentation method**. Synchronized videos $\mathcal{V}_n$s of the same scene are partitioned into short clips $\mathcal{C}_r$s. At a small number of instants $f \in \mathcal{F}$ where motion is sufficiently informative, cross-view correspondences between superpixels with similar appearance and motion are obtained by graph matching. In each view n, matched superpixels, which are likely to lie on moving foreground objects, are used as sparse anchors to guide the selection process among a large pool $\{\mathcal{O}_t^n\}_{t=1}^T$ of moving objects proposals extracted from all clips [10].

1. A graph based approach for multi-view region matching that exploits motion descriptors.

2. A generic framework that leverages multi-view matching constraints to select from monocular segmentation proposals.

After reviewing existing methods (§2), we provide a detailed overview of the proposed approach (§3) and we discuss monocular video segment proposals (§4). Our multi-view motion constraint and selection methodology are respectively described in (§5) and (§6). (§7) describes the finalization step before evaluation (§8).

2. Related work

Calibrated Multi-View Segmentation. When calibration is available, multi-view segmentation methods usually enforce some form of geometric consistency between silhouettes of the different views, such that segmentation information can be propagated from one view to the other. Propagation along the epipolar lines was among the first strategies tested [33, 19, 12, 3], where a change in pixel decision in one view influences an epipolar line or band in other views. A number of methods consider background segmentation and scene reconstruction simultaneously, with the rationale that both tasks are mutually cooperative and can be alternated. The reconstruction can either be a discrete volumetric shape, *e.g.*, voxels with assigned deterministic or probabilistic occupancy [9, 17, 25]. Stereo cues have been used as an alternative 3D representation of the subject of interest [18, 23]. While all previously mentioned methods consider only a static scene, temporal information has recently been used in this context as a complementary way to propagate segmentation cues along a given video as well as between views [7]. While calibration can be obtained in a number of multi-view situations, it places additional constraints on the acquisition protocol and sequence processing and is not always achievable, especially with a low number of wide baseline views of the same scene. We instead consider more generic situations where calibration is not required, as useful *e.g.*, for moving cameras, while still using the *cotemporal* property of this type of sequences, *i.e.*, that they are simultaneous takes of a single common event.

Co-segmentation. Co-segmentation approaches [28] are a complementary category of methods which address simultaneous segmentation of objects in different pictures. Co-segmentation encompasses several sub-categories [31], with a subset of the methods aiming at the segmentation of various instances of the same class, and another subset interested in segmenting multiple images of the same instance (not necessarily at identical times). These methods also depart from multi-view segmentation methods in that they usually do not assume available calibration and thus do not propagate any cross-view geometric cues. Furthermore they strongly rely on appearance similarity to propagate segmentation information [28, 14, 15, 31]. More recently, additional structural cues have been used to constrain propagation of information [30] and relate regions in different images. We extend this type of approach to the

case of multiple, cotemporal videos. While initially applied to single subjects in a static scenario or with several time distinct pictures of the same object, co-segmentation has recently been extended to the multi-video case [29, 4, 11, 34]. Rubio *et. al.* [29] extract temporal tubes and match them among views based on local features. Chiu *et. al.* [4] use bag of words to segment several classes of objects in multiple videos. Zhang *et al.* [34] generate a graph on object tracklets where cliques of nodes correspond to the same object. Fu *et. al.* [11] reason on object-like candidates. However the bulk of these methods still rely on a rather fragile assumption of low-level appearance similarity and do not specifically address the case of cotemporality. Our approach examines efficient use of the cotemporality hypothesis by introducing additional distinctive motion-based features, and also uses the objectness prior by selecting segments among likely object proposals.

3. Overview

To perform foreground-background segmentation in multi-view videos, we rely on the following assumptions: (a) the objects of interest to be segmented are objects commonly observed moving relative to a quasi-static background; (b) we assume cotemporality of videos, *i.e.*, they are simultaneous synchronized takes of a common event in time. We do not assume cameras are static, *i.e.*, viewpoints can undergo motion as long as objects of interest are visible.

Our proposed method is as follows (Fig.1). First, each video is divided into clips of 15 to 20 frames. These short clips are preprocessed using automatic video segmentation [10]. Using the author's implementation we observed that this setting corresponds to the best compromise between clip length and region propagation performance. The result of this step is a set of segmentation proposals for each clip of each video camera. Second, we select the *foreground* segment proposals for each clip. Since foreground is defined as the objects seen by all the cameras, consequently it corresponds to image regions with similar motion. For this purpose we isolate frames with most salient non-global motion and use graph matching to link superpixels from different viewpoints, using both appearance (color and texture) and motion [5] descriptors to estimate superpixel affinities. In the case of a moving camera, we compensate for background induced motion using an affine transform. The graph matching step provides regions that are consistent across viewpoints, that should thus be part of the foreground and act as priors on the selection of segmentation proposals. As a third step, we select proposals which include as many of the matched superpixels as possible, while satisfying temporal continuity in each viewpoint. The selection of proposals is expressed as an energy minimization problem and a genetic algorithm [6] is used as a heuristic search method. Finally pixel-level segmentation is achieved with standard methods [26] over a sliding window of frames using the selected regions as initialization.

4. Monocular Video Segmentation Proposals

Using category independent segmentation candidates [8] is a common practice in many recent methods for video segmentation [21] and co-segmentation [11, 34]. Using a set of candidates in the segmentation introduces a notion of semantic that is extremely useful when color information is ambiguous. Work related to visual saliency [22] explores the same ideas in order to find regions of interest in a given image based on the color distribution. In the case of videos, clustering of trajectories [24] can also be another source of segmentation candidates. In related work on video segmentation [20], this preprocessing is done on a frame by frame basis generating a large set of candidates for a few seconds of video. For a higher level task like multi-view segmentation, one can reason on temporally propagated proposals to avoid an excessive complexity. In this work we use the method proposed by Fragkiadaki *et al.* [10]. In the following[1] we consider a set of N synchronized input videos (views) of the same scene, $\mathcal{V}^n = \{\mathbf{I}_1^n \cdots \mathbf{I}_F^n\}$, $n = 1 \cdots N$, each with F frames. Note that for conciseness, view index superscript and frame index subscript will be dropped when unnecessary. The segmentation method [10] extracts a set of Moving Object Proposal Tubes from each video, noted MOPTs from now on. They are obtained through the extraction of instantaneous moving object proposals and the associated clustering of key-point trajectories. In the case of long sequences with complex motion, and using the author's implementation, we observed that the proposed segments often deviate from the original objects. To circumvent this problem, we split each video into R short clips of 15 frames, $\mathcal{C}_r = \{\mathbf{I}_{f_{r-1}} \cdots \mathbf{I}_{f_r}\}$, with $f_0 = 1$ and $f_R = F$. Two consecutive clips share one frame, *i.e.*, $\mathcal{C}_r \cap \mathcal{C}_{r+1} = \{\mathbf{I}_{f_r}\}$.

For a given video $\mathcal{V}$, this clip-wise segmentation yields a total collection of T spatio-temporal MOPTs, $\mathcal{O}_t \in \mathcal{P} \times [\![1, F]\!]$, $t = 1 \cdots T$, where $\mathcal{P}$ denotes the pixel grid. In frame f, the pixels associated to MOPT $\mathcal{O}_t$, if any, form a set $\mathcal{O}_{t,f} \subset \mathcal{P}$ (see Fig. 1).

We formulate the problem of cotemporal multi-view segmentation as the one of selecting a subset of MOPTs with the help of multi-view constraints. In absence of calibration, we propose a method relying on apparent motion cues that will play a key role in the identification of the foreground objects of interest.

5. Multi-view Constraints from Movement

A key aspect of multi-view segmentation is to take advantage of the inter-view information. In this section

[1] Throughout, we use a standard font for scalars (X), bold for vectors ($\mathbf{X}$), sans serif for matrices (X) and curvilinear for sets ($\mathcal{X}$).

we explain how motion is used as a supplementary cue to identify regions simultaneously seen by all the cameras. These regions will later act as constraints for the selection of MOPTs.

View invariant motion descriptor. To compare and recognize actions from different viewpoints, Ciptadi *et al.* [5] use the correlation between orientations of 2D apparent movement in different views. In each camera, motion is estimated using optical flow [2] and all motion vectors are clustered according to their orientation. Accordingly, a histogram of 2D motion directions, weighted by motion amplitudes, is built at each instant in each view. Pearson correlation coefficients computed between time series of bin counts permit to establish correspondences between motion directions across views. Leveraging this knowledge, it is possible to assess that the two very different motions undergone by a same scene element in two widely separated views are in fact related. More generally, if quantized motion directions that match across views are ordered identically, histograms of optical flow orientations can be compared bin-to-bin in a view invariant fashion.

To identify corresponding image fragments in different viewpoints, we propose a *graph matching* framework with motion-based descriptors that can be compared in a meaningful way across any pair of views. Each descriptor is an amplitude-weighted histogram of coarsely quantized motion directions.

Graph Matching. Graph matching plays an important role in solving matching problems in computer vision. Multi-view segmentation is a typical situation where one tries to match and identify similar regions in different images. In the following we describe how graph matching is estimated between superpixels of different views using appearance and motion descriptors.

Each input image is oversegmented into superpixels using SLIC [1]. Since the local spatial arrangement of these small segments can vary drastically from one view to another, the graph that is built over them does not rely on spatial proximity but rather on motion and appearance similarities. We equip each superpixel with an appearance-motion descriptor, and construct a symmetrized nearest-neighbour graph in this descriptor space. Since motion cues must be key to our construct, we restrict ourselves to "moving" superpixels, *i.e.*, those with more than half of pixels exhibiting non negligible residual optical flow with respect to dominant scene motion.

More formally, a video frame $\mathbf{I} = (I_p)_{p\in\mathcal{P}}$ is partitioned into superpixels. We retain the ones in motion, $\mathcal{S}_k \subset \mathcal{P}$, $k = 1 \cdots K$. Each of these K image fragments is equipped with a three-fold descriptor $\mathbf{f}_k = (\mathbf{f}_{\mathrm{c},k}, \mathbf{f}_{\mathrm{t},k}, \mathbf{f}_{\mathrm{m},k}) \in \mathbb{R}^{305}$. The color part, $\mathbf{f}_{\mathrm{c},k}$, is a normalized histogram associated to 250 centroids in `Lab` color space. The appearance description is complemented by a 50-dimensional texture descriptor $\mathbf{f}_{\mathrm{t},k}$ obtained by binning 4-scale intensity gradient amplitudes and 2-scale Laplacians. This is similar to many state of the art methods in segmentation [32, 7]. Finally, the motion descriptor $\mathbf{f}_{\mathrm{m},k}$ is built as explained above, using 5 motion direction centroids matched across views. The distance between two superpixels is the sum of the χ^2 distances between color, texture and motion sub-descriptors.

Using this distance in descriptor space, we build the undirected superpixel graph $\mathcal{G} = ([\![1, K]\!], \mathcal{N})$ as the symmetrized 5-NN graph. The k-th superpixel has a neighorhood $\mathcal{N}_k$ of at least five other superpixels with similar appearance and motion. Figure 2(a) shows examples of this graph structure.

We are now ready to define the graph matching problem. Given two views m and n and a given instant, we aim at matching their superpixel graphs $\mathcal{G}^m$ and $\mathcal{G}^n$. Denoting $\mathsf{X} = [x_{k\ell}]_{K^m\times K^n}$ the unknown binary matching matrix, the matching cost reads:

$$J(\mathsf{X}) = \sum_{k=1}^{K^m}\sum_{\ell=1}^{K^n} x_{k\ell}K(\mathbf{f}_k^m, \mathbf{f}_\ell^n) + \sum_{k=1}^{K^m}\sum_{\ell=1}^{K^n}\sum_{k'\in\mathcal{N}_k^m}\sum_{\ell'\in\mathcal{N}_\ell^n} x_{k\ell}x_{k'\ell'}, \tag{1}$$

where affinities between nodes are measured by an additive kernel over color, texture and motion descriptors of corresponding superpixels:

$$\begin{aligned} K(\mathbf{f}_k, \mathbf{f}_\ell) &= \exp\Big(-\frac{\chi^2(\mathbf{f}_{\mathrm{c},k}, \mathbf{f}_{\mathrm{c},\ell})}{2\langle d_c\rangle}\Big) \\ &+ \exp\Big(-\frac{\chi^2(\mathbf{f}_{\mathrm{t},k}, \mathbf{f}_{\mathrm{t},\ell})}{2\langle d_t\rangle}\Big) + \exp\Big(-\frac{\chi^2(\mathbf{f}_{\mathrm{m},k}, \mathbf{f}_{\mathrm{m},\ell})}{2\langle d_m\rangle}\Big). \end{aligned} \tag{2}$$

Normalizations parameters, $\langle d_{\mathrm{c}}\rangle$, $\langle d_{\mathrm{t}}\rangle$ and $\langle d_{\mathrm{m}}\rangle$ are average χ^2 distances between all possible matching pairs of superpixels on respectively color, texture and motion descriptors.

The matching cost J is minimized under the constraint that X defines a one-to-one mapping, *i.e.*, it is a partial permutation matrix:

$$\begin{aligned} &\text{minimize } J(\mathsf{X}) \\ &\text{w.r.t. } \mathsf{X} \in \{0,1\}^{K^m\times K^n} \\ &\text{sb.to } \mathsf{X}\mathbf{1}_{K^n} \le \mathbf{1}_{K^m},\ \mathsf{X}^\top\mathbf{1}_{K^m} \le \mathbf{1}_{K^n}. \end{aligned} \tag{3}$$

This quadratic assignment problem is a special case of the one addressed in Zhou and De la Torre [35], with node affinities being equal to one. We thus resort to the fast method proposed in [35].

Figure 2(b) shows some superpixels correspondences across views that are obtained as a result. Many of them are correct despite large changes of view-points, demonstrating the relevance of motion cues as a complement to appearance resemblance for bridging uncalibrated views. A

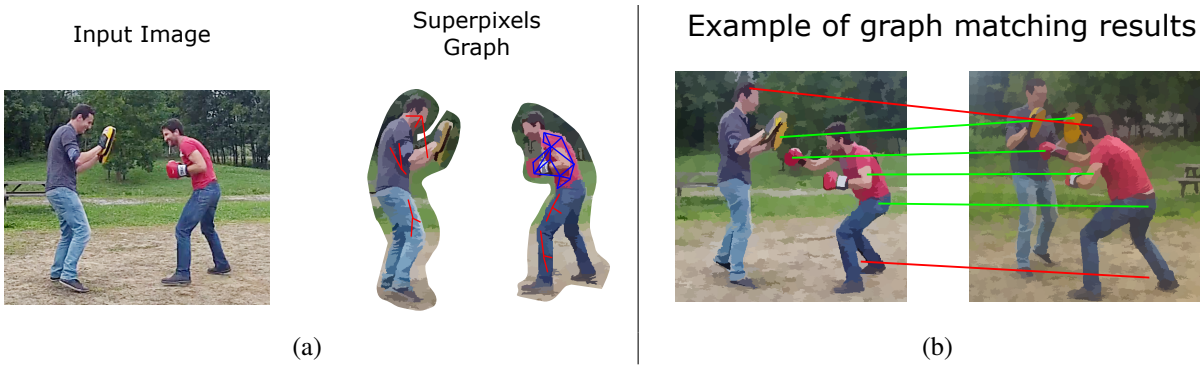

Figure 2: **Superpixel graphs and their matching across views**. (a) In each frame of each view, a graph is constructed over "moving" superpixels. Each of these superpixels is linked to its 5 closest neighbors in appearance–motion descriptor space (red links, only 3 displayed for clarity). In resulting graphs, tightly linked groups of superpixels with similar color, texture and movement emerge (blue sub-graphs). (b) Using graph matching, it is possible to find correspondences between superpixels from different views. Those with highest descriptor similarities are kept (green) while most erroneous matches are filtered out (red).

fraction of spurious correspondences is nonetheless present. A first source of error lies in inaccuracies of the optical flow, which cause the inclusion of non-moving superpixels in the graph matching. A second problem is simply the absence of good matches for superpixels that are occluded in some views (even if they belong to an object of interest that is visible in all views). Ranking all matched pairs (k, ℓ) with $x_{k,\ell} = 1$ by decreasing order of distance between descriptors and retaining only the best pairs permit to filter out most erroneous matches. In our tests, keeping 70% of the matched pairs was empirically found to perform well. Remaining superpixels, which are confidently matched with other views, are likely to be part of a foreground object, seen by all the cameras. The final selection of MOPTs in each view will thus be guided by these superpixels.

6. Multi-View Selection of MOPTs

The previous superpixel graph matching cannot be performed reliably on all frames of the videos because motion is not present (nor discriminant) at all instants. For that reason, we only use the graph matching on some frames. More precisely, for every group of 30 consecutive frames we use the frame where the total magnitude of optical flow is maximum. We shall denote $\mathcal{F} \subset [\![1, F]\!]$ the index subset of these frames. For $f \in \mathcal{F}$, the superpixels in video n that have correspondences in other views form a set of "anchor" pixels that we shall denote $\mathcal{M}_f^n \subset \mathcal{P}$. These pixels are key locations for view n: The MOPTs that cover as many of these anchor locations in each video are likely to be multi-view consistent. Overall, the MOPTs that are finally selected in each view should have the following desired properties:

- *Multi-view consistency.* As explained above, the pixel sets $\mathcal{M}_f^n,\ f \in \mathcal{F}$ encompass regions that are likely to be foreground and they should be segmented as such. Consequently, one needs to select the MOPTs that produce the maximum coverage of these regions. Several MOPTS may be needed to cover the entire mask.
- *High objectness.* If only multi-view consistency was used, then retaining all the MOPTs would produce the best coverage of anchor regions. By taking into account a "moving objectness" score, we favour the proposals that are more likely to be a whole object.
- *Temporal continuity.* Since the MOPTs are proposed on short clips, it is important that selected ones exhibit spatio-temporal continuity at clip boundaries.

After cross-view graph matchings have been performed, multi-view constraints are simply enforced through resulting anchor pixels within each video. Therefore, it is now possible to process each view independently. In the following, we describe the selection of relevant MOPTs within a single video for which sets $\mathcal{M}_f,\ f \in \mathcal{F}$, of anchor pixels are given.

Given the T MOPTs in this view, $\mathcal{O}_t$, $t = 1 \cdots T$, we aim to select those associated to foreground objects of interest. Formally, we seek a binary labelling $\mathbf{y} = (y_t)_{t=1}^T$ of all MOPTs –with $y_t = 1$ for a selected MOPT, 0 otherwise– that fulfils at best the above criteria, namely multi-view consistency, moving objectness and continuity across successive clips. To this end, we maximize the following threefold score:

$$S(\mathbf{y}) = \sum_{f \in \mathcal{F}} \log\left(\frac{|\mathcal{M}_f \cap \mathcal{X}(\mathbf{y}, f)|}{|\mathcal{M}_f|}\right) + \sum_{t=1}^{T} y_t \log(o_t) + \sum_{r \in [\![1, R-1]\!]} \log\left(\frac{|\mathcal{X}^-(\mathbf{y}, f_r) \cap \mathcal{X}^+(\mathbf{y}, f_r)|}{|\mathcal{X}^-(\mathbf{y}, f_r) \cup \mathcal{X}^+(\mathbf{y}, f_r)|}\right), \quad (4)$$

(a) Objectness (b) Anchor Cover (c) Anchor Cover (d) Proposed

Figure 3: **Selecting MOPTs**. For a given video sequence, several segmentation proposals are available. (a) Using only "moving objectness" score, one might select moving background elements (not visible in all views) as the foreground object. (b) Maximizing the coverage with anchor regions from graph matchings reduces this problem. (c) However, this coverage criteria is often not sufficient to capture whole objects. (d) Using our proposed selection method, it is possible to select the proposals that together result in a better segmentation of the foreground objects. In this case, 2 MOPTs were selected.

with the following notations:

- $\mathcal{X}(\mathbf{y}, f) = \cup_{t:y_t=1} \mathcal{O}_{t,f}$ is the set of pixels in frame f covered by the MOPTs that are selected according to labelling $\mathbf{y}$.
- $\mathcal{X}^-(\mathbf{y}, f_r)$, resp. $\mathcal{X}^+(\mathbf{y}, f_r)$, is the set of pixels in boundary frame f_r that are covered by selected MOPTs in clip $\mathcal{C}_r$, resp. $\mathcal{C}_{r+1}$.
- Scalar o_t is the moving objectness score defined in [10] and averaged over the tube $\mathcal{O}_t$.

The first part of score $S(\mathbf{y})$ encourages the selected MOPTs to cover large proportions of view-consistent regions in frames where such regions are defined as a result of successful graph matching. The second term favors MOPTs with highest moving objectness, which is especially useful when several MOPTs compete over the same anchor pixels. The third and final term enforces spatio-temporal consistency of the selected proposals over adjacent clips.

Due to the first and third terms, which are global and not sub-modular, maximizing this score is a combinatorial problem that does not lend itself to optimization techniques such as graph cuts. Instead, search heuristics such as genetic algorithms offer suitable choices. In our case, we use the Matlab implementation of the method proposed by Kusum *et al* [6] which was empirically found to deliver good quality local optima.

7. Finalization and implementation details

In the previous step, we select in each view the MOPTs that best explain the matched superpixels while favoring inter-clip continuity and high objectness score. It is however important to note that the pool of MOPTs does not always contain the segmentations of the foreground objects. Sometimes the candidates miss out parts of the moving objects or include portions of the background. Figure 4 shows two examples where the selected MOPTs do not produce a good segmentation of the foreground. To overcome this limitation we estimate a final pixel-level segmentation guided by the selected MOPTs.

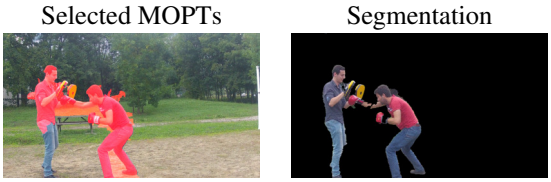

Figure 4: **From selected MOPTs to final segmentation**. As MOPTs do not precisely delineate moving objects in general, those selected by the proposed method (red overlays) also suffer from such problems Conducting a final pixel-level segmentation based on selected MOPTs improves the quality of the final results.

For this step we initialize color models at a frame $f \in \mathcal{F}$ so that the score $\frac{|\mathcal{M}_f \cap \mathcal{X}(\mathbf{y},f)|}{|\mathcal{M}_f \cup \mathcal{X}(\mathbf{y},f)|}$ is maximum, indicating a good fit of the selected MOPTs with the graph matching anchors in $\mathcal{M}_f$. Foreground and background color models are initialized using $\mathcal{X}(\mathbf{y}, f)$, the segmentation result from the selected MOPTs.

To segment the entire video, we use a simple spatio-temporal graph cut over a 10-frame sliding window. A classic graph structure is used, based on a contrast weighted spatial graph and temporal consistency links derived from optical flow. The MRF energy to be minimized is:

$$E = \sum_p \phi_p(l_p) + \sum_{\{p,q\} \in \mathcal{N}} \lambda \phi_s. \tag{5}$$

Multi-view consistency is enforced as selected MOPTs are taken into account by defining the following unary potentials $\{\phi_p\}_{p \in \mathcal{P}}$:

$$\phi_p(1) = -\max\Big(\log P_{\text{fg}}(\mathbf{I}_p), [p \in \mathcal{X}(\mathbf{y}, f)] \log(0.5)\Big),$$
$$\phi_p(0) = -\max\Big(\log P_{\text{bg}}(\mathbf{I}_p), [p \notin \mathcal{X}(\mathbf{y}, f)] \log(0.5)\Big)$$

where $[.]$ is Iverson bracket, and P_{fg} and P_{bg} are the foreground and background color model respectively. For pixels in a selected MOPT, $\phi_p(1)$ can't exceed $\log(2)$ while

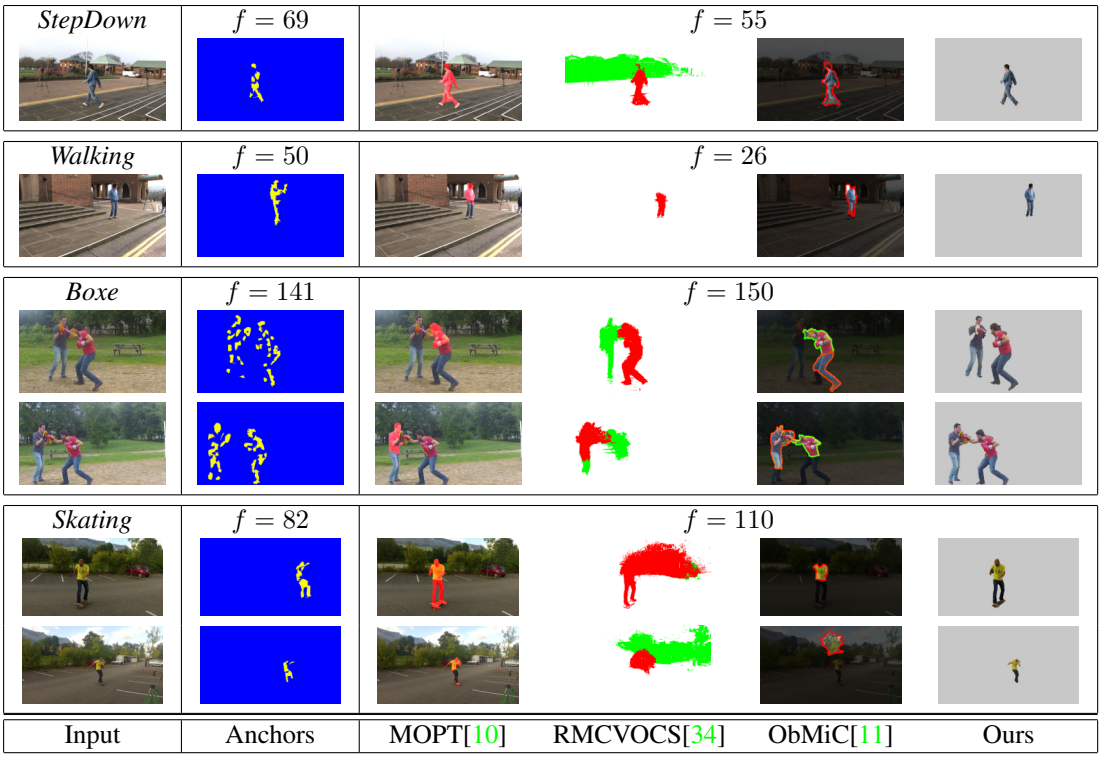

Figure 5: **Results and comparisons**. For each dataset, the leftmost column corresponds to two different input views. The second column contains the anchors for the closest instant f where graph matching was estimated. Finally the last columns are, from left to right, the MOPT [10] with highest objectness score, Video Co-Segmentation results for RMCVOCS [34] and ObMiC [11] and finally segmentation results using our method.

$\phi_p(0)$ is not bounded, which biases the labelling of this pixel toward 1 (foreground), and conversely for pixels not in a selected MOPT. ϕ_s is the smoothness term over the set of neighbor pixels in space and time ($\mathcal{N}$). It can be any energy that favors consistent labeling in homogeneous regions. In our implementation we use a simple inverse distance between neighbor pixels.

8. Results

In this section the method is evaluated on different multi-view video datasets. We use the sequence *HalfPipe* (3 cameras) from [13], two sequences *StepDown* and *Walking* (4 cameras) from [16] and we also propose two new sequences *Boxe* (3 cameras) and *Skating* (4 cameras). In each dataset the cameras are synchronized and the calibration information ignored except for comparison. The *HalfPipe* and *Skating* datasets include moving cameras. In this case, we compensate for background induced motion using an affine transform.

Our method. The first step is to estimate graph matching between superpixels of different views. This results in the anchor pixels shown in Fig. 5. The graph matching is only estimated for some frames, which present strong apparent motion (in practice about 1 graph matching every 30 frames). This is illustrated in Fig. 5 which shows for each input frame on the leftmost column, the corresponding closest frame with anchors. To obtain the set of monocular segmentation proposals, we use the authors Matlab implementation of MOPTs [10]. The set of proposals can vary significantly between datasets and video clips. To illustrate this, the third column in Fig. 5 shows the proposal with the highest motion objectness score. Depending on the scene, the set of proposals can include different parts of foreground and background elements. In this case, we can see that using the anchors, obtained with the graph matching, we can select appropriate MOPTs and obtain therefore a good pixel level segmentation of the foreground (For all results: https://hal.inria.fr/hal-01367430).

Comparisons with Co-Seg. We compare our results with the closest related work in video co-segmentation [34, 11] using the authors implementations available online (with the default parameters for all tests). Results obtained using the regulated maximum weight cliques method [34] (referenced as RMCVOCS in figures) are illustrated in the fourth column in Fig. 5. This methods automatically identifies the number of shared objects in different videos. The same object gets the same color in the result image. A common error of this method is to identify many elements of

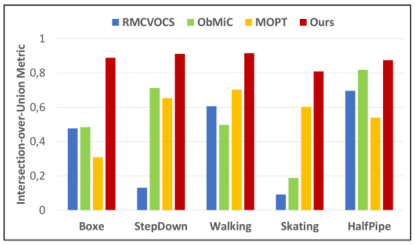

Figure 6: Evaluation with video co-segmentation methods: The chart shows performance using *Intersection-over-Union* metric (Higher is better).

	MOPT [10]	RVOCS [34]	ObMiC [11]	Ours
Boxe	0.71	0.98	0.72	**0.11**
HalfPipe	0.52	0.43	0.19	**0.13**
Skating	0.45	13.54	2.31	**0.19**
StepDown	0.43	6.95	0.35	**0.09**
Walking	0.32	0.53	0.55	**0.08**

Figure 7: Evaluation with video co-segmentation methods using the ratio between pixel errors and total number of foreground pixels (lower is better).

the background as shared objects. Considering only the first object, this approach performs well on the *StepDown* and *Walking* sequences.

The second video co-segmentation method [11] (referenced as ObMiC in figures) also relies on object segmentation candidates. It can handle several objects but their exact number must be specified at runtime. For all the datasets the number of objects was set to 1 except for the *Boxe* sequence were it was set to 2. We can see that both video co-segmentation methods suffer from similar issues resulting from inherent ambiguity in the appearance cues. It should also be noted that using spatio-temporal object candidates [10] helps reducing the pool of candidates to the most relevant ones in terms of appearance and motion.

A quantitative evaluation of these methods is proposed in figures 6 and 7, using the *intersection-over-union* metric [4] and the ratio between pixel errors and foreground pixels. It shows that our method outperforms both video co-segmentation methods. Interestingly, these two methods perform better when the foreground object exhibits more appearance similarity between viewpoints. This is particularly true for the *HalfPipe* dataset where the foreground is a single object with the same black color in all views. On the contrary, the *Skating* dataset is more challenging as the size of the object is smaller. In this case, video co-segmentation methods tend to identify background regions as the shared object. When the scene is composed of a single moving object, choosing the MOPTs with the best score can be a good starting point for the segmentation but as soon as the number of distracting objects increases in the scene,

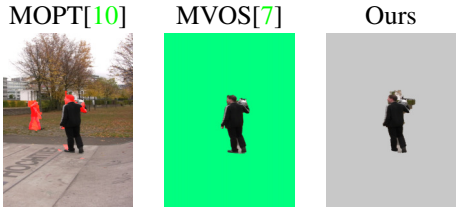

Figure 8: **Comparison with a multi-view object segmentation** (MVOS [7]). Our approach achieves competitive results without the need for calibration that can be cumbersome in such contexts.

other sources of information must be considered and we can clearly see the advantage of a multi-view strategy.

Comparison with the calibrated case. Finally we compare our method with a multi-view segmentation method [7] that exploits calibration to enforce multi-view consistency of foreground segmentation. Figure 8 shows for two viewpoints: the MOPT with the highest score, the segmentation obtained in [7] and the segmentation obtained with our method. These results demonstrate that the proposed approach can provide segmentations that are comparable to the calibrated case, though they are less precise (*e.g.*, near the head in Fig. 8). This is an important feature of the approach since calibration is not always available or often cumbersome to estimate, in particular with moving cameras and wide baselines as in the example [13].

Computation time. It can be broken down as follows: First, optical flows and moving object proposals [10] for each video clip (15 frames) are estimated. This task can take up to 5 min for each clip, but clips can be processed in parallel. Graph matching step takes around 2 min and is estimated every 30 frames. Using a multi-threaded implementation of the genetic algorithm [6], the proposal selection step converges in 5 to 10 minutes, depending on the number of proposals. Finally 1 to 2 minutes are needed for the pixel level segmentation.

9. Conclusion

In this paper we have presented a new approach to solve the cotemporal multi-view segmentation problem without calibration. The proposed approach reasons on monocular spatio-temporal object proposals. Using multi-view constraints, proposals likely to correspond to the foreground object are selected. The evaluation on different datasets demonstrated better performance than video co-segmentation methods that use only appearance. We believe the proposed framework is a solid basis to explore more complex multi-view datasets.

Acknowledgments

Research funded by EU FP7 project 611089 CR-PLAY and French ANR project SEMAPOLIS (ANR-13-CORD-0003).

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. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.

[2] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):500–513, 2011.

[3] N. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Automatic object segmentation from calibrated images. In *CVMP*, 2011.

[4] W.-C. Chiu and M. Fritz. Multi-class video co-segmentation with a generative multi-video model. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 321–328. IEEE, 2013.

[5] A. Ciptadi, M. S. Goodwin, and J. M. Rehg. Movement pattern histogram for action recognition and retrieval. In *Computer Vision–ECCV 2014*, pages 695–710. Springer, 2014.

[6] K. Deep, K. P. Singh, M. L. Kansal, and C. Mohan. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, 212(2):505–518, 2009.

[7] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, and P. Perez. Multi-view object segmentation in space and time. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2640–2647. IEEE, 2013.

[8] I. Endres and D. Hoiem. Category independent object proposals. In *Computer Vision–ECCV 2010*, pages 575–588. Springer, 2010.

[9] T. Feldmann, L. Dießelberg, and A. Wörner. Adaptive foreground/background segmentation using multiview silhouette fusion. In *DAGM-Symposium*, 2009.

[10] K. Fragkiadaki, P. Arbeláez, P. Felsen, and J. Malik. Learning to segment moving objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4083–4090, 2015.

[11] H. Fu, D. Xu, B. Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3166–3173. IEEE, 2014.

[12] J.-Y. Guillemaut and A. Hilton. Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. *International journal of computer vision*, 93(1):73–100, 2011.

[13] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 224–231. IEEE, 2009.

[14] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009.

[15] A. Joulin, F. R. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[16] H. Kim and A. Hilton. Influence of colour and feature geometry on multi-modal 3d point clouds data registration. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 202–209. IEEE, 2014.

[17] K. Kolev, T. Brox, and D. Cremers. Fast joint estimation of silhouettes and dense 3d geometry from multiple images. *IEEE Trans. PAMI*, 34(3):493–505, 2011.

[18] A. Kowdle, S. N. Sinha, and R. Szeliski. Multiple view object cosegmentation using appearance and stereo cues. In *ECCV*, 2012.

[19] W. Lee, W. Woo, and E. Boyer. Silhouette segmentation in multiple views. *IEEE Trans. PAMI*, 33(7):1429–1441, 2010.

[20] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002. IEEE, 2011.

[21] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2192–2199. IEEE, 2013.

[22] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2):353–367, 2011.

[23] A. Mustafa, H. Kim, J. Guillemaut, and A. Hilton. General dynamic scene reconstruction from multiple view video. *CoRR*, abs/1509.09294, 2015.

[24] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(6):1187–1200, 2014.

[25] C. Reinbacher, M. Rüther, and H. Bischof. Fast variational multi-view segmentation through backprojection of spatial constraints. *Image and Vision Computing*, 30(11):797–807, 2012.

[26] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, 2004.

[27] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[28] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, 2006.

[29] J. C. Rubio, J. Serrat, and A. López. Video co-segmentation. In *Computer Vision–ACCV 2012*, pages 13–24. Springer, 2013.

[30] J. C. Rubio, J. Serrat, A. López, and N. Paragios. Unsupervised co-segmentation through region matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 749–756. IEEE, 2012.

[31] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[32] T. Wang and J. Collomosse. Progressive motion diffusion of labeling priors for coherent video segmentation. *IEEE Transactions on Multimedia*, 14(2):389–400, April 2012.

[33] G. Zeng and L. Quan. Silhouette extraction from multiple images of an unknown background. In *ACCV*, 2004.

[34] D. Zhang, O. Javed, and M. Shah. Video object co-segmentation by regulated maximum weight cliques. In *Computer Vision–ECCV 2014*, pages 551–566. Springer, 2014.

[35] F. Zhou and F. De la Torre. Factorized graph matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 127–134. IEEE, 2012.