Instance-level video segmentation from object tracks

Guillaume Seguin[1,*] Piotr Bojanowski[2,*] Rémi Lajugie[2,†] Ivan Laptev[2,*]
[1]École Normale Supérieure [2]Inria

Abstract

We address the problem of segmenting multiple object instances in complex videos. Our method does not require manual pixel-level annotation for training, and relies instead on readily-available object detectors or visual object tracking only. Given object bounding boxes at input, we cast video segmentation as a weakly-supervised learning problem. Our proposed objective combines (a) a discriminative clustering term for background segmentation, (b) a spectral clustering one for grouping pixels of same object instances, and (c) linear constraints enabling instance-level segmentation. We propose a convex relaxation of this problem and solve it efficiently using the Frank-Wolfe algorithm. We report results and compare our method to several baselines on a new video dataset for multi-instance person segmentation.

1. Introduction

Semantic object segmentation in images and videos is a challenging computer vision task [24, 29, 31, 39, 45]. Common difficulties arise from frequent occlusions [43] and background clutter, as well as variations in object shape and appearance. Video object segmentation also requires accurate tracking of object boundaries over time in the presence of possibly fast and non-rigid motions. An additional challenge addressed by several recent works is in segmentation of individual instances of the same object class [18, 19, 45, 44, 51]. Indeed, while it may be easy to segment a herd of cows from a grass field, segmenting each cow separately is a much harder task.

Instance-level object segmentation in video is an interesting and understudied problem at the intersection of semantic and motion-based video segmentation. Solutions to this problem can benefit from class-specific object models and motion cues. Segmentation of static and/or partially occluded objects of the same class, however, pose additional challenges, difficult to solve with existing methods of motion-based and semantic segmentation. Meanwhile, successful solutions to instance-level video segmentation can serve in several tasks such as video editing and dynamic scene understanding.

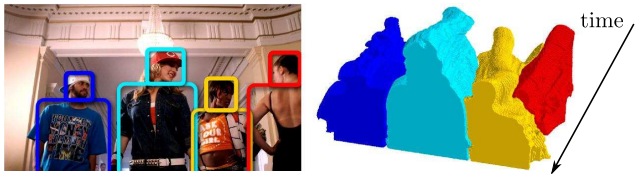

Figure 1: Results of our method applied to multi-person segmentation in a sample video from our database. Given an input video together with the tracks of object bounding boxes (left), our method finds pixel-wise segmentation for each object instance across video frames (right).

Given recent advances in object detection [36] and visual object tracking [11], coarse object localization in the form of object bounding boxes can now be used as input for solving other problems. In particular, we address in this paper the problem of instance-level video segmentation given object tracks. We assume that prior (weak) knowledge about objects is available in the form of tracked object bounding boxes, obtained by a separate process. For instance, pre-trained object detectors or visual object tracking algorithms as the ones cited above can be used.

Segmentation methods typically optimize carefully designed objective functions combining data terms and prior knowledge. Object prior knowledge in such methods is often encoded by higher-order potentials [26, 27, 38], which enable richer modeling but lead to hard optimization problems. Here we take an alternative approach and build on the discriminative clustering framework [3, 17]. Following previous work on co-segmentation [24] and weakly-supervised classification [6], we formulate our problem as a quadratic program under linear constraints. We use object tracks as constraints to guide segmentation, but other forms of prior knowledge could easily be integrated in our method. Our final segmentation is obtained by solving a convex relaxation of our objective with the Frank-Wolfe algorithm [15].

We compare our method to the state of the art and show competitive results on a new dataset for instance-level video segmentation. In contrast to most previous methods, our approach segments multiple instances of the same object class

*WILLOW project-team, Département d'Informatique de l'École Normale Supérieure, ENS/Inria/CNRS UMR 8548, Paris, France.
†SIERRA project-team, Département d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548, Paris, France.

and supports reasoning about occlusions. Figure 1 illustrates results of our method on a video from our dataset.

The contributions of this paper are three-fold. **(i)** We propose a discriminative clustering approach for instance-level video segmentation using external guidance in the form of object bounding boxes. **(ii)** We introduce a new dataset for multiple person segmentation with challenging sequences, including self-occlusions, crowded scenes and varied poses. **(iii)** We demonstrate the high accuracy and flexibility of our model on the task of multi-instance person segmentation in video.

The rest of the paper is organized as follows. We discuss related work in Section 2 and then present our problem formulation in Section 3. We describe the convex relaxation of our model and the optimization of the cost function with the Frank-Wolfe algorithm in Section 4. The new dataset is presented in Section 5. Finally, Section 6 presents our experimental setup and results.

2. Related Work

Segmentation of multiple objects in video has been addressed for example in [10, 14, 23]. Typical approaches include (a) pure color- or motion-based segmentation [28, 34, 43], for instance using long term tracks [30, 32], (b) tracking segmentation proposals through the entire video [10, 31] or (c) learning instance-specific object appearance models [14]. Deep neural networks can be used on each frame to perform high quality semantic segmentation [52] or multi-instance segmentation for a specific class and setup, for instance to segment cars in images recorded by a car-mounted camera [51].

Prior information, such as object bounding boxes or pose estimation can be used to guide the segmentation. Person and body-part detectors as well as skin color models have been used to seed the GrabCut algorithm [20], as unary potentials in CRFs [29, 37] or as higher order terms [27, 45]. Parameters of semantic segmentation models can be estimated using bounding boxes instead of exact ground-truth segmentations [33]. Pose estimation has also been used as a unary term for pixel-wise segmentation in 3D movies [38], although erroneous pose estimation will cause false segmentations. The use of object detectors as weak cues for semantic video segmentation been explored in [50]. Multiple forms of weak annotations, such as image-level tags, bounding boxes and incomplete pixel-level labels, have been combined for semantic segmentation [47]. A recently proposed task, simultaneous detection and segmentation [18], closes the gap between object detection and segmentation. In our work, bounding boxes are used to constrain the set of possible segmentations instead of being involved in the energy function.

Our formulation of the segmentation problem is related to discriminative clustering [3, 17]. The idea is to jointly partition the data and learn a discriminative model for each cluster seen as a class. Discriminative clustering has recently been applied to several problems including image co-segmentation [24, 25], object co-localization in images [42], finding actor identities in movies [6, 35] and temporal action localization [7]. Each of these techniques is built upon a task-dependent set of constraints, modeling simple assumptions and encoding prior knowledge. We use a similar framework for video segmentation and define an original set of constraints that are well suited to our problem.

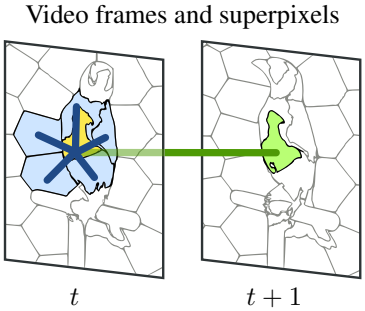

Figure 2: Spatio-temporal graph of superpixels. For the yellow superpixel, spatial edges are shown in dark blue and temporal edges in dark green.

We build upon these works and combine a grouping model, a foreground-background model and weak priors. Our proposed method is experimentally compared to several previous approaches [18, 32, 34, 38, 52] in Section 6.

3. Problem formulation

The segmentation problem we aim to solve is to assign to every pixel a label in $\{0, \dots, K\}$. To design a suitable cost function, we follow previous work on co-segmentation [24, 25]. This implies using two complementary cost functions: the first one is a spectral clustering term [39], which enforces spatial and temporal consistencies according to some descriptors ϕ. The second term is a discriminative clustering cost based on the square loss [3] which learns a foreground *vs*. background classifier. In order to include prior information, we propose several constraints which we detail in Section 3.4. The proposed constraints are linear, leading to a tractable (relaxed) optimization problem (see Section 4).

The intuition behind our approach is that constraints provide weak localization cues for each object instance. Discriminative clustering separates foreground objects from the background based on appearance features. Spectral clustering helps producing clean spatial boundaries, separating different instances of the same class and smoothing the segmentation in time for each object instance.

3.1. Notations and model

We are given a video clip composed of T frames indexed by t. Our problem is to assign a label k in $\{0, 1, \dots, K\}$ to each pixel in each frame, where label $k = 0$ corresponds to the background and all other integers in $\{1, \dots, K\}$ correspond to the K object instances in the video. Since the number of pixels in a video is usually high, we propose to work with superpixels instead. Assuming that there are N superpixels in the whole video, we index them by n in $\{1, \dots, N\}$.

Let us define a label matrix y in $\{0, 1\}^{N \times (K+1)}$. The matrix y is such that y_{nk} is equal to one if and only if the superpixel n is of label k. This matrix sums up to one along rows, since every superpixel is assigned to a single label. In Section 3.4, we propose several constraints that will restrain the set of admissible matrices y. We denote by $\mathcal{Y}$ this set. The constraints can be indexed by c in $\{1, \dots, C\}$. Since some of them may not be satisfied, for every constraint c, we define a slack variable ξ_c which will allow us to violate it. Let ξ be the concatenation of all the ξ_c into a single vector. We denote by $\mathcal{C}(y, \xi)$ the set of constraints over a specific y with slack ξ.

The cost we minimize is a sum of three terms: a grouping term E_G, a discriminative term E_D, and a term penalizing the slack ξ:

$$\min_{y \in \mathcal{Y},\ \xi \in \mathbb{R}^C_+} \quad E_G(y) + \alpha\, E_D(y) + \beta \|\xi\|^2, \qquad (1)$$

under linear constraints $\mathcal{C}(y, \xi)$, where α and β allow us to weigh the different terms. We provide a detailed description of these terms in the following sections.

3.2. Grouping term

The grouping term E_G is a classic spectral clustering term meant to ensure spatial and temporal consistency of the segmentation. To this end, we define a *superpixel graph* $G = (S, \mathcal{E})$, whose nodes correspond to superpixels and edges encode spatio-temporal neighborhood information. A sample graph G is illustrated in Fig. 2. For two nodes n and n' from the same frame, there is an edge (n, n') in $\mathcal{E}$ if the two superpixels are spatial neighbours. For node n in frame t and node n' in frame $t + 1$, we add an edge (n, n') to $\mathcal{E}$ if n and n' are temporal neighbours. The exact way we define neighbourhoods is discussed in Section 6.1.

For each superpixel n, we define a set of descriptors ϕ^i_n indexed by i in $\{1, \dots, I\}$. We denote by d_i the dimension of ϕ^i_n and by d the sum of all the d_i. Let us denote by ϕ_n the concatenation of all the ϕ^i_n. We then define the similarity matrix W in $\mathbb{R}^{N \times N}$ which encodes the similarities between superpixels: $W_{nn'} = \sum_{i=1}^{I} \mu_i \exp(-\lambda_i \|\phi^i_n - \phi^i_{n'}\|^2)$ if $(n, n') \in \mathcal{E}$ and 0 otherwise. μ_i and λ_i are weighting parameters for the i-th descriptor.

Following [39], we define the associated unnormalized Laplacian matrix $L = D - W$. D is the diagonal matrix composed of the row sums of W: $D = \mathrm{Diag}(W \mathbf{1}_N)$. Using these definitions, the grouping term can be written as the following quadratic form:

$$E_G(y) = \frac{1}{N} \mathrm{Tr}(y^T L y). \qquad (2)$$

3.3. Discriminative term

E_D is a standard discriminative clustering term, which aims to learn an affine classifier for separating foreground *vs*. background. Let M be a binary matrix in $\{0, 1\}^{(K+1) \times 2}$ which maps labels to foreground and background. Let us denote by $w \in \mathbb{R}^{d \times 2}$ and b in $\mathbb{R}^2$ the parametrization of this model. We also define the matrix Φ in $\mathbb{R}^{N \times d}$ whose rows are the ϕ_n. The discriminative cost is defined as follows:

$$E_D(y) = \min_{\substack{w \in \mathbb{R}^{d \times 2} \\ b \in \mathbb{R}^2}} \frac{1}{N} \|yM - \Phi w - 1_N b^T\|_F^2 + \kappa ||w||_F^2. \quad (3)$$

The minimization w.r.t. w in (3) is a ridge regression problem, whose solution can be found in closed form, and E_D is easily rewritten [24] as a quadratic form in y:

$$E_D(y) = \frac{1}{N} \mathrm{Tr}(M^T y^T A y M), \qquad (4)$$

where $A = \frac{1}{N} \Pi_N (I_N - \Phi(\Phi^T \Pi_N \Phi + N\kappa I_d)^{-1} \Phi^T) \Pi_N$ and Π_N is the centering projection matrix $I_N - \frac{1}{N} 1_N 1_N^T$.

Note that the foreground *vs*. background model we learn is fit for segmenting the background from multiple instances of the same object category. If needed, we could easily learn one model per object category by adapting the M matrix.

3.4. Constraints

As mentioned earlier, our model incorporates constraints on the y matrix. They allow us to encode simple priors as well as more complicated, instance-specific information. We can constrain the number of superpixels assigned to a given label in a spatio-temporal region using linear inequalities. We can also use strict equality constraints to fix the labels of some superpixels. We first provide a general form and then describe the different variants used in our experiments. Some of them are also illustrated in Fig. (3) for multi-instance person segmentation using head and full-body tracks.

Object tracks. We assume that we are given a track of bounding boxes for each object in the video. We denote by $\mathcal{B}$ this set and index the elements B of $\mathcal{B}$ by k in $\{1, \dots, K\}$ and t in $\{1, \dots, T\}$, such that B^t_k denotes the bounding box of the k-th object in frame t.

Inequality constraints. We want to impose linear inequality constraints on a set of superpixels in the video. In the following sections we will describe in details what these sets can correspond to. For now, let us denote by R a subset of the indices of superpixels, $R \subset \{1, \dots, N\}$. We can represent R by the indicator vector $\mathbb{1}_R$, such that the n-th entry is equal to one if the superpixel n is in R. Note that

(a) Head constraint. (b) Body constraint. (c) Background constraint. (d) Non-person. (e) Must-be background.

Figure 3: Constraints (see Section 3.4) used in our model for multi-person segmentation. In this setup we are provided head detections, from which we derive body boxes. We require 75% of pixels inside head detections (a) and 50% of pixels inside body boxes (b) to belong to the instance. Part (c) illustrates the background constraint (96% of this surface should be background); non-person constraints which enforce superpixels far from the person to be assigned to the corresponding label (d) ; and the superpixels which can only be background (e).

for videos, this set R can correspond to a spatio-temporal region. We use the notation $\mathbf{e}_k$ to denote the k-th vector of the canonical basis of $\mathbb{R}^{K+1}$.

For some region R and a label k, we propose to constrain the matrix y using constraints of the following form:

$$0 \geq \sigma \left(\mathbb{1}_R^T \, y \, \mathbf{e}_k - \rho \right) - \xi_c, \tag{5}$$

where $\sigma \in \{-1, 1\}$ controls whether this is an *at least* or an *at most* constraint, ρ a parameter and ξ_c is the slack variable allowing this constraint to be violated. The parameters R, σ, k and ρ depend on the kind of prior we want to enforce.

Note that while our notations refer to superpixels and counts of superpixels, in practice we weigh the contribution of each superpixel to the constraint by its relative area in region R. Likewise, we reason in terms of pixels when computing the ρ parameters.

Equality constraints. When some supervision is available (semi-supervised setting), or when a strong cue allows us to freeze variables, we want to use equality constraints. Let us suppose that we have a set of superpixels R and a set of labels Q. We set variables for region R and labels Q to predetermined values stored in $\tilde{y}$:

$$\forall r \in R, \; \forall q \in Q, \quad y_{rq} = \tilde{y}_{rq}. \tag{6}$$

As for the inequality constraints, the definitions of R, Q and $\tilde{y}$ depend on the prior.

Track constraints. Given an object bounding box B_k^t, we require that at least ρ_B superpixels inside B_k^t get assigned the label k. This can be enforced by setting R, and σ appropriately in Eq. (5). We set R to the set of superpixels that lie inside B_k^t. Since this is an *at least* constraint, we set $\sigma = -1$. The amount of superpixels ρ_B is set to a ratio of the total number of superpixels in B_k^t. In Figure 3 (a) and (b), head tracks and object tracks are used for such constraints.

In complex videos picturing multiple objects, the bounding boxes, and thus the corresponding constraint regions, can heavily overlap. Without slack variables, our problem may be infeasible in such situations, and even with slack variables the constraints may still be misleading. To cope with occlusions, we propose a simple occlusion reasoning. In a given frame, for each pair of overlapping bounding boxes, pixels inside the region of overlap are marked as occluded. In turn, we reduce the strength of each such constraint by multiplying ρ_B by $(1 - o)$ where o is the ratio of occluded pixels in the bounding box.

Area constraints. To reduce "leaking" effects in the segmentation, we constrain the area of each object segment in each frame. For object k in frame t, we impose that at most ρ_{area} of the superpixels in frame t get assigned the label k. This can be expressed by setting R to be the set of superpixels in frame t. Since this is an *at most* constraint, we have $\sigma = +1$. We set ρ_{area} to the amount of superpixels in track B_k^t times a constant, to take object size into account.

We can also enforce a minimal amount of superpixels per label and per frame. We do so by changing σ to -1 and setting an appropriate ρ. This constraint can be used if we know the object is in the frame but lack the corresponding bounding box.

Background constraints. We request that most superpixels which are outside object bounding boxes belong to the background label. The rationale is that only a few of the superpixels outside object detections may belong to objects, as shown in Figure 3 (c). Typically, in the case of multiple people segmentation, these superpixels belong to lower arms. We express this constraint by setting R to the set of superpixels that do not belong to any track in frame t. This is an *at least* constraint so we set $\sigma = -1$. We set $\rho = \rho_{\text{bg}}$ to a ratio of the cardinality of R.

Non-object constraints. In our work, we make the assumption that if a pixel is far enough from an object detection, it is reasonable to assume that it does not belong to the corresponding object. We assume that when there are no detections at all, we do not apply these constraints. For a bounding box B_k^t, we build R as the set of superpixels in frame t that are further away from B_k^t than a given distance, as shown in Figure 3 (d). In practice, we set this minimum distance to the width of the object bounding box. R can be computed by performing a distance transform and then thresholding. We then enforce an equality constraint with Q containing only the label k and $\tilde{y}$ filled with zeros.

4. Optimization

4.1. Continuous relaxation

The quadratic problem defined in Eq. (1) is known to be NP hard when y takes binary values. Indeed, when the quadratic cost matrix has positive off diagonal entries, this

is as hard as solving a max-cut problem. Classic relaxations of such problems [24] imply working with equivalence matrices $Y = yy^T$. Doing so in our case would be intractable due to the problem size and would prevent us from imposing constraints relating superpixels to labels. Instead, we propose a continuous relaxation of our problem by solving it over the convex hull $\overline{\mathcal{Y}}$ of the initial set $\mathcal{Y}$. Then, we aim at solving the minimization of a positive semi-definite quadratic form over a convex compact set defined by a large number of linear constraints. Due to the size of y (of the order of 10^6 entries) and the number of constraints it is not realistic to use a standard off-the-shelf quadratic programming solver based on interior point methods [8]. Nevertheless it is possible to solve linear programs of such a size. This is why, following other approaches to discriminative clustering [7], we propose to use the Frank-Wolfe optimization algorithm [15, 22] which only relies on the minimization of linear forms over $\overline{\mathcal{Y}}$.

4.2. Frank-Wolfe algorithm

The Frank-Wolfe algorithm is an iterative method to optimize convex objectives over compact convex sets and suites well for our problem. Let us now briefly describe the iterations. We define our optimization variable $z = (y, \xi)$ in $\mathcal{Z} = \overline{\mathcal{Y}} \times \mathbb{R}_+^C$. For the sake of simplicity, we rewrite as $E(z)$ the sum of the three terms from Eq. (1). Let us denote by z_k the current point at iteration k. At iteration k, we compute the gradient $\nabla_z E(z_k)$ and minimize the following linear form: $\mathrm{Tr}(\nabla_z E(z_k)(z - z_k))$. This can be easily done using a generic LP solver, and yields a corner of the polytope that we will denote z_{FW}. We then update the current point as follows: $z_{k+1} = z_k + \gamma(z_{\text{FW}} - z_k)$. The optimal parameter γ^* leading to the best improvement in that direction can be found in closed form by doing an exact line search.

Rounding. Using the Frank-Wolfe algorithm we obtain a solution $z^* = (y^*, \xi^*)$. The solution continuous solution we obtain needs to be rounded. We first freeze the slack variables of the constraints to the values ξ^*. We then round y^* into a binary matrix by finding the closest point to y^* in $\mathcal{Y}$ in terms of Frobenius norm $\|y - y^*\|_F^2$ which is equivalent to

$$\min_{y \in \mathcal{Y}} -2\mathrm{Tr}(y^{*T} y). \tag{7}$$

We solve this linear program using the LP solver.

4.3. Non-convex refinement

Experimentally, we observe that the convex relaxation of our problem may lead to sub-optimal rounded solutions. Indeed, our model is attracted to a degenerate solution with all constant entries of value $\frac{1}{K+1}$, which has a low objective value for the discriminative term. This is a common drawback of discriminative clustering techniques, as noted by [24, 17]. In order to push our solution away from these near-constant solution, and following the approach of graduated non-convexity [5, 49], we propose to add a concave quadratic term to our objective: $\mathrm{Tr}(y^T(1 - y))$, and weight it using a parameter δ. This term encourages the entries y to be close to either 0 or 1. The corresponding optimization problem is the following:

$$\min_{y \in \mathcal{Y},\ \xi \in {\mathbb{R}_+}^C} E_G(y) + \alpha E_D(y) + \beta \|\xi\|^2 + \delta\ \mathrm{Tr}(y^T(1 - y)).$$

The parameter δ can be a function of the iteration count k. In practice however, choosing a scalar value is already complicated and we therefore use a piecewise constant function. We first optimize the convex relaxation of our problem with $\delta = 0$. Then we perform Frank-Wolfe steps on the non-convex objective with a non-zero δ which has been selected by parameter search. Although we are only guaranteed to converge to a local optimum of this non-convex function [4, Section 2.2.2], we empirically observe a drastic improvement of performance as shown in Table 1.

5. Dataset

To evaluate the performance of our method on the task of instance-level video segmentation, we have collected a dataset composed of 27 video clips, corresponding to a total of 2476 frames. The video clips are taken from the 3D feature movie "StreetDance 3D" [Giwa and Pasquini, 2010]. The proposed dataset is an improved version of the *Inria 3D Movie Dataset* [38] adding a substantial amount of challenges, such as longer shots, self-occlusions, inter-person occlusions, and hard poses such as dancing or jumping.

Providing ground-truth annotations for evaluation in an entire video is a highly time-consuming task. As a consequence, we have only annotated a sparse subset of 235 frames out of 2476, for all 632 person instances present in these frames. We split the dataset into a set of 7 clips for adjusting hyperparameters and a set of 20 clips for evaluation. Note that there is no training step in our method, but only a validation step to find appropriate hyperparameters. Our dataset and code, including the procedure for adjusting parameters using the Bayesian framework of [41], are available on the project website [1].

6. Experiments

In this section, we describe experimental details and evaluation procedures for the proposed method. We evaluate multi-instance person segmentation in 3D movies using head tracks and full-body bounding boxes.

6.1. Implementation details

Superpixels. We extract video superpixels using [9]. The superpixels are evenly distributed, fairly compact, and tracked in time. We use temporal links obtained from superpixel tracks as edges in the superpixel graph (Section 3.2). We also add edges between superpixels from consecutive frames if sufficient pixel-wise correspondence is provided

Figure 4: Qualitative results of our method. Note that most of the visually unpleasant artifacts are due to the use of superpixels.

by optical flow.

Features. We first compute dense optical flow between consecutive frames using DeepFlow [46]. Then, we use two different sets of features ϕ_n for the grouping and discriminative terms. These features are computed for each superpixel based on the underlying image pixels. For the spatial edges in the similarity matrix W of the grouping term, we use: **(i)** a histogram of optical flow with 8 bins for orientations and one bin for no motion, and **(ii)** the average CIE L*a*b* color, over the superpixel. For the temporal edges of W, we use the average CIE L*a*b* color. As discriminative features in Φ, we use: **(i)** the same histogram of optical flow, **(ii)** a color histogram computed over RGB colors, with 8 bins per color channel, 512 bins in total, and **(iii)** the average SIFT descriptor over the superpixel, obtained by first computing dense SIFTs over the whole image, and then averaging the SIFTs which cover the superpixel.

We also optionally exploit recent advances in semantic segmentation by including features produced by a deep neural network trained for semantic segmentation for the PASCAL dataset [52]. We take the output of the method for each pixel and pool it (either using max-pooling or mean-pooling) over the superpixel, and use it as an additional discriminative feature in Φ. As this output represents a strong semantic cue, it should help our discriminative term to separate the foreground from the background.

For 3D movies, we also include median disparity over the superpixel in both spatial grouping and discriminative features. The method of Ayvaci *et al*. [2] is used to estimate the disparity map from stereo pairs.

Person detection and tracking. We evaluate our method on ground-truth (manually annotated) head tracks as well as on tracks automatically produced by a tracking-by-detection method: we use a CNN-based detector [16] trained on heads in movies. The tracker associates these detections based on KLT tracks [40], interpolates missing detections and smooths the tracks in time [12]. Using ground-truth or automatic tracks, we extrapolate full-body bounding boxes from the head bounding boxes using a linear transformation. Note that our full-body bounding boxes start below the head, as shown in Fig. 1. This way, the superpixels on the sides of the head are not involved in the corresponding constraints, since they do not belong to the person in most cases.

Table 1: Comprehensive study of the influence of each component of our method on our new dataset. See Section 6.3 for comments.

Method	F_1	Precision	Recall	Overlap
Ours + semantic cue	**80.1%**	**81.9%**	**79.6%**	**68.6%**
Ours	78.3%	80.8%	77.3%	66.0%
No temporal smoothness	76.4%	79.2%	75.4%	63.7%
Single frames	76.4%	77.9%	76.4%	63.7%
Grouping term only	77.6%	79.4%	77.2%	65.0%
Discriminative term only	66.9%	70.7%	64.7%	52.1%
No constraint	12.8%	10.4%	40.0%	09.0%
Convex only	75.6%	78.0%	74.1%	62.4%
No disparity	74.0%	77.5%	72.6%	59.9%

Occlusion reasoning. We adapt the occlusion reasoning of Section 3.4 to stereo videos by computing a depth estimate from the median disparity inside the head box. Given two overlapping bounding boxes in the frame, we mark the pixels of the object which is behind as occluded. This procedure allows a more accurate handling of occlusions than the original reasoning, since constraint strength will only be reduced for objects which may actually be occluded.

We evaluate the proposed method on stereo videos where head (bounding boxes) tracks for multiple people are given as input to our algorithm. We use these tracks and extrapolated full-body bounding boxes, to derive two types of *track constraints* in our framework. We also integrate the corresponding *background* and *non-object* constraints from Section 3.4. We combine disparity, appearance and motion cues and evaluate performance on a new dataset extracted from 3D movies with challenging scenes and poses.

6.2. Baselines

We compare our method to multiple baselines, spanning the whole range of methods from pure semantic segmentation to pure motion segmentation. Some of them are completely unsupervised: *Multi-modal motion segm.* [32],

FG/BG motion segm. [34]. Some other require pixel-wise supervision to train appearance models: *Pose & segm.* [38], *SDS* [18], *CRF as RNN* [52]. We used the publicly available code and models for all methods.

CRF as RNN [52][1] is the state-of-the-art semantic segmentation method. It uses an end-to-end deep network combining a standard Convolutional Neural Network with a Recurrent Neural Network to perform dense CRF inference. We adapt this method to the task of instance-level segmentation for a given semantic class by assigning each pixel labelled with the said semantic class to the instance which has the closest bounding box. In practice, for humans we assign the pixels to the person which spine (derived from the head bounding box) is the closest.

SDS [18][2] is a simultaneous detection and segmentation method. It classifies region proposals by scoring CNN features extracted from the region and the corresponding bounding box. [18] is an instance-level segmentation method, and we evaluate it directly. Since [18] uses its own set of detections, we use the same set of detection within our method when comparing results with *SDS*.

Pose & segm. [38][3] is based on multi-class graph cuts, has been designed for a similar dataset, and uses the same set of features. Given person tracks, it combines pose estimates and disparity cues in an unary term after reasoning on occlusions. A binary term encodes spatio-temporal smoothness using color and motion features.

Multi-modal motion segm. [32][4] separates objects which exhibit different motions. It is a classic method for video segmentation. We adapt it to our problem by assigning the biggest segment (in terms of surface) to be the background segment, and inside each object bounding box we label the largest non-background segment as belonging to the instance.

FG/BG motion segm [34][5] is a pure figure-ground motion segmentation method. We adapt it to the task of instance-level segmentation using the same method as for the first baseline, by splitting the foreground segment in multiple segments.

6.3. Results

We evaluate segmentation by computing per-person precision, recall, overlap (defined as the intersection over union between the ground-truth and predicted labels [13, 21]) and F_1 score (the harmonic mean between precision and recall). We report the average of these measures over people and frames. We show qualitative results of our method in Figure 4 and video results on the project website [1].

Comprehensive analysis. We first analyze each component of our method in Table 1. It is interesting to note that similar results are achieved when removing temporal edges from the graph (*No temporal smoothness*), or when processing frames one by one (*Single frames*). Experiments on single frames have a higher recall, while segmenting all frames at once without temporal smoothness produces higher precision, showing the influence of the discriminative term when it has access to the whole video context. Results obtained using the *Grouping term only* are quite good, whereas using the *Discriminative term only* has a lower performance since it only models foreground *vs*. background segmentation without any spatial or temporal consistency. Still, combining the two terms (*Ours*) leads to the best performance as the discriminative term helps to improve precision. Performance is pushed even further when the discriminative term contains strong semantic cues (*Ours + semantic cue*). The non-convex refinement from Section 4.3 used in *Full method* produces significantly better performance than using *Convex only* optimization. As discussed in [3, 6], using *No constraint* leads to trivial solutions and very poor results. Last, even without disparity features (*No disparity*), which are strong cues, our method produces decent results.

Table 2: Quantitative performance comparison of our method with 5 baselines. Please note that the results from *Ground truth tracks*, *Automatic tracks* and *SDS detections* sections are not comparable as they use different sets of detections. See Section 6.3 for comments.

Method	F_1	Precision	Recall	Overlap
Ground truth tracks:				
Ours	78.3%	80.8%	77.3%	66.0%
Ours (+ semantic cue)	**80.1%**	81.9%	79.6%	**68.6%**
CRF as RNN [52]	78.5%	**83.2%**	77.7%	66.5%
Pose & segm. [38]	68.5%	68.3%	76.1%	55.0%
Multi-modal motion segm. [32]	27.4%	41.0%	30.4%	19.4%
FB/BG motion segm. [34]	52.2%	65.1%	49.8%	38.8%
Automatic tracks:				
Ours	63.6%	61.6%	68.6%	52.0%
CRF as RNN [52]	56.2%	58.2%	54.9%	46.5%
Pose & segm. [38]	52.7%	57.2%	59.5%	40.8%
Multi-modal motion segm. [32]	27.4%	40.6%	30.4%	19.4%
FB/BG motion segm. [34]	48.4%	57.6%	50.7%	34.9%
SDS detections:				
Ours	72.5%	68.4%	**80.8%**	59.3%
SDS [18]	65.1%	73.5%	62.8%	52.6%

Baselines comparison. Quantitative and qualitative comparisons between our method and baselines are shown in Table 2 and Figure 5.

The motion segmentation baselines *Multi-modal motion segm.* and *FB/BG motion segm.* perform poorly on this challenging dataset. Both methods completely miss non-moving and almost non-moving person by nature. *Multi-modal motion segm.* also tends to separate the different limbs of a single person into multiple segments.

The *SDS* method performs fairly well. Its detection performance is better than the automatic detector we used (on some key sequences *SDS* detects twice more people than our detector), but it still misses a significant part of person

[1] http://www.robots.ox.ac.uk/~szheng/CRFasRNN.html
[2] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sds/
[3] http://www.di.ens.fr/willow/research/stereoseg/
[4] http://lmb.informatik.uni-freiburg.de/resources/software.php
[5] http://groups.inf.ed.ac.uk/calvin/FastVideoSegmentation/

(a) Ours (b) CRF as RNN

(c) FB/BG motion segm. (d) Pose & segm.

(e) SDS (f) Multi-modal motion segm.

Figure 5: Qualitative comparison between our method and the five baselines. Note that *Pose & segm.* may drop detections if the pose estimator fails, and that *SDS* is producing both detection and segmentation, so it uses its own set of detections. See Section 6.3 for comments.

instances. For instance, it misses most heavily occluded persons. The other main downside is that the method mostly provides upper body segmentations (due to either the region proposals or the classifier itself which has been trained on a mix of face, upper body and full body examples), in spite of the refinement procedure which is applied at the end of their method and is meant to provide more complete segmentations.

The *CRF as RNN* method is the best performing baseline. It produces a clean figure-ground segmentation for a given object class. When people are separated in the image, our relabelling procedure inherently produces good instance-level segmentation results. However, when the person instances are close by or overlap, our method often outperforms the baseline. Our method, which uses only generic features (color, motion, SIFT) and ad-hoc constraints, still performs as well as this strong baseline. It successfully segments each object instance with only coarse localization cues (encoded in the constraints) and without training a pixel-level appearance model for the segmentation as does the baseline. Moreover, when using semantic features of the baseline in our discriminative term, our method outperforms the baseline.

Pose & segm., which uses instance-specific pose masks, performs significantly worse than our method as it makes strong assumptions about the pose or disparity priors. For instance, it can not recover from errors from the pose estimator. In comparison, our constraints only restrict the space of possible segmentations. They can even be violated in

Figure 6: Results of our method applied to two multi-instance videos from SegTrack v2 [31].

situations which do not satisfy the implicit priors they are enforcing. However, they are strong enough to successfully guide the segmentation even for complicated poses, crowded scenes and cluttered backgrounds. We provide per-video quantitative results on the project website [1].

Other object classes. The major strength of our method is that it is mostly agnostic to the underlying object class. We provide the method with a single floating point parameter specifying which amount of each bounding box is expected to belong to the object. With this single parameter, the video input and the corresponding bounding box tracks, our method is able to properly segment the object instance from the background of the video and from the other object instances. To the best of our knowledge, there is no proper complete dataset for instance-level segmentation in videos for the moment. To show that our method can handle non-person object classes, we ran it on two videos with multiple object instances from the popular SegTrack v2 dataset [31]. We show two sample frames in Figure 6 and videos on the project website [1]. Quantitatively, our method achieves an average overlap of 86.8% on the *drifting car* sequence, compared to 79.9% by the recent state-of-the-art method [48] which combines the analysis of appearance and motion cues with a reasoning on disocclusions.

7. Discussion and future work

We have presented a flexible and effective framework for multi-instance object segmentation. We have demonstrated its experimental performance on a challenging dataset, showing that constraining the space of segmentations is a robust way to incorporate object tracks information. We plan to extend this method to multiple instances of multiple object categories. This implies having a multi-class discriminative model instead of a foreground *vs*. background one. More class- or instance-specific knowledge can be incorporated in our constraints. This includes weighing our constraints using noisy pixel-level information such as pose masks. Also, more complex models – including non-convex costs – could use our convex relaxation as an initialization. These refinements could lead to improved segmentation quality.

Acknowledgements. The authors would like to thank Jean Ponce for helpful suggestions. This work is partly supported by the MSR-INRIA laboratory and ERC grants Activia and VideoWorld.

References

[1] http://www.di.ens.fr/willow/research/instancelevel/, 2016.

[2] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *IJCV*, 2012.

[3] F. Bach and Z. Harchaoui. Diffrac: a discriminative and flexible framework for clustering. In *NIPS*, 2007.

[4] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[5] A. Blake and A. Zisserman. *Visual reconstruction*. MIT press Cambridge, 1987.

[6] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding actors and actions in movies. In *ICCV*, 2013.

[7] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014.

[8] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[9] J. Chang, D. Wei, and J. W. Fisher III. A video representation using temporal superpixels. In *CVPR*, 2013.

[10] A. Colombari, A. Fusiello, and V. Murino. Segmentation and tracking of multiple video objects. *Pattern Recognition*, 2007.

[11] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[12] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy" – automatic naming of characters in tv video. In *BMVC*, 2006.

[13] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[14] A. Fathi, M.-F. Balcan, X. Ren, and J. M. Rehg. Combining self training and active learning for video segmentation. In *BMVC*, 2011.

[15] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 1956.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[17] Y. Guo and D. Schuurmans. Convex relaxations of latent variable training. In *NIPS*, 2007.

[18] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[19] X. He and S. Gould. Multi-instance object segmentation with exemplars. In *ICCV Workshop*, 2013.

[20] A. Hernández-Vela, M. Reyes, V. Ponce, and S. Escalera. Grabcut-based human segmentation in video sequences. *Sensors*, 2012.

[21] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 1912.

[22] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013.

[23] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014.

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

[25] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012.

[26] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical CRFs for object class image segmentation. In *ICCV*, 2009.

[27] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and CRFs. In *ECCV*, 2010.

[28] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.

[29] V. S. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009.

[30] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*, 2011.

[31] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.

[32] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 2014.

[33] G. Papandreou, L. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. In *ICCV*, 2015.

[34] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.

[35] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. Linking people with "their" names using coreference resolution. In *ECCV*, 2014.

[36] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[37] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[38] G. Seguin, K. Alahari, J. Sivic, and I. Laptev. Pose estimation and segmentation of people in 3d movies. *PAMI*, 2014.

[39] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR*, 1997.

[40] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[41] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *NIPS*, 2012.

[42] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014.

[43] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, 2015.

[44] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, 2014.

[45] V. Vineet, J. Warrell, L. Ladicky, and P. Torr. Human instance segmentation from video using detector-based conditional random fields. In *BMVC*, 2011.

[46] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *ICCV*, 2013.

[47] J. Xu, A. G. Schwing, and R. Urtasun. Learning to segment under various forms of weak supervision. In *CVPR*, 2015.

[48] Y. Yang, G. Sundaramoorthi, and S. Soatto. Self-occlusions and disocclusions in causal video object segmentation. In *ICCV*, 2015.

[49] M. Zaslavskiy, F. Bach, and J.-P. Vert. A path following algorithm for the graph matching problem. *PAMI*, 2009.

[50] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia. Semantic object segmentation via detection in weakly labeled video. In *CVPR*, 2015.

[51] Z. Zhang, A. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *ICCV*, 2015.

[52] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.