Unsupervised object segmentation in video by efficient selection of highly probable positive features

Emanuela Haller[1,2] and Marius Leordeanu[1,2]

[1]University Politehnica of Bucharest, Romania
[2]Institute of Mathematics of the Romanian Academy, Romania
haller.emanuela@gmail.com marius.leordeanu@imar.ro

Abstract

We address an essential problem in computer vision, that of unsupervised foreground object segmentation in video, where a main object of interest in a video sequence should be automatically separated from its background. An efficient solution to this task would enable large-scale video interpretation at a high semantic level in the absence of the costly manual labeling. We propose an efficient unsupervised method for generating foreground object soft masks based on automatic selection and learning from highly probable positive features. We show that such features can be selected efficiently by taking into consideration the spatio-temporal appearance and motion consistency of the object in the video sequence. We also emphasize the role of the contrasting properties between the foreground object and its background. Our model is created over several stages: we start from pixel level analysis and move to descriptors that consider information over groups of pixels combined with efficient motion analysis. We also prove theoretical properties of our unsupervised learning method, which under some mild constraints is guaranteed to learn the correct classifier even in the unsupervised case. We achieve competitive and even state of the art results on the challenging Youtube-Objects and SegTrack datasets, while being at least one order of magnitude faster than the competition. We believe that the strong performance of our method, along with its theoretical properties, constitute a solid step towards solving unsupervised discovery in video.

1. Introduction

Unsupervised learning in video is a very challenging task in computer vision. Fully solving this problem would shed new light on our understanding of intelligence from a scientific perspective. It would also have a strong impact in many real-world applications, as large datasets of unlabeled videos could be collected at a relatively low cost. There are several different published approaches for unsupervised learning and discovery of the salient object in video [20, 12, 17, 16], but most have a high computational cost. In general, algorithms for unsupervised mining and clustering are expected to be computationally expensive due to the inherent combinatorial nature of the problem [7].

In this paper we address the computational cost challenge and propose a method that is both accurate and fast. We achieve our goal based on a key insight: we focus on selecting and learning from features that are highly correlated with the presence of the object of interest and can be rapidly selected and computed. **Note:** in this paper, when referring to highly probable positive features, we use "feature" to indicate a feature vector sample, not a feature type. While we do not require these features to cover all instances and parts of the object of interest (we could expect low recall), we show that it is possible to find, in the unsupervised case, positive features with high precision (a large number of those selected are indeed true positives). Then we prove theoretically that we can reliably train an object classifier using sets of positive and negative samples, both selected in an unsupervised way, as long as the set of features considered to be positive has high precision, regardless of the recall, if certain conditions are met (and they are often met in practice). We present an algorithm that can effectively and rapidly achieve this task in practice, in an unsupervised way, with state-of-the art results in difficult experiments, while being at least 10x faster than its competition. The proposed method outputs both the soft-segmentation of the main object of interest as well as its bounding box. Two examples are shown in Figure 1.

While we do not make any assumption about the type of object present in the video, we do expect the sequence to contain a single salient object, as our method performs foreground soft-segmentation and doesn't expect videos with no

Figure 1. Qualitative results of our method, which provides the soft-segmentation of the main object of interest and its bounding box.

salient object or with multiple objects of interest. The key insights that led to our formulation and algorithm are the following:

1) First, the foreground and background are complementary and in contrast to each other - they have different sizes, appearance and movements. We observed that the more we can take advantage of these contrasting properties the better the results, in practice. While the background occupies most of the image, the foreground is usually small and has distinct color and movement patterns - it stands out against its background scene.

2) The second main idea of our approach is that we should use this foreground-background complementarity in order to automatically select, with high precision, foreground features, even if the expected recall is low. Then, we could reliably use those samples as positives, and the rest as negatives, to train a classifier for detecting the main object of interest. We present this formally in Sec. 2.2.

These insights lead to our two main contributions in this paper: first, we show theoretically that by selecting features that are positive with high probability, a robust classifier for foreground regions can be learned. Second, we present an efficient method based on this insight, which in practice outperforms its competition on many different object classes, while being 10x faster.

Related work on object discovery in video: The task of object discovery in video has been tackled for many years, with early approaches being based on local features matching [20, 12]. Current literature offers a wide range of solutions, with varying degrees of supervision, going from fully unsupervised methods [17, 16] to partially supervised ones [10, 25, 24, 11, 21] - which start from region, object or segmentation proposals estimated by systems trained in a supervised manner [1, 4, 3]. Some methods also require user input for the first frame of the video [8]. Most object discovery approaches that produce a fine shape segmentation of the object also make use of off-the-shelf shape segmentation methods [19, 5, 14, 2, 15].

2. Approach

Our method receives as input a video sequence, in which there is a main object of interest, and it outputs its soft-segmentation masks and associated bounding boxes. The proposed approach has, as starting point, a processing stage based on principal component analysis of the video frames, which provides an initial soft-segmentation of the object - similar to the recent VideoPCA algorithm introduced as part of the object discovery approach of [21]. This soft-segmentation usually has high precision but may have low recall. Starting from this initial stage that classifies pixels independently based only on their individual color, next we learn a higher level descriptor that considers groups of pixel colors and is able to capture higher order statistics about the object properties, such as different color patterns and textures. During the last stage we combine the soft-segmentation based on appearance with foreground cues computed from the contrasting motion of the main object vs. its scene. The resulting method is accurate and fast ($\approx$ 3 fps in Matlab, 2.60GHz CPU - see Sec. 3.3). Our code is available online[1].

Below, we summarize the steps of our approach (also see Figure 2), in relation with Algorithm 1 (the pseudocode of our approach).

- **Step 1**: select highly probable foreground pixels based on the differences between the original frames and the frames projected on their subspace with principal component analysis (Sec. 2.1, Alg. 1 - lines [2, 5]).
- **Step 2**: estimate empirical color distributions for foreground and background from the pixel masks computed at Step 1. Use these distributions to estimate the probability of foreground for each pixel independently based on its color (Sec. 2.1.1, Alg. 1 - line 6).
- **Step 3**: improve the soft-segmentation from Step 2, by projection on the subspace of soft-segmentations (Sec. 2.3, Alg. 1 - lines [7, 9]).

[1] https://goo.gl/2aYt4s

- **Step 4**: re-estimate empirical color distributions for foreground and background from the pixel masks updated at Step 3. Use these distributions to estimate the probability of foreground for each pixel independently based on its color (Sec. 2.1.1, Alg. 1 - line 10).
- **Step 5**: learn a discriminative classifier of foreground regions with regularized least squares regression on the soft segmentation real output $\in [0, 1]$. Use a feature vector that considers groups of colors that co-occur in larger patches. Run classifier at each pixel location in the video and produce improved per frame foreground soft-segmentation (Sec. 2.4, Alg. 1 - lines [11, 15]).
- **Step 6**: combine soft-segmentation using appearance (Step 5) with foreground motion cues efficiently computed by modeling the background motion. Obtain the final soft-segmentation (Sec. 2.5, Alg. 1 - lines [16, 23]).
- **Step 7**: Optional: refine segmentation using GrabCut [19], by considering as potential foreground and background samples the pixels given by the soft-segmentation from Step 6 (Sec. 2.5).

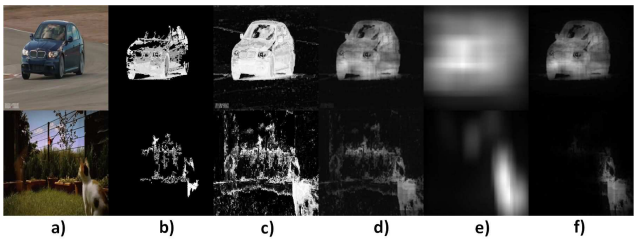

Figure 2. Algorithm overview. **a)** original image **b)** first pixel-level appearance model, based on initial object cues (Step 1 & Step 2) **c)** refined pixel-level appearance model, built from the projection of soft-segmentation (Step 3 & Step 4) **d)** patch-level appearance model (Step 5) **e)** motion estimation mask (part of Step 6) **f)** final soft-segmentation mask (Step 6).

We reiterate: our algorithm has at its core two main ideas. The first is that the object and the background have contrasting properties in terms of size, appearance and movement. This insight leads to the ability of reliably selecting a few regions in the video that are highly likely to belong to the object. The following, second idea, which brings certain formal guarantees, is that if we are able to select, in an unsupervised manner, even a small portion of the foreground object, but with high precision, then, under some reasonable assumptions, we could train a robust foreground-background classifier that can be used for the automatic discovery of the object. In Table 1 we present the improvements in precision, recall and F-measure between the different steps of our algorithm. Note that the arrows go from the precision and recall of the samples initially considered to be positive, to the precision and recall of the pixels finally classified as positive. The significant improvement

	Step 1&2	Step 3&4	Step 5
precision	$66 \rightarrow 70$	$62 \rightarrow 60$	$64 \rightarrow 74$
recall	$17 \rightarrow 51$	$45 \rightarrow 60$	$58 \rightarrow 68$
F-measure	$27 \rightarrow 59$	$53 \rightarrow 60$	$61 \rightarrow 72$

Table 1. Evolution of precision, recall and F-measure of the feature samples considered as positives (foreground) at different stages of our method (SegTrack dataset). We start with a corrupted set of positive samples with high precision and low recall, and improve both precision and recall through the stages of our method. Thus the soft masks become more and more accurate from one stage to the next.

	Step 1&2	Step 3&4	Step 5	Step 6
F-meas. (SegTrack)	59.0	60.0	72.0	74.6
F-meas. (YTO)	53.6	54.5	58.8	63.4
Runtime (sec/frame)	0.05	0.03	0.25	0.02

Table 2. Performance analysis and execution time for all stages of our method.

in F-measure is explained by our theoretical result (stated in Proposition 1), which shows that under certain conditions, a reliable classifier will be learned even if the recall of the corrupted positive samples is low, as long as the precision is relatively high. In Table 2 we introduce quantitative results of the different stages of our method, along with the associated execution times.

Algorithm 1 Video object segmentation

get input frames $\mathbf{F}^i$

PCA($\mathbf{A}_1$) => $\mathbf{V}_1$ eigenvectors; $\mathbf{A}_1(i,:) = \mathbf{F}^i(:)$
$\mathbf{R}_1 = \bar{\mathbf{A}}_1 + (\mathbf{A}_1 - \bar{\mathbf{A}}_1) * \mathbf{V}_1 * \mathbf{V}_1^T$ - reconstruction
${\mathbf{P}_1}^i = \mathbf{d}(\mathbf{A}_1(i,:), \mathbf{R}_1(i,:))$
${\mathbf{P}_1}^i = {\mathbf{P}_1}^i \otimes \mathbf{G}_{\sigma_1}$

${\mathbf{P}_1}^i$ => pixel-level appearance model => ${\mathbf{S}_1}^i$

PCA($\mathbf{A}_2$) => $\mathbf{V}_2$ eigenvectors; $\mathbf{A}_2(i,:) = {\mathbf{S}_1}^i(:)$
$\mathbf{R}_2 = \bar{\mathbf{A}}_2 + (\mathbf{A}_2 - \bar{\mathbf{A}}_2) * \mathbf{V}_2 * \mathbf{V}_2^T$ - reconstruction
${\mathbf{P}_2}^i = {\mathbf{R}_2}^i \otimes \mathbf{G}_{\sigma_2}$

${\mathbf{P}_2}^i$ => pixel-level appearance model => ${\mathbf{S}_2}^i$

$\mathbf{D}$ - data matrix containing patch-level descriptors
$\mathbf{s}$ patch labels extracted from ${\mathbf{S}_2}^i$
select k features from $\mathbf{D}$ => $\mathbf{D_s}$
$\mathbf{w} = (\lambda\mathbf{I} + {\mathbf{D}_s}^T\mathbf{D}_s)^{-1}{\mathbf{D}_s}^T\mathbf{s}$
evaluate => patch-level appearance model => ${\mathbf{S}_3}^i$

for each frame i **do**
    compute $\mathbf{I}_x$, $\mathbf{I}_y$ and $\mathbf{I}_t$
    build motion matrix $\mathbf{D}_m$
    $\mathbf{w}_m = (\mathbf{D_m}^T\mathbf{D_m})^{-1}{\mathbf{D}_m}^T\mathbf{I}_t$
    compute motion model $\mathbf{M}^i$
    $\mathbf{M}^i = \mathbf{M}^i \otimes {\mathbf{G}_\sigma}^i$
    combine ${\mathbf{S}_3}^i$ and $\mathbf{M}^i$ => ${\mathbf{S}_4}^i$
end for

2.1. Select highly probable object regions

We estimate the initial foreground regions by Principal Component Analysis, an approach similar to the recent method for soft foreground segmentation, VideoPCA [21]. Other approaches for soft foreground discovery could have been applied here, such as [26, 6, 9], but we have found the direction using PCA to be both fast and reliable and to fit perfectly with the later stages of our method. The principal components will represent a linear subspace of the background, as the object is expected to be an outlier, not obeying the principal variation observed in the video, thus harder to reconstruct. At this step, we project the frames on the resulted subspace and compute reconstruction error images as differences between original frames and their PCA reconstructed counter parts. If principal components are $\mathbf{u}_i,\ i \in [0 \dots n_u]$ (we used $n_u = 3$) and frame $\mathbf{f}$ projected on the subspace is $\mathbf{f}_r \approx \mathbf{f}_0 + \sum_{i=1}^{n_u}((\mathbf{f}-\mathbf{f}_0)^\top \mathbf{u}_i)\mathbf{u}_i$, where $\mathbf{f}_0$ is the average frame, then we compute the error image $\mathbf{f}_{diff} = |\mathbf{f} - \mathbf{f}_r|$. High value pixels in the error image are more likely to belong to foreground. If we further smooth these regions with a large enough Gaussian and multiply the resulting smoothed difference with another large centered Gaussian (which favors objects in the center of the image), we obtain soft foreground masks that have high precision (most pixels on these masks indeed belong to true foreground), even though they often have low recall (only a small fraction of all object pixels are selected). As discussed, high precision and low recall is all we need at this stage (see Table 1)

2.1.1 Initial soft-segmentation

Considering the small fraction of the object regions obtained at the previous step, the initial whole object soft segmentation is computed by capturing foreground and background color distributions, followed by an independent pixel-wise classification. Let $p(c|fg)$ and $p(c|bg)$ be the true foreground (fg) and background (bg) probabilities for a given color c. Using Bayes' formula with equal priors, we compute the probability of foreground for a given pixel, with an associated color c, as $p(fg|c) = \frac{p(c|fg)}{p(c|fg)+p(c|bg)}$. The foreground color likelihood is computed as $p(c|fg) = \frac{n(c,fg)}{n(c)}$, where $n(c, fg)$ is the number of considered foreground pixels having color c and $n(c)$ is the total number of pixels having color c. The background color likelihood is computed in a similar manner. Note that when computing the color likelihoods, we take into consideration information gathered from the whole movie, obtaining a robust model. The initial soft segmentation produced here is not optimal but it is computed fast (20 fps) and of sufficient quality to ensure the good performance of the subsequent stages. The first two steps of the method follow the algorithm VideoPCA first proposed in [21]. In Sec. 2.2 we present and prove our main theoretical result (Proposition 1), which explains in large part why our approach is able to produce accurate object segmentation in an unsupervised way.

2.2. Learning with HPP features

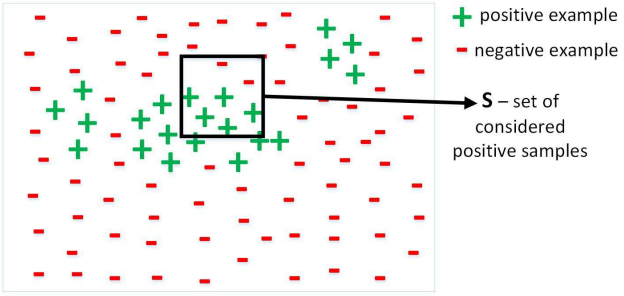

Figure 3. Learning with HPP feature vectors. Essentially, Proposition 1 shows that we could learn a reliable discriminative classifier from a small set of corrupted positive samples, with the rest being considered negatives, if the corrupted positive set contains mostly good features such that the ratio of true positives in the corrupted positive set is greater than the overall ratio of true positives. This assumption can often be met in practice and efficiently used for unsupervised learning.

In Proposition 1 we show that a classifier trained on corrupted sets of positive and negative samples, can learn the right thing as if true positives and negatives were used for training, if the following condition is met: the set of corrupted positives should contain positive samples in a proportion that is greater than the overall proportion of true positives in the whole training set. This proposition is the basis for both stages of our method, the one that classifies pixels independently based on their colors and the second in which we consider higher order color statistics among groups of pixels.

Let us start with the example in Figure 3, where we have selected a set of samples S (inside the box) as being positive. The set S has high precision (most samples are indeed positive), but low recall (most true positives are wrongly labeled). Next we show that the sets S and $\neg S$ could be used reliably (as defined in Proposition 1, below) to train a binary classifier.

Let $p(E_+)$ and $p(E_-)$ be the true distributions of positive and negative elements, and $p(\mathbf{x}|S)$ and $p(\mathbf{x}|\neg S)$ be the probabilities of observing a sample inside and outside the considered positive set S and negative set $\neg S$, respectively.

Proposition 1 (learning from highly probable positive (HPP) features): Considering the following hypotheses $\mathbf{H_1}$: $p(E_+) < q < p(E_-)$, $\mathbf{H_2}$: $p(E_+|S) > q > p(E_-|S)$, where $q \in (0, 1)$, and $\mathbf{H_3}$:

$p(\mathbf{x}|E_+)$ and $p(\mathbf{x}|E_-)$ are independent of S, then, for any sample $\mathbf{x}$ we have: $p(\mathbf{x}|S) > p(\mathbf{x}|\neg S) <=> p(\mathbf{x}|E_+) > p(\mathbf{x}|E_-)$. In other words, a classifier that classifies pixels based on their likelihoods w.r.t to S and $\neg S$ will take the same decision as if it was trained on the true positives and negatives, and we refer to it as a *reliable* classifier.

Proof: We express $p(E_-)$ as $\frac{(p(E_-)-p(E_-|S)\cdot p(S))}{(1-p(S))}$ (Eq 1), using the hypothesis and the sum rule of probabilities. Considering (Eq 1), hypothesis $\mathbf{H_1}$, $\mathbf{H_2}$, and the fact that $p(S) > 0$, we obtain that $p(E_-|\neg S) > q$ (Eq 2). In a similar fashion, $p(E_+|\neg S) < q$ (Eq 3). The previously inferred relations (Eq 2 and Eq 3) generate $p(E_-|\neg S) > q > p(E_+|\neg S)$ (Eq 4), which along with hypothesis $\mathbf{H_2}$ help as conclude that $p(E_+|S) > p(E_+|\neg S)$ (Eq 5). Also, from $\mathbf{H_3}$, we infer that $p(\mathbf{x}|E_+, S) = p(\mathbf{x}|E_+)$ and $p(\mathbf{x}|E_-, S) = p(\mathbf{x}|E_-)$ (Eq 6). Using the sum rule and hypothesis $\mathbf{H_3}$, we obtain that $p(\mathbf{x}|S) = p(E_+|S) \cdot (p(\mathbf{x}|E_+)-p(\mathbf{x}|E_-))+p(\mathbf{x}|E_-)$ (Eq 7). In a similar way, it results that $p(\mathbf{x}|\neg S) = p(E_+|\neg S)\cdot(p(\mathbf{x}|E_+)-p(\mathbf{x}|E_-))+ p(\mathbf{x}|E_-)$ (Eq 8).

$p(\mathbf{x}|S) > p(\mathbf{x}|\neg S) => p(\mathbf{x}|E_+) > p(\mathbf{x}|E_-)$: using the hypothesis and previously inferred results (Eq 5, 7 and 8) it results that $p(\mathbf{x}|E_+) > p(\mathbf{x}|E_-)$.

$p(\mathbf{x}|E_+) > p(\mathbf{x}|E_-) => p(\mathbf{x}|S) > p(\mathbf{x}|\neg S)$: from the hypothesis we can infer that $p(\mathbf{x}|E_+) - p(\mathbf{x}|E_-) > 0$, and using (Eq 5) we obtain $p(\mathbf{x}|S) > p(\mathbf{x}|\neg S)$. □

2.3. Object proposals refinement

During this stage, the soft segmentations obtained so far are improved using a projection on their PCA subspace. In contrast to 2.1, now we select the probable object regions as the PCA projected versions of the soft segmentations computed in previous steps. For the projection we consider the first 8 principal components, with the purpose of reducing the amount of noise that might be leftover from the previous steps. Further, color likelihoods are re-estimated to obtain the soft-segmentation masks.

2.4. Considering color co-occurrences

The foreground masks obtained so far were computed by treating each pixel independently, which results in masks that are not always correct, as first-order statistics, such as colors of individual pixels, cannot capture more global characteristics about object texture and shape. At this step we move to the next level of abstraction by considering groups of colors present in local patches, which are sufficiently large to capture object texture and local shape. We define a patch descriptor based on local color occurrences, as an indicator vector $\mathbf{d}_W$ over a given patch window W, such that $\mathbf{d}_W(c) = 1$ if color c is present in window W and 0 otherwise (Figure 4). Colors are indexed according to their values in HSV space, where channels H, S and V are discretized in ranges $[1, 15]$, $[1, 11]$ and $[1, 7]$, generating a total of 1155 possible colors. The descriptor does not take in consideration the exact spatial location of a given color in the patch, nor its frequency. It only accounts for the presence of c in the patch. This leads to invariance to most rigid or non-rigid transformations, while preserving the local appearance characteristics of the object. Then, we take a classification approach and learn a classifier (using regularized least squares regression, due to its considerable speed and efficiency) to separate between highly probable positive (HPP) descriptors and the rest, collected from the whole video according to the soft masks computed at the previous step. The classifier is generally robust to changes in viewpoint, scale, illumination, and other noises, while remaining discriminative (Figure 2).

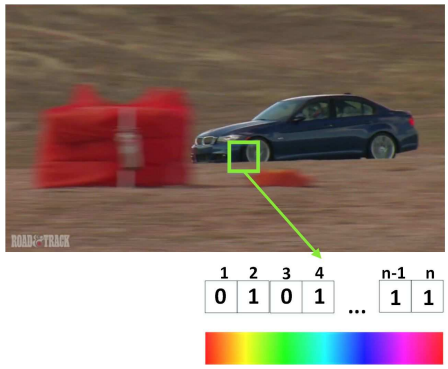

Figure 4. Initial patch descriptors encoding color occurrences (n number of considered colors).

Unsupervised descriptor learning: Not all 1155 colors are relevant for our classification problem. Most object textures are composed of only a few important colors that distinguish them against the background scene. Effectively reducing the number of colors in the descriptor and selecting only the relevant ones can improve both speed and performance. We use the efficient selection algorithm presented in [13]. The method proceeds as follows. Let n be the total number of colors and $k < n$ the number of relevant colors we want to select. The idea is to identify the group of k colors with the largest amount of covariance - they will be the ones most likely to select well the foreground versus the background (see [13] for details). Now consider $\mathbf{C}$ the covariance matrix of the colors forming the rows in the data matrix $\mathbf{D}$. The task is to solve the following optimization problem:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}}\, \mathbf{w}^T\mathbf{C}\mathbf{w}$$
$$s.t. \sum_{i=1}^{n} w_i = 1, w_i \in [0, \frac{1}{k}] \quad (1)$$

The non-zero elements of $\mathbf{w}^*$ correspond to the colors we need to select for creating our descriptor used by the classifier (based on regularized least squares regression

model), so we define a binary mask $\mathbf{w}_s \in \mathbb{R}^{n\times 1}$ over the colors (that is the descriptor vector) as follows:

$$\mathbf{w_s}(i) = \begin{cases} 1 & \text{if } \mathbf{w}^*(i) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The problem above is NP-hard, but a good approximation can be efficiently found by the method presented in [13], based on a convergent series of integer projections on the space of valid solutions. The optimal number of selected colors is a relatively small fraction of the total number, as expected. Besides the slight increase in performance, the real gain is in the significant decrease in computation time (see Figure 5).

Next we define $\mathbf{D}_s \in \mathbb{R}^{m\times(1+k)}$ to be the data matrix, with a training sample per row, after applying the selection mask to the descriptor; m is the number of training samples and k is the number of colors selected to form the descriptor (we add a constant column of 1's for the bias term). Then, the weights $\mathbf{w} \in \mathbb{R}^{(1+k)\times 1}$ of the regularized regression model are learned very fast, in closed-form:

$$\mathbf{w} = (\lambda \mathbf{I} + {\mathbf{D}_s}^T \mathbf{D}_s)^{-1} {\mathbf{D}_s}^T \mathbf{s} \quad (3)$$

where $\mathbf{I}$ is the identity matrix, λ is the regularization term and $\mathbf{s}$ is the vector of soft-segmentation masks values (estimated at the previous step) corresponding to the samples chosen for training of the descriptor. Then, the final appearance based soft-segmentation masks are generated by evaluating the regression model for each pixel.

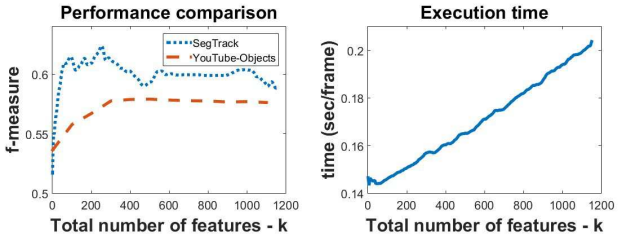

Figure 5. Features selection - optimization and sensitivity analysis.

2.5. Combining appearance and motion

The foreground and background have complementary properties at many levels, not just that of appearance. Here we consider that the object of interest must distinguish itself from the rest of the scene in terms of its motion pattern. A foreground object that does not move in the image, relative to its background, cannot be discovered using information from the current video alone. We take advantage of this idea by the following efficient approach.

Let $\mathbf{I}_t$ be the temporal derivative of the image as a function of time, estimated as difference between subsequent frames $\mathbf{I}_{t+1} - \mathbf{I}_t$. Also let $\mathbf{I}_x$ and $\mathbf{I}_y$ be the partial derivatives in the image w.r.t x and y. Consider $\mathbf{D}_m$ to be the motion data matrix, with one row per pixel p in the current frame corresponding to $[\mathbf{I}_x, \mathbf{I}_y, x\mathbf{I}_x, x\mathbf{I}_y, y\mathbf{I}_x, y\mathbf{I}_y]$ at locations estimated as background by the foreground segmentation estimated so far. Given such a matrix at time t we linearly regress $\mathbf{I}_t$ on $\mathbf{D}_m$. The solution would be a least square estimate of an affine motion model for the background using first order Taylor expansion of the image w.r.t time: $\mathbf{w}_m = ({\mathbf{D}_m}^T \mathbf{D}_m)^{-1} {\mathbf{D}_m}^T \mathbf{I}_t$. Here $\mathbf{w_m}$ contains the six parameters defining the affine motion (including translation) in 2D.

Then, we consider deviations from this model as potential good candidates for the presence of the foreground object, which is expected to move differently than the background scene. The idea is based on an approximation, of course, but it is very fast to compute and can be reliably combined with the appearance soft masks. Thus we evaluate the model in each location p and compute errors $|\mathbf{D}_m(p)\mathbf{w}_m - \mathbf{I}_t(p)|$. We normalize the error image and map it to $[0, 1]$. This produces a soft mask (using motion only) of locations that do not obey the motion model - they are usually correlated with object locations. This map is then smoothed with a Gaussian (with σ proportional to the distribution on x and y of the estimated object region).

At this point we have a soft object segmentation computed from appearance alone, and one computed independently, based on motion cues. The two soft results are multiplied to obtain the final segmentation.

Optional: refinement of video object segmentation Optionally we can further refine the soft mask by applying an off-the-shelf segmentation algorithm, such as GrabCut [19] and feeding it our soft foreground segmentation. **Note:** in our experiments we used GrabCut only for evaluation on SegTrack, where we were interested in the fine details of the objects shape. All other experiments are performed without this step.

3. Experimental analysis

Our experiments were performed on two datasets: YouTube-Objects dataset and SegTrack v2 dataset. We first introduce some qualitative results of our method, on the considered datasets (Figure 6). Note that for the final evaluation on the YouTube-Objects dataset, we also extract object bounding boxes, that are computed using the distribution of the pixels with high probability of being part of the foreground. Both position and size of the boxes are computed using a mean shift approach. For the final evaluation on the SegTrack dataset, we have refined the soft-segmentation masks, using the GrabCut algorithm [19]. In Tabel 2 we present evaluation results for different stages of our algorithm, along with the execution time, per stage. The F-measure is increased with each stage of our algorithm.

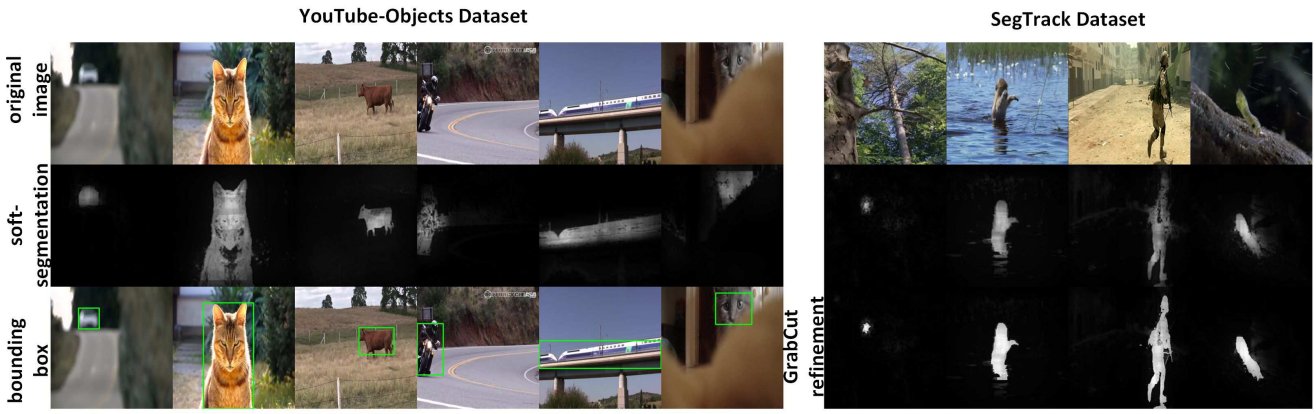

Figure 6. Qualitative results on YouTube-Objects dataset and SegTrack dataset.

3.1. YouTube-Objects dataset

Dataset: The YouTube-Objects dataset [18] contains a large number of videos filmed in the wild, collected from YouTube. It contains challenging, unconstrained sequences of ten object categories (aeroplane, bird, boat, car, cat, cow, dog, horse, motorbike, train). The sequences are considered to be challenging as they are completely unconstrained, displaying objects performing rapid movements, with difficult dynamic backgrounds, illumination changes, camera motion, scale and viewpoint changes and even editing effects, like flying logos or joining of different shots. The ground truth is provided for a small number of frames, and contains bounding boxes for the object instances. Usually, a frame contains only one primary object of the considered class, but there are some frames containing multiple instances of the same class of objects. Two versions of the dataset were released, the first (YouTube-Objects v1.0) containing 1407 annotated objects from a total of $\approx$ 570 000 frames, while the second (YouTube-Objects v2.2) contains 6975 annotated objects from $\approx$ 720 000 frames.

Metric: For the evaluation on the YouTube-Objects dataset we have adopted the CorLoc metric, computing the percentage of correctly localized object bounding-boxes. We evaluate the correctness of a box using the PASCAL-criterion (intersection over union ≥ 0.5).

Results: We compare our method against [10, 25, 18, 21, 17]. We considered their results as originally reported in the corresponding papers. The comparison is presented in Table 3. From our knowledge, the other methods were evaluated on YouTube-Objects v1.0, on the training samples (the only exception would be [21], where they have considered the full v1.0 dataset). Considering this, and the differences between the two versions, regarding the number of annotations, we have reported our performances on both versions, in order to provide a fair comparison and also to report the results on the latest version, YouTube-Objects v2.2 (not considered for comparison). We report results of the evaluation on v1.0 by only considering the training samples, for a fair comparison with other methods. Our method, which is unsupervised, is compared against both supervised and unsupervised methods. In the table, we have marked state-of-the-art results for unsupervised methods (bold), and overall state-of-the-art results (underlined). We also mention the execution time for the considered methods, in order to prove that our method is one order of magnitude faster than others (see Sec. 3.3 for details).

The performances of our method are competitive, obtaining state-of-the-art results for 3 classes, against both supervised and unsupervised methods. Compared to the unsupervised methods, we obtain state-of-the-art results for 7 classes. On average, our method performs better than all the others, and also in terms of execution time (also see Sec. 3.3). The fact that, on average, our algorithm outperforms other methods proves that it generalizes better for different classes of objects and different types of videos. Our solution performs poorly on the ”horse” class, as many sequences contain multiple horses, and our method is not able to correctly separate the instances. Another class with low performance is the ”cow” class, where we deal with same problems as in the case of ”horse” class, and where objects are usually still, being hard to segment in our system.

3.2. SegTrack v2 dataset

Dataset. The SegTrack dataset was originally introduced by [22], for evaluating tracking algorithms. Further, it was adapted for the task of video object segmentation [16]. We work with the second version of the dataset (SegTrack v2), which contains 14 videos ($\approx$ 1000 frames), with pixel level ground truth annotations for the object of interest, in every frame. The dataset is difficult as the included objects can be easily confused with the background, appear in different sizes and display complex deformations. There are 8 videos with one primary object and 6 with multiple objects, from 8 different categories (bird, cheetah, human,

Method Supervised?	[10] Y	[25] Y	[18] N	[21] N	[17] N	Ours $v1.0$ N	Ours $v2.2$ N
aeroplane	64.3	75.8	51.7	38.3	65.4	**76.3**	76.3
bird	63.2	60.8	17.5	62.5	67.3	**71.4**	68.5
boat	73.3	43.7	34.4	51.1	38.9	**65.0**	54.5
car	68.9	71.1	34.7	54.9	**65.2**	58.9	50.4
cat	44.4	46.5	22.3	64.3	46.3	**68.0**	59.8
cow	62.5	54.6	17.9	52.9	40.2	**55.9**	42.4
dog	71.4	55.5	13.5	44.3	65.3	**70.6**	53.5
horse	52.3	54.9	**48.4**	43.8	**48.4**	33.3	30.0
motorbike	78.6	42.4	39.0	41.9	39.0	**69.7**	53.5
train	23.1	35.8	25.0	**45.8**	25.0	42.4	60.7
Avg	60.2	54.1	30.4	49.9	50.1	**61.1**	54.9
time sec/frame	N/A	N/A	N/A	6.9	4	**0.35**	

Table 3. The CorLoc scores of our method and 5 other state-of-the-art methods, on the YouTube-Objects dataset (note that result for v2.2 of the dataset are not considered for comparison).

worm, monkey, dog, frog, parachute).

Metric. For the evaluation on the SegTrack we have adopted the average intersection over union metric. We specify that for the purpose of this evaluation, we use GrabCut for refinement of the soft-segmentation masks.

Results. We compare our method against [11, 24, 23, 17, 16]. We considered their results as originally reported by [23]. The comparison is presented in Table 4. Again, we compare our method against both supervised and unsupervised methods, and, in the table, we have marked state-of-the-art results for unsupervised methods (bold), and overall state-of-the-art results (underlined). The execution times are also introduced, to highlight that our method outperforms other approaches in terms of speed (see Sec. 3.3).

The performance of our method is competitive, while being an unsupervised method. Also, we prove that our method is one order of magnitude faster than the previous state-of-the-art [17] (see Sec. 3.3).

Method Supervised?	[11] Y	[24] Y	[23] Y	[17] N	[16] N	Ours N
bird of paradise	92	-	95	66	**94**	93
birdfall	49	71	70	59	**63**	58
frog	75	74	83	**77**	72	58
girl	88	82	91	73	**89**	69
monkey	79	62	90	65	**85**	69
parachute	96	94	92	91	93	**94**
soldier	67	60	85	69	**84**	60
worm	84	60	80	74	83	**84**
Avg	79	72	86	72	**83**	73
time sec/frame	>120	>120	N/A	4	242	**0.73**

Table 4. The average IoU scores of our method and 5 other state-of-the-art methods, on the SegTrack v2 dataset. Our reported time also includes the computational time required for GrabCut.

3.3. Computation time

One of the main advantages of our method is the reduced computational time. Note that all per pixel classifications can be efficiently implemented by linear filtering routines, as all our classifiers are linear. It takes only **0.35 sec/frame** for generating the soft segmentation masks (initial object cues: 0.05 sec/frame, object proposals refinement: 0.03 sec/frame, patch-based regression model: 0.25 sec/frame, motion estimation: 0.02 sec/frame (Table 2)). The method was implemented in Matlab, with no special optimizations. All timing measurements were performed using a computer with an Intel core i7 2.60GHz CPU. The method of Papazoglou et al. [17] report a time of 3.5 sec/frame for the initial optical flow computation, on top of which they run their method, which requires 0.5 sec/frame, leading to a total time of 4 sec/frame. The method introduced in [21] has a total of 6.9 sec/frame. For other methods, like the one introduced in [24, 11], it takes up to 120 sec/frame only for generating the initial object proposals using the method of [3]. We have no information regarding computational time of other considered methods, but due to their complexity we expect them to be orders of magnitude slower than ours.

4. Conclusions

We have presented an efficient fully unsupervised method for object discovery in video that is both fast and accurate. It achieves state of the art results on a challenging benchmark for bounding box object discovery and very competitive performance on a video object segmentation dataset. At the same time, our method is fast, being at least an order of magnitude faster than competition. We achieve an excellent combination of speed and performance by exploiting the contrasting properties between objects and their scenes, in terms of appearance and motion, which makes it possible to select positive feature samples with a very high precision. We show, theoretically and practically, that high precision is sufficient for reliable unsupervised learning (since positives are generally less frequent than negatives), which we perform both at the level of single pixels and at the higher level of groups of pixels, which capture higher order statistics about objects appearance, texture and shape. The top speed and accuracy of our method, combined with theoretical guarantees that hold in practice under mild conditions, make our approach unique and valuable in the quest for solving the unsupervised learning problem in video.

Acknowledgements: The authors thank Otilia Stretcu for helpful feedback. This work was supported by UEFISCDI, under project PN-III-P4-ID-ERC-2016-0007.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2189–2202, 2012.

[2] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, 2012.

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

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[5] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 670–677. IEEE, 2009.

[6] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[7] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[8] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *European Conference on Computer Vision*, pages 656–671. Springer, 2014.

[9] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2083–2090, 2013.

[10] Y. Jun Koh, W.-D. Jang, and C.-S. Kim. Pod: Discovering primary objects in videos based on evolutionary refinement of object recurrence, background, and primary object models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1068–1076, 2016.

[11] 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.

[12] M. Leordeanu, R. Collins, and M. Hebert. Unsupervised learning of object features from video sequences. In *IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*, volume 1, page 1142. IEEE Computer Society; 1999, 2005.

[13] M. Leordeanu, A. Radu, S. Baluja, and R. Sukthankar. Labeling the features not the samples: Efficient video classification with minimal supervision. *arXiv preprint arXiv:1512.00517*, 2015.

[14] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2290–2297, 2009.

[15] F. Li, J. Carreira, G. Lebanon, and C. Sminchisescu. Composite statistical inference for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3302–3309, 2013.

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

[17] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013.

[18] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3282–3289. IEEE, 2012.

[19] 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.

[20] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 370–377. IEEE, 2005.

[21] O. Stretcu and M. Leordeanu. Multiple frames matching for object discovery in video. In *BMVC*, pages 186–1, 2015.

[22] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label mrf optimization. *International journal of computer vision*, 100(2):190–202, 2012.

[23] L. Wang, G. Hua, R. Sukthankar, J. Xue, Z. Niu, and N. Zheng. Video object discovery and co-segmentation with extremely weak supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[24] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635, 2013.

[25] Y. Zhang, X. Chen, J. Li, C. Wang, and C. Xia. Semantic object segmentation via detection in weakly labeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3641–3649, 2015.

[26] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.