Causal Video Object Segmentation From Persistence of Occlusions

Brian Taylor   Vasiliy Karasev   Stefano Soatto
UCLA Vision Lab, University of California, Los Angeles, CA 90095
btay@cs.ucla.edu   vasiliykarasev@ucla.edu   soatto@ucla.edu

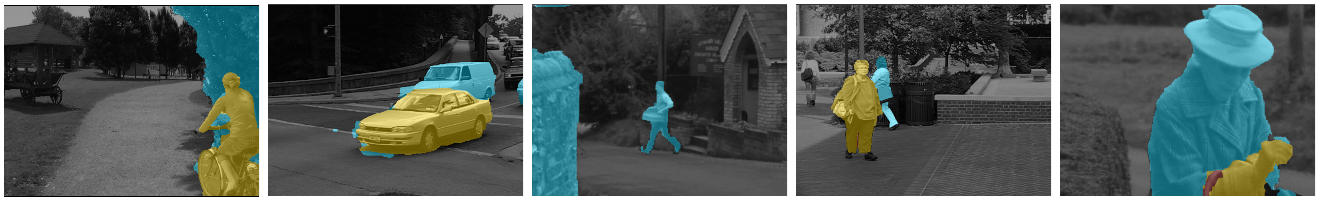


Figure 1: *Sample outcomes of our scheme: background* $c(x) = 0$ *(gray) and foreground layers* $c(x) = 1$, $c(x) = 2$, $c(x) = 3$ *indicated by* 🟦, 🟨, 🟥 *respectively. On the far right, our algorithm correctly infers that the bag strap is in front of the woman's arm, which is in front of her trunk, which is in front of the background. Project page:* http://vision.ucla.edu/cvos/

Abstract

Occlusion relations inform the partition of the image domain into "objects" but are difficult to determine from a single image or short-baseline video. We show how long-term occlusion relations can be robustly inferred from video, and used within a convex optimization framework to segment the image domain into regions. We highlight the challenges in determining these occluder/occluded relations and ensuring regions remain temporally consistent, propose strategies to overcome them, and introduce an efficient numerical scheme to perform the partition directly on the pixel grid, without the need for superpixelization or other preprocessing steps.

1. Introduction

Partitioning the image domain into regions that correspond to "objects" is elusive absent an explicit definition of objects that has a measurable correlate in the image. Gestalt principles [33] provide grouping criteria: continuity, regularity, proximity, compactness, the last of which (figure/ground, or occlusion) is best informed by video. Occlusions have been used extensively for grouping [32, 5, 7, 3]. A feature of [3] is that grouping is obtained via a linear program: local ordering constraints provided by *occluder/occluded relations* are integrated to globally partition the image domain into *depth layers*. The challenge is that errors in determining occlusion relations can have a cascading effect.

Occlusions are usually detected from the residual of optical flow, but even assuming this detection is correct, *occluder relations* are non-trivial to determine. As we show in Fig. 2, correct determination of the occluder requires either knowledge of the motion of the occluded region (which is undefined), or knowledge of its partition into regions. Hence the conundrum: to determine occlusion relations, so that objects can be segmented, we need to know the objects in the first place. The *first contribution* of our work is to break the conundrum by leveraging motion and appearance priors to hallucinate motion in the occluded region. With the *occluder/occluded* relations we can obtain a depth-layer partition for the image domain. In video, however, nuisances such as motion blur, quantization, scale, and lack of motion can cause layer segmentation to fail. Thus, the *second contribution* is a causal framework for integrating occlusion cues exploiting temporal consistency priors to partition the video into depth layers. Our *third contribution* is to make the solution of the resulting optimization problem efficient using a primal-dual scheme. Our proposed method is competitive to state-of-the-art approaches qualitatively in visual boundaries and quantitatively in numerical benchmarks, while processing video sequences causally, rather than in batch. Samples from our scheme are shown in Fig. 1.

The paper is organized as follows: we set up our problem in Sec. 2. We describe our first contribution in determining occluder relations in Sec. 2.1 and how we leverage prior work [3] in Sec. 2.2. Sec. 3 explores how we causally integrate cues to construct priors for foreground regions in Sec. 3.1, obtain persistent object boundaries in Sec. 3.2, and aggregate occluder relations in Sec. 3.3. Our final model is presented in Sec. 3.4. Implementation and optimization details are covered in Sec. 4–5, including our approach for hallucinating motion in the occluded regions in Sec. 4.2. Empirical evaluation appears in Sec. 6, where we show that the typical failure modes of prior approaches stemming from unreliable occlusion relations are mitigated.

1.1. Related Work

A large number of methods have been proposed for partitioning a video sequence into non-overlapping regions with unique labels, using motion, appearance or their combination [24, 6, 17, 31, 21, 10, 34, 15, 36, 23, 22]. These approaches are susceptible to oversegmentation, which video *object* segmentation attempts to mitigate by assigning a single label to each object. The problem can be cast as multi-label classification, in which a unique label is attached to each object [22], or as binary "foreground"/"background" (FG/BG) classification [11, 21, 36, 23]. While our work produces *depth layers*, and not object labels, these could be added post-mortem.

Many approaches operate *offline* (or *non-causally*), with the entire video available for processing [22, 21, 36, 23], which scales poorly with sequence length, although "streaming" approaches can be used [31, 34]. Our approach is *online* (or *causal*), and is closely related to tracking [24, 4, 9], which, unlike us, requires manual initialization.

Estimation of segmentation masks, motion, and depth ordering can be formulated jointly [12, 32, 5, 19, 26, 18, 20, 28, 25, 9, 35], but the resulting problem is nonconvex and requires a substantial computational effort. We separate motion estimation from segmentation and depth ordering, and focus on the latter, which makes a scalable convex formulation possible.

2. Video segmentation with layers

Let $I_t : D \to \mathbb{R}^3$ be an image of a video $\{I_t\}_{t=1}^T$ defined on the domain $D \subset \mathbb{R}^2$. We seek to partition D into regions, each associated with an integer *depth order*, represented by a function $c_t : D \to \mathbb{Z}_+$ indicating to which layer each pixel belongs. A layer is then $c_t^{-1}(i) = \{x \in D | c_t(x) = i\}$, where $c_t(x) = 0$ denotes the background and larger values of c_t indicate "foreground" regions $c_t(x) = 1, 2, 3, \dots$. The connected components of non-zero regions correspond to individual objects. It was shown by [5, 3] that depth layers can be inferred from occlusion phenomena, that occur as a result of object or viewer motion, causing parts of the scene to become hidden and others revealed. These inform local order relations between surfaces in the scene: when a surface becomes *occluded*, the image region where it projected to becomes occupied by the *occluder*, which is therefore closer to the viewer. These occluder-occluded relationships can be used as cues for segmenting regions in the image that back-project to distinct objects in the scene.

2.1. The "occluder" and the "occluded"

Under the assumptions of Lambertian reflection, constant illumination, and co-visibility typically implicit in most optical flow algorithms, $I_t(x)$ is related to $I_{t+1}(x)$ by the brightness-constancy equation

$$I_t(x) = I_{t+1}(w_t^{t+1}(x)) + n_t(x), \; x \in D \backslash \Omega_t^{t+1}(x), \quad (1)$$

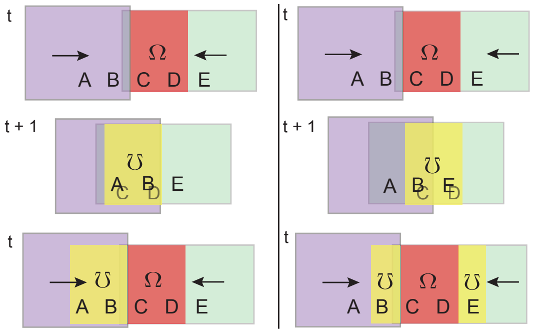


Figure 2: *Initial (top) and final (middle) views of a smaller square sliding under a larger one, producing an occluded region* Ω *in red (subscripts dropped for readability). Two alternate hypotheses (left and right) for the occluder* (℧) *in yellow produce different constraints (bottom).* Left*:* Ω *moves with* E *and slides under* A ∪ B. Right*: the occluder is split in two—*B *occludes* C *and* E *occludes* D*. Disambiguation requires either knowledge of the motion in* Ω*, which is undetermined as it is occluded, or the object segmentation, which is the final goal.*

where w_t^{t+1} is the deformation field that warps the domain of I_t into I_{t+1} and n_t lumps together all un-modeled phenomena and violations of the assumptions. Often, w_t^{t+1} is represented by the optical flow field v_t^{t+1} by $w_t^{t+1}(x) = x + v_t^{t+1}(x)$. The above holds on the entire image domain except in the *occluded regions* Ω_t^{t+1}, where surfaces visible at time t are no longer visible at $t+1$. In this region, the optical flow is not defined, but can be *extrapolated* from the "co-visible" regions via regularization. Occluded regions are easy to find as a byproduct of optical flow estimation [2], as they yield a large residual n_t via backward flow. What is not easy to find is the *occluder*.

The defining characteristic of the occluder point $y_t^c \in \mho_t^{t+1}$ (the occluder region) corresponding to the occluded point $y_t \in \Omega_t^{t+1}$ (the occluded region) is

$$w_t^{t+1}(y_t^c) = y_t. \quad (2)$$

This equation is somewhat unintuitive as the left hand-side lives in the domain of the image at time $t+1$ whereas the right-hand side is defined only at time t. This can be interpreted as

$$y_t^c = w_{t+1}^t(y_t), \quad (3)$$

which is completetely agnostic of the motion of the occluded region.

Consider Fig. 2: The occluded region, C ∪ D, could slide under the larger rectangle, and become occluded by A ∪ B. However, C and D could also actually correspond to different objects, and move independently. In this case, B could be the occluder of C and E could be the occluder of D. To disambiguate between these two hypotheses, we need to know either the motion of D, which is not possible since it is occluded, or the object partition, which is our goal in the first place. In the example in question, using (2) would favor the hypothesis of B occluding C and E occluding D (right half of Fig. 2). This would yield two ordering constraints, $c(\mathsf{B}) > c(\mathsf{C})$ and $c(\mathsf{E}) > c(\mathsf{D})$ that hinge on the occluded region and impose no constraints between the visible regions B and E. The latter constraint is also incorrect in the example (Fig. 2 bottom right).

However, while the motion in the occluded region is not determined, it can be hallucinated exploiting regularization priors. Even with a coarse estimate of the motion of D, we could determine if it moves similarly to E, in which case it cannot be occluded by it and must instead be occluded by B. Therefore, in our approach we extrapolate motion to the occluded region, so as to attribute it to a possible occluder. In Sec. 4.2, we discuss how to exploit natural image and motion priors to achieve this. Of course, one could resort to such priors and photometric characteristics of the occluded region to directly determine the grouping of C, D, and E. But again if this was easy, we would have already solved the problem of object segmentation.

2.2. From local ordering constraints to layers

In [3], the following convex model for inferring c_t from occlusion cues was proposed:

$$\begin{aligned} c_t = \arg \min_{c_t : c_t \geq 0} \int_D g_t(x) |\nabla c_t(x)| dx \\ \text{s.t. } c_t(y^c) - c_t(y) \geq 1 \;\; \forall (y^c, y) \in O_t. \end{aligned} \tag{4}$$

O_t denotes the set of occlusion cues composed of pairs (y^c, y), where y^c lies on the occlud*ing* surface, and y lies on the surface that was (will be) occlud*ed* in the previous (next) frame. The objective $\int_D g_t(x)|\nabla c_t(x)|dx$ is just weighted total variation (TV), with the data-dependent affinity weights (denoted by $g_t(x)$) being small at image and motion boundaries and large otherwise. Note that the "data-term" enters the optimization as a set of constraints which require occluded-occluder pairs to lie in different layers: specifically, the occluder must lie in the layer closer to the viewer (higher values of c_t). An overview of this approach is shown in Fig. 3. While this optimization problem relaxes the integer constraint ($c_t : D \to \mathbb{Z}_+$), empirically the solutions are piecewise constant and integer valued.

Although this model is formulated for a single time instant t, three frames $(t-1, t, t+1)$ are necessary to obtain occlusion cues. However, they are typically not sufficient when small inter-frame motion produces unreliable occlusion constraints. Next, we exploit temporal persistence to overcome this problem.

3. Incorporating motion cues causally

Our causal framework leverages a rich history of image frames, the segmentation cues from those frames (occlusions and weights), and previous layer estimates to facilitate segmentation in the current frame. Large-displacement propagation of these cues via w_t^{t-1} is unstable, rendering cues unusable. But when the motion becomes large, occlusions become easier to detect, making the past unnecessary for segmentation. Thus, these cues are complementary—when the motion is large, sufficient occlusion cues are produced, and w_t^{t-1} is erroneous. When the motion is small, occlusion cues are few, but propagation is reliable. This motivates an adaptive integration of cues based on motion. A weighting function $m_t(x) \doteq \alpha \exp(-|v_t^{t-1}|/\mu_v)$ is used, where $\alpha \in [0,1]$, v_t^{t-1} is the optical flow, and μ_v is the mean value of v for this frame. The weight decreases with large motion, regardless of how long ago it occurred. The following sections describe the temporal cues leveraged in our framework. Note that the variable being optimized over is always c_t, and c_{t-1} is always available as a result of previous optimization.

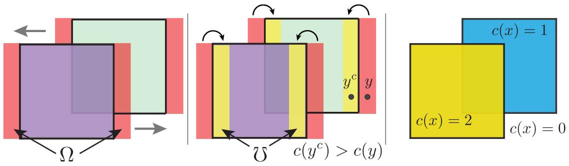


Figure 3: Left: *The motion of two objects generate occlusions and disocclusions (both denoted by Ω, shown in red).* Middle: *each occluded region is attributed to a local occluder ($\mho$, shown in yellow). Occluder-occluded relationship constrains objects' depth-order.* Right: *resulting depth layers.*

3.1. Once an object, always an object

Layer values $c_t(x)$ are not constant over time, as objects can move in front of one another and switch order of their distance to the viewer. However, once an object is detected, it should not later be labeled as background–even if it stops moving and produces no occlusion cues for segmentation.

This can be enforced causally using the prior segmentation result (c_{t-1}) via a (convex) constraint:

$$c_t(x) \geq 1 \;\; \forall x \in F, \;\; F = \{c_{t-1}(w_t^{t-1}(x)) \geq 1\} \tag{5}$$

where F is the indicator of the previous frame's foreground region warped into the current frame. To mitigate errors in prior segmentations, we relax the constraint and penalize violations with a hinge loss:

$$\int_D \kappa_t(x) \max\big(0, 1 - c_t(x)\big) dx \tag{6}$$

with κ_t being the cost of violating the constraint. Choosing $\kappa_t(x) = 0$ for x outside F allows us to write the penalty as an integral over entire image domain D. As $\kappa_t(x) \to \infty$ for $x \in F$, the hard constraint (5) is recovered.

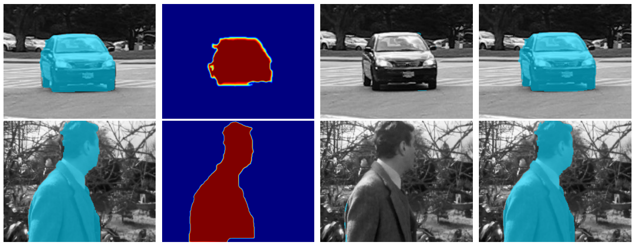


Figure 4: *c_{t-1} (column 1) is used to compute the foreground prior (κ_t) (column 2). Without κ_t, the resulting c_t completely misses the objects (column 3), however with κ_t, c_t succeeds (last column). Note c_{t-1} and c_t look very similar—κ_t helps most during small-baseline motion when occlusion cues are weak but c_{t-1} easily predicts c_t.*

The cost of violating the constraint is computed recursively, with initial condition $\kappa_1(x) = 0$, as

$$\kappa_t(x) = m_t(x)\kappa_{t-1}(w_t^{t-1}(x)) + \mathbb{1}\{c_{t-1}(w_t^{t-1}(x)) \geq 1\}$$

where $\mathbb{1}$ is a characteristic function ($\mathbb{1}\{X\} = 1$ if X is true, and is 0 otherwise). This *foreground prior* boosts $\kappa_t(x)$ wherever the corresponding points are labeled as foreground in the previous frame and diminishes it over time and motion as described above. As demonstrated in Fig. 4, whenever motion is small, instantaneous occlusion cues are insufficient to perform segmentation, and this notion of temporal consistency is helpful.

To avoid the entire image domain from becoming foreground, we introduce an additional regularization penalizing layer values

$$\tau \int_D c_t(x)dx. \tag{7}$$

This is similar to the regularization used in [3], although they use the ℓ_∞ norm, whereas here we use ℓ_1. This term encourages pixels to lie in the background layer, unless sufficient evidence pushes them into the foreground.

3.2. Persistent layer boundaries

While depth-layer *values* are not persistent, their boundaries are. Unless objects split or merge, we have

$$\mathbb{1}\{\nabla c_t(x) \neq 0\} = \mathbb{1}\{\nabla c_{t-1}(w_t^{t-1}(x)) \neq 0\}. \tag{8}$$

This is a nonconvex constraint. However, enforcing $\nabla c_t(x) = 0$ wherever $c_{t-1}(w_t^{t-1}(x)) = 0$ is simple (a linear constraint), and its relaxed version with a hinge loss and associated cost $u_t(x)$ is equivalent to increasing weights in TV regularization (shown in [30]). This leaves the hard part: enforcing $\nabla c_t(x) \neq 0$ wherever $c_{t-1}(w_t^{t-1}(x)) \neq 0$. To remain within a convex optimization framework, we treat this as a bias and set the corresponding $u_t(x)$ to be negative, which *decreases* the corresponding TV weights (which are kept nonnegative to preserve convexity). This *layer unity* prior is also computed recursively, with $u_1(x) = 0$, as

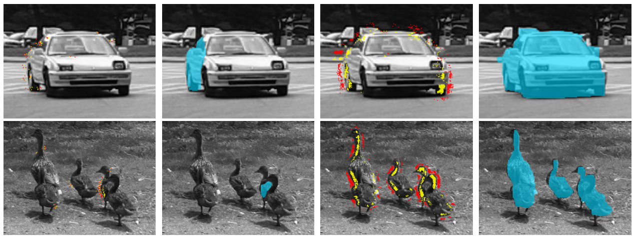


Figure 5: *Occlusion cues from the current frame alone (O_t), with occluded points (Ω) in red and occluder points ($\mho$) in yellow, (column 1) fail to segment the objects (column 2). However, aggregating constraints over time ($\bar{O}_t$) (column 3) succesfully recovers all of them (last column).*

$$\begin{aligned} u_t(x) =& m_t(x)u_{t-1}(w_t^{t-1}(x)) + \mathbb{1}\{\nabla c_{t-1}(w_t^{t-1}(x)) = 0\} \\ &- \mathbb{1}\{\nabla c_{t-1}(w_t^{t-1}(x)) \neq 0\}. \end{aligned}$$

We also perform temporal aggregation of the TV affinity weights. In each frame, we compute the *boundary strength* $\rho_t(x) \in \mathbb{R}_+$, as described in Sec. 4. The aggregated boundary strength $\bar{\rho}_t(x)$ is (with $\bar{\rho}_1(x) = 0$)

$$\bar{\rho}_t(x) = m_t(x)\bar{\rho}_{t-1}(w_t^{t-1}(x)) + \rho_t(x). \tag{9}$$

The aggregated TV weights used in the optimization are

$$g_t(x) = \max(0, 1 - \bar{\rho}_t(x) + u_t(x)). \tag{10}$$

3.3. Occlusion cue aggregation

Instantaneous occlusion constraints (O_t) are accumulated into the aggregated constraints set $\bar{O}_t = w_{t-1}^t(\bar{O}_{t-1}) \cup O_t$, where past constraints $\bar{O}_{t-1}$ are propagated to the current frame by the motion of the occluder $w_{t-1}^t(y^c)$ (see Fig. 5). The base condition is $\bar{O}_1 = O_1$. The constraint penalty weights λ, computed by (4.1), are adjusted over time by

$$\lambda_{t,i} = m_t(y_i^c)\lambda_{t-1,i} + \mathbb{1}\{c_{t-1}(w_t^{t-1}(y_i^c)) \geq c_{t-1}(w_t^{t-1}(y_i))\}.$$

3.4. Overall model

The final model that incorporates occlusion cues, weights, foreground and unity priors is

$$\begin{aligned} c_t = \arg\min_{c_t \geq 0} & \int_D g_t(x)|\nabla c_t(x)|dx + \tau \int_D c_t(x)dx \\ &+ \int_D \kappa_t(x)\max\big(0, 1 - c_t(x)\big)dx \\ &+ \sum_{\substack{i=1 \\ (y_i^c, y_i) \in \bar{O}_t}}^{N} \lambda_i \max\big(0, 1 - c_t(y_i^c) - c_t(y_i)\big), \end{aligned} \tag{11}$$

where the first term (weighted TV) ensures that the result is piecewise constant, the second term (foreground prior) encourages regions to have nonzero layer values wherever $\bar{\kappa}_t(x)$ is large, the third (model selection) term prevents the creation of spurious layers, and the fourth is the penalty for violating the occlusion constraints.

4. Implementation details

For each frame, we incorporate appearance, edge, and motion information into the weights $\rho_t(x)$ in (9) as follows:

$$\rho_t(x) = 1-\big(\beta_I h(|\nabla I(x)|)+\beta_E h(E(x))+\beta_v h(|\nabla v_t^{t+1}(x)|)\big)$$

where $h(x) = \exp(-x/\mu_x)$, μ_x is the average value of x. $E(x) \in [0,1]$ is the output of an edge detector [13] with $E(x) \approx 1$ at the boundaries. In our experiments, $(\beta_I, \beta_E, \beta_v) = (0.2, 0.4, 0.4)$. Following [23], we also adjust the motion term by the difference in flow angles at the pixels where flow magnitude is small.

4.1. Occlusion constraint weights

Often the occluded and occluding surfaces differ in appearance, motion, and are separated by a strong image boundary, suggesting λ be computed in a fashion similar to (4):

$$\lambda_i = \eta_i\big(1 - \big(\beta_I h(|I(y_i^c) - I(y_i)|) + \beta_E h(\hat{E}(y_i^c, y_i)) \\ + \beta_v h(|v_t^{t+1}(y_i^c) - v_t^{t+1}(y_i)|)\big)\big)$$

where the gradient operator is replaced by a difference between appearance, edge, and motion statistics of y^c and y. Here, $E(x)$ is replaced by $\hat{E}(x_1, x_2)$—the strongest edge response on the line connecting y^c and y. We additionally validate y^c and y as an occluder-occluded pair with weight η, which measures the degree to which y and y^c move toward each other. Indeed, unless they do so, $\mho_t^{t+1}$ cannot take the place of Ω_t^{t+1}, i.e. when $\eta(y^c, y)$ in

$$\begin{aligned} \Delta(y^c, y) &= (v_t^{t+1}(y^c) - v_t^{t+1}(y))^T(\frac{y^c - y}{\|y^c - y\|}) \\ \eta(y^c, y) &= \max(0, 1 - \exp(-\theta\, \Delta(y^c, y))) \end{aligned} \tag{12}$$

is small, then y^c is less likely to occlude y. We choose $\theta = 2$ so that $\Delta(y^c, y) = 1$ yields a high score. $\lambda_i \approx 1$ whenever the appearance and motion of y^c and y are "different" and the points are moving toward each other. Finally, assuming that the occluded and occluding surfaces differ in appearance, we can locally perturb constraints with the goal of correcting them; this procedure is described in [30]. Altogether, these factors alter the constraints to help us discount potentially erroneous cues, which occur due to inevitable errors in optical flow and occlusion estimation.

4.2. Flow extrapolation over the occlusion region

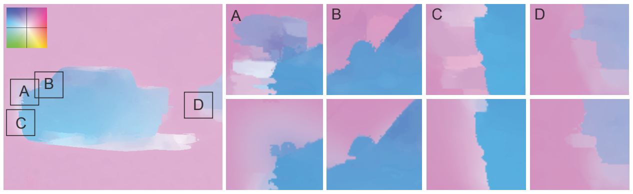


Figure 6: *Cross bilateral filtering extrapolates flow in Ω via motion and appearance priors, facilitating reliable occluder determination. Left: The extrapolated motion field $\hat{v}_t^{t+1}$. Boxes highlight occluded regions where notable change (often improvement) occurs. Right: For each box, v_t^{t+1} (top) and $\hat{v}_t^{t+1}$ (bottom) are shown.*

As noted in Sec. 2.1, $v_t^{t+1}(x)$ for $x \in \Omega_t^{t+1}$ is undefined (1) and filled in by the regularizer, which corresponds to enforcing priors on motion. The simplest priors rely solely on continuity, tending to smooth motion boundaries, while more sophisticated ones attempt to preserve them. We use the cross-bilateral filter [14] to enforce such priors on v_t^{t+1} in the occluded regions based on the backward flow v_t^{t-1}:

$$\begin{aligned} \hat{v}_t^{t+1}(z) = \frac{1}{V_z}\int_D v_t^{t+1}(x) P(x \notin \Omega_t^{t+1}) \\ \mathcal{G}(v_t^{t-1}(x) - v_t^{t-1}(z), \sigma_v)\mathcal{G}(x - z, \sigma_x)dx, \end{aligned} \tag{13}$$

where $\hat{v}_t^{t+1}$ is the extrapolated forward flow, $P(x \notin \Omega_t^{t+1})$ is the probability of x being visible, $\mathcal{G}$ is the gaussian kernel $\mathcal{G}(x, \sigma) = \exp(-\|x\|^2/2\sigma^2)$, and V_z is a normalization term. We can filter the backward flow $\hat{v}_t^{t-1}$ by exchanging $t+1$ with $t-1$ and vice versa. Extrapolating flow is key to determining the occluder (Fig. 2), but cannot be proven "correct" as it hinges critically on the choice of prior. v_t^{t+1} is computed using publicly-available code [27] ("classic-nl"), and occlusions are computed by thresholding the residual image. See [30] for further details.

4.3. Foreground prior region

In practice, motion estimation makes mistakes near object boundaries (e.g. occluded regions). When computing κ_t, we first warp κ_{t-1} to the current frame and then use morphological operations to erode the edges proportionally to the magnitude of the flow in that region. This ensures the prior does not leak outside of the object regions, but produces a poor estimate near the boundaries. To help recover the structure of these edges, we incorporate a set of local shape classifiers as in [4] to better capture and predict the shape of the object boundary, the details of which are in [30].

5. Optimization

The optimization problem (11) is convex but large enough that off-the-shelf methods cannot solve it without resorting to superpixels or other pre-processing to reduce its dimension. Here we present an efficient numerical primal-dual scheme based on [8] that allows us to solve it on the pixel grid.

The indicator function – not to be confused with characteristic function $\mathbb{1}$ used above – of a set A is defined by $I_A(x) = 0$ for $x \in A$ and $I_A(x) = \infty$ for $x \neq A$. For a function f, the *convex conjugate* is defined as $f^*(y) \doteq$

Algorithm 1 Layer Solver

```
Initialize: Pick $\sigma_y, \sigma_c > 0$, $\sigma_y \sigma_c \leq \frac{1}{8}$, and $\theta \in [0,1]$.
Arbitrarily initialize feasible $y_1^0, y_2^0, c^0$. Set $\bar{x}^0 = c^0$.
Perform iterates for $k = 0, 1, 2, \ldots$:
    $y_1^{k+1} = \mathbf{prox}_{\sigma_y F_1^*}\left(y_1^k + \sigma_y \mathcal{D} \bar{x}^k\right)$
    $y_2^{k+1} = \mathbf{prox}_{\sigma_y F_2^*}\left(y_2^k + \sigma_y \mathcal{D}_{occ} \bar{x}^k\right)$
    $c^{k+1} = \mathbf{prox}_{\sigma_c G}\left(c^k - \sigma_c(\mathcal{D}^T y_1^{k+1} + \mathcal{D}_{occ}^T y_2^{k+1})\right)$
    $\bar{x}^{k+1} = c^{k+1} + \theta(c^{k+1} - c^k)$.
```

$\sup_x y^T x - f(x)$. The *prox operator* of f is defined as

$$\mathbf{prox}_{\sigma f}(y) \doteq \arg\min_x \frac{1}{2\sigma} \|x - y\|^2 + f(x). \quad (14)$$

Since the optimization is performed on a finite pixel grid, the depth values c can be written as a vector $c \in \mathbb{R}_+^n$, with c_i indicating the layer value at the i-th pixel. We denote by $\mathcal{D}$ the gradient operator represented by a matrix of finite differences. Weights associated with the edges are denoted by the diagonal matrix W. A difference matrix for occlusion constraints is denoted by $\mathcal{D}_{occ}$ and the cost of violating constraints by λ. As before, τ is used for regularization and κ is a weighted indicator of the foreground region. We can then write the objective in shorthand as

$$\min_c \|W\mathcal{D}c\|_1 + \tau^T c + \kappa^T \max(0, 1 - c) + \lambda^T \max(0, 1 - \mathcal{D}_{occ} c) + I_{\{c \geq 0\}}(c). \quad (15)$$

Let $G(c) = \tau^T c + \kappa^T \max(0, 1 - c) + I_{\{c \geq 0\}}(c)$. Also, let $z_1 = \mathcal{D}c$, $z_2 = \mathcal{D}_{occ} c$, construct the functions $F_1(z_1) = \|W z_1\|_1$, $F_2(z_2) = \lambda^T \max(0, 1 - z_2)$, and introduce the dual variables y_1, y_2. The augmented Lagrangian follows as

$$\min_{z_1, z_2, c} \max_{y_1, y_2} F_1(z_1) + F_2(z_2) + G(c) + y_1^T (\mathcal{D}c - z_1) + y_2^T (\mathcal{D}_{occ} c - z_2), \quad (16)$$

or, equivalently, using the convex conjugates, as

$$\min_c \max_{y_1, y_2} G(c) - F_1^*(y_1) - F_2^*(y_2) + y_1^T \mathcal{D} c + y_2^T \mathcal{D}_{occ} c.$$

This saddle-point problem is addressed in [8], so we can apply their primal-dual algorithm shown in Alg. 1.

Alg. 1 depends on the ability to compute proximal operators for G, F_1^* and F_2^*. All three operators have simple closed form solutions that require few arithmetic operations:

$$\mathbf{prox}_{\sigma G}(y) = \max\left(0, \min\left(y - \sigma\tau + \sigma\kappa, \max\left(1, y - \sigma\tau\right)\right)\right) \quad (17)$$

$$\mathbf{prox}_{\sigma F_1^*}(y) = \text{sign}(y) \min\{\text{diag}(W), |y|\} \quad (18)$$

$$\mathbf{prox}_{\sigma F_2^*}(y) = \min\left(\max\left(y - \sigma 1, -\lambda\right), 0\right) \quad (19)$$

Derivation details are reported in [30].

6. Experiments

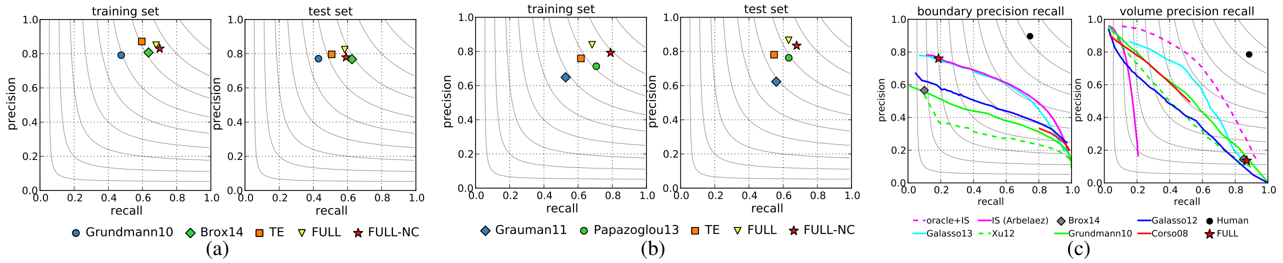
Our method segments video into depth layers. Unfortunately, no benchmark dataset is available to evaluate it directly. However, our method can be modified to produce binary and multi-label segmentations; leveraging this, we evaluate the algorithm on two datasets: MoSeg [22] (designed for video *object* segmentation with no consideration for depth ordering), on which we focus, as well as BVSD [16] (designed for video segmentation).

Evaluation methodology. We follow the process described in [22]. The dataset contains 59 sequences, ranging from 19 to 800 frames. Each has pixel-wise ground truth annotation for a sparse subset of frames (3–41). As in [22], we report *precision*, *recall*, *F-measure*, and the number of extracted objects (regions with F-measure ≥ 0.75). For multi-label segmentation tasks, we treat each connected component of the depth layers as a unique "object". We also evaluate on foreground/background (FG/BG) video object segmentation, which come directly from depth layers as $FG \doteq \{x : c(x) \geq 1\}$, $BG \doteq \{x : c(x) = 0\}$. *Precision*, *recall*, and *F-measure* are reported on the ground truth annotations converted to binary masks. Note we cannot evaluate "number of extracted objects" in the FG/BG scenario.

The methods we compare against ([17, 21, 23, 22]) are non-causal and "batch", whereas our method is causal. Since we do not know the future, we do not detect objects until they undergo sufficient motion, which sometimes causes us to miss objects in the beginning of video sequences. To fairly compare against non-causal methods, we also perform a non-causal evaluation (reported as "NC")—we run our algorithm forward in time to accumulate all priors, and then backward in time. The latter half is used for evaluation.

Effects of system components. In Sec. 3 we described individual components of our model and showed examples where they improved results (see Fig. 4, 5). Here we quantify this improvement. We evaluate [3] ("BASIC"), their temporal extension ("TE"), foreground-background prior (Sec. 3.1, "FG"), and the full model ("FULL"). In addition, we evaluated the full model without flow extrapolation (Sec. 4.2) to understand its effects ("NOFE"). These results are reported in Table 1. "BASIC" does not use long-term temporal information. "TE" integrates weights using previous segmentations, increasing the cost of making a cut away from object boundaries. "FG" discourages previously segmented regions from falling into background. "FULL" is a combination of all components.

The "BASIC" method does not use temporal information, so on the multi-label benchmark, whenever objects disappear (as they often do, due to insufficient motion) and re-appear, they are assigned a *new* object label. Long-term integration helps avoid missed detections and propagates object labels throughout the sequence. Performance on the FG/BG evaluation suggests that objects are often not detected at all.



(a) (b) (c)

Figure 7: *(a-b) Comparison on MoSeg: (a) multi-label segmentation, (b) FG/BG segmentation. (c) Comparison on BVSD.*

Multi-label segmentation

	Training set (29 sequences)				**Test set** (30 sequences)			
	P	R	F	$N/65$	P	R	F	$N/69$
BASIC	84.90	53.10	65.34	10	78.80	44.49	56.87	4
TE	**87.20**	59.60	70.81	17	79.64	50.73	61.98	7
FG	86.98	60.99	71.71	18	79.04	52.08	62.79	10
NOFE	86.67	58.06	69.54	14	80.71	50.64	62.24	8
FULL	85.00	67.99	75.55	21	**82.37**	58.37	**68.32**	17
FULL-NC	83.00	**70.10**	**76.01**	**23**	77.94	59.14	67.25	15
[17]	79.17	47.55	59.42	4	77.11	42.99	55.20	5
[22]	81.50	63.23	71.21	16	74.91	**60.14**	66.72	**20**

Binary segmentation

	Training set (29 sequences)				**Test set** (30 sequences)			
	P	R	F	-	P	R	F	-
BASIC	**89.99**	40.86	56.21	-	**93.21**	33.69	49.49	-
TE	75.94	61.64	68.05	-	78.11	54.68	64.33	-
FG	75.93	63.07	68.91	-	76.97	56.16	64.94	-
NOFE	68.92	66.09	67.48	-	74.27	53.99	62.52	-
FULL	83.92	68.19	75.24	-	86.54	63.20	73.05	-
FULL-NC	79.26	**78.99**	**79.12**	-	83.41	**67.91**	**74.87**	-
[21]	64.86	52.70	58.15	-	62.32	55.97	58.97	-
[23]	71.34	70.66	71.00	-	76.29	63.29	69.18	-

Table 1: Comparison of our approach (rows 4–5) to baselines using individual components (rows 1–3) and state-of-the-art (rows 6–7) on the MoSeg dataset. **R**≐recall, **P**≐precision, **F**≐F-measure, **N**≐ number of extracted objects.

Precision decreases for the "FULL" system due to an increased number of "false positives"—often we detect more objects than labeled in the annotation (see Fig. 8). "NC" provides a small performance boost by allowing us to label objects before they move.

Video object segmentation. In Table 1 and Fig. 7 we report results of the comparison with multi-label dense motion segmentation [22], video over-segmentation [17], as well as binary (i.e. FG/BG) video object segmentation methods [21, 23]. On multi-label segmentation, we outperform [17],[3], and [22] in F-measure. The improvement from the latter is not great; however, note that unlike theirs, our method is causal and has a small memory footprint. We are not the best in terms of "number of extracted objects". As mentioned before, unless the object undergoes sufficient motion, it will not be detected. On FG/BG segmentation, we outperform [21],[23], and [3].

Video segmentation. BVSD [29, 16] contains 40 training and 60 testing sequences, each up to 121 frames. Pixel-wise ground truth annotation is provided for a subset of frames. Video sequences are in HD; we resize images to 540×960. While we report results for a variety of algorithms [10, 1, 15, 34] (with data from [16]), our primary point of comparison is [22]. Performance is benchmarked using "boundary precision-recall" (BPR) and "volume precision-recall" (VPR) metrics. BPR is commonly used in image segmentation, while VPR quantifies the spatiotemporal overlap between machine-generated and ground-truth segmentations (see [16] for details).

Video *object* segmentation algorithms are expected to be in the high-precision regime in BPR, and in the high-recall regime in VPR, which indeed both we and [22] satisfy (see Fig. 7). We obtain $(P, R, F) = (0.760, 0.186, 0.299)$ and $(0.136, 0.870, 0.234)$ on BPR and VPR respectively, while they obtain $(0.566, 0.100, 0.170)$ and $(0.146, 0.852, 0.249)$. Sample results are in Fig. 9. Note that the ground truth is often fine-grained—with objects spanned by multiple regions. Thus, on this benchmark, object segmentation methods will not obtain the best F-measure.

Timing. Given optical flow (which video segmentation often requires as input), our algorithm takes 30s for VGA images on a standard desktop; most of the time is spent solving (11), but a GPU implementation can reduce this.

7. Discussion

Occlusion relations inform the partition of the image domain into segments, but proper inference of such relations requires knowledge of the segments in turn. Rather than tackling an intractable chicken-and-egg problem, we use priors informed by Gestalt principles to arrive at a convex optimization scheme that can be efficiently solved with primal-dual methods. To compare with existing benchmarks, we converted our layers into "objects" and into "foreground/background". The evaluation highlights strengths and limitations of our method, with some of the latter due to the particular characteristics of the benchmarks. While our scheme still relies on decent optical flow and occlusion detection to bootstrap layer segmentation, it is less prone to cascading failure than previous methods, as it better exploits priors on motion, appearance, and layer consistency.

Acknowledgements. We would like to thank Virginia Estellers for many useful suggestions and discussions on optimization. This work was supported by NSF RI-1422669, ONR N00014-13-1-034, FA8650-11-1-7156.

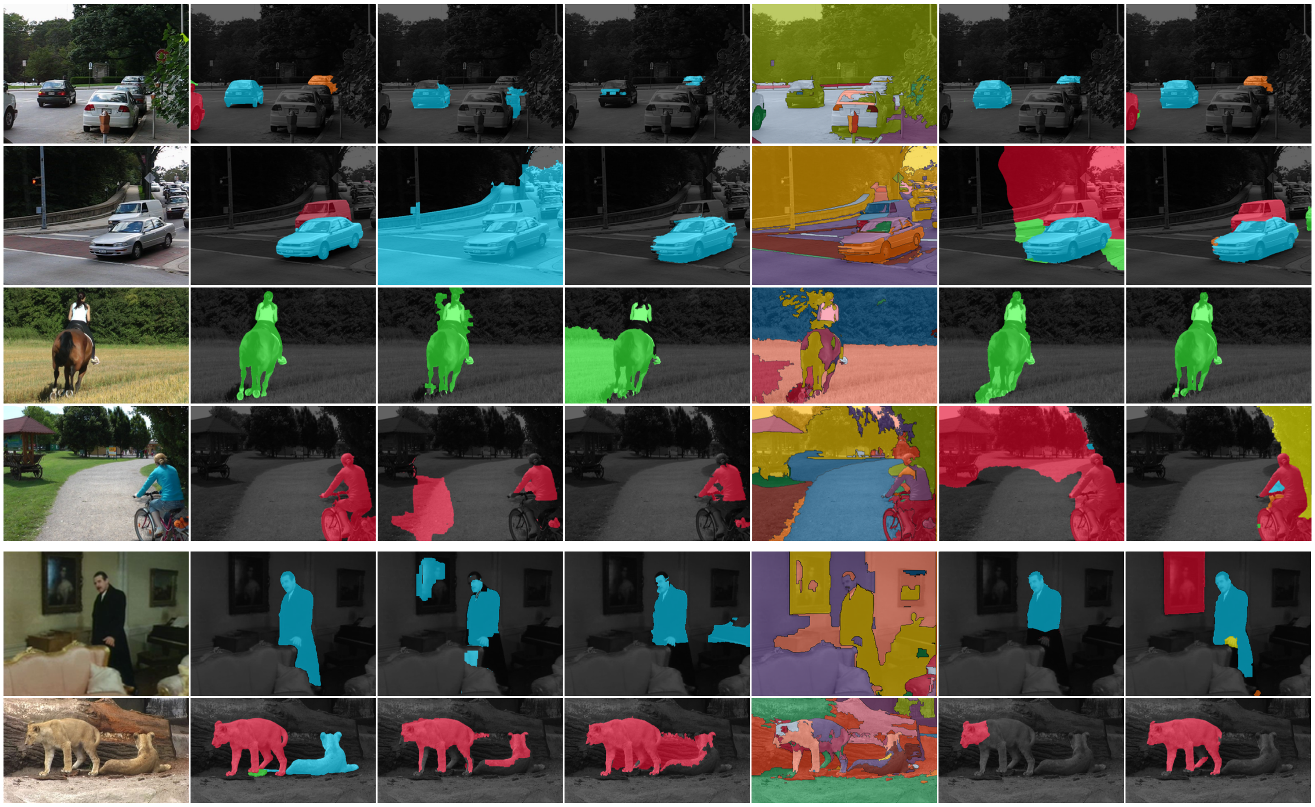


Figure 8: *Sample results on MoSeg.* Left to right: *original image, ground truth, [21], [23], [17], [22], ours (object maps). Top two rows: occlusion cues allow us to obtain even the barely visible cars (row 1 - orange, row 2 - red, row 2 - green). Row 3: use of both motion and appearance cues allows us to generate an accurate object boundary. Row 4: occlusion cues yield three depth layers (bicyclist, tree, background) (see also Fig. 1). Notice that the tree (and some cars in rows 1–2) is not annotated, so our scheme is penalized despite providing the correct answer. [21, 23] suffer from trailing and only produce binary segmentation. [17] suffers from oversegmentation. [22] performs comparably to our method; The last two rows show failure cases. Row 5: the painting is recognized as an "object" due to false occlusion detection; the hand is assigned to a separate layer. Row 6: the lioness is missed due to insufficient motion and lack of occlusions.*

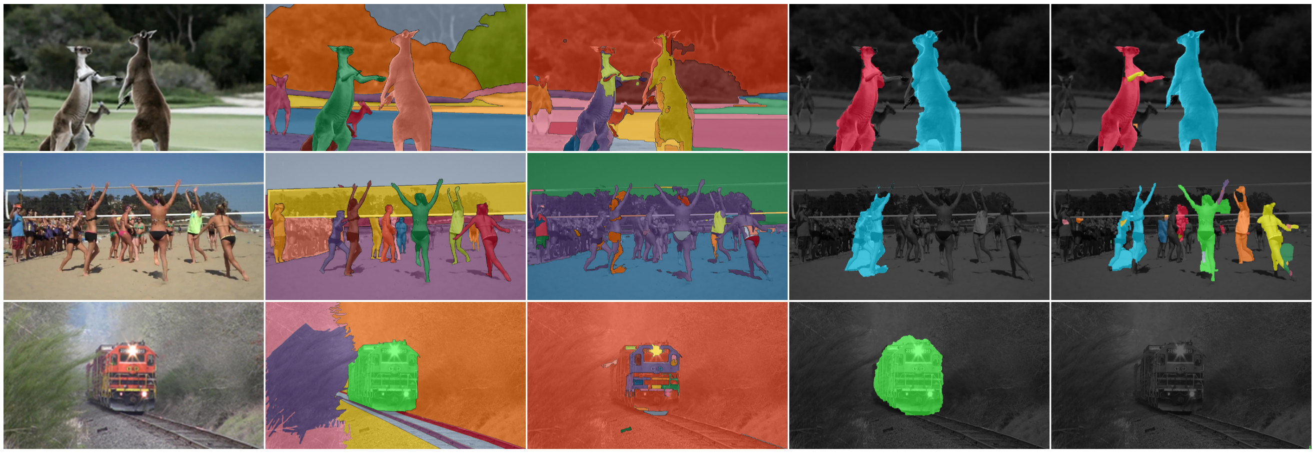


Figure 9: *Sample results on BVSD.* Left to right: *original image, ground truth, [17],[22], ours (object maps). Row 1: as reflected by BPR, our method produces accurate boundaries (see Fig. 7). Both actors are correctly segmented—the arm occluding the animal's body is a distinct depth layer. Row 2: "failure case"—complex motion and inaccurate flow can result in inaccurate segmentations. Row 3: failure case—object is not detected throughout the sequence due to lack of occlusions.*

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5), 2011.

[2] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *International Journal of Computer Vision*, 97(3), May 2012.

[3] A. Ayvaci and S. Soatto. Detachable object detection: Segmentation and depth ordering from short-baseline video. *PAMI*, 34(10), 2012.

[4] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: robust video object cutout using localized classifiers. In *ACM Transactions on Graphics (TOG)*, 2009.

[5] L. Bergen and F. Meyer. Motion segmentation and depth ordering based on morphological segmentation. In *ECCV*, 1998.

[6] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *ICCV*, 2009.

[7] G. J. Brostow and I. Essa. Motion based decompositing of video. In *ICCV*, 1999.

[8] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1), 2011.

[9] J. Chang and J. W. Fisher III. Topology-constrained layered tracking with latent flow. In *ICCV*, 2013.

[10] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille. Efficient multilevel brain tumor segmentation with integrated bayesian model classification. *IEEE Transactions on Medical Imaging*, 27(5), 2008.

[11] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *CVPR*, 2006.

[12] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *IEEE Workshop on Visual Motion*, 1991.

[13] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[14] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, 23(3), 2004.

[15] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *ACCV*, 2012.

[16] F. Galasso, S. Naveen, T. J. Cardenas, T. Brox, and B. Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013.

[17] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph based video segmentation. In *CVPR*, 2010.

[18] J. Jackson, A. J. Yezzi, and S. Soatto. Dynamic shape and appearance modeling via moving and deforming layers. *IJCV*, 2008.

[19] N. Jojic and B. J. Frey. Learning flexible sprites in video layers. In *CVPR*, 2001.

[20] M. P. Kumar, P. H. Torr, and A. Zisserman. Learning layered motion segmentations of video. *IJCV*, 76(3), 2008.

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

[22] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 36(6), 2014.

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

[24] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, 2007.

[25] T. Schoenemann and D. Cremers. A coding-cost framework for super-resolution motion layer decomposition. *TIP*, 2012.

[26] P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *TPAMI*, 26(4), 2004.

[27] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

[28] D. Sun, E. Sudderth, and M. Black. Layered segmentation and optical flow estimation over time. In *CVPR*, 2012.

[29] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011.

[30] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistent occlusion relations. Technical Report 150002, UCLA, CSD, 2015.

[31] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*, 2010.

[32] J. Y. Wang and E. H. Adelson. Representing moving images with layers. *TIP*, 3(5), 1994.

[33] M. Wertheimer. *Laws of organization in perceptual forms*. W. D. Ellis, 1939.

[34] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012.

[35] Y. Yang and G. Sundaramoorthi. Modeling self-occlusions in dynamic shape and appearance tracking. In *ICCV*, 2013.

[36] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013.