Detachable Object Detection:

Segmentation and Depth Ordering From Short-Baseline Video

Alper Ayvaci Stefano Soatto

September 5, 2011

Abstract

We describe an approach for segmenting an image into regions that correspond to surfaces in the scene that are partially surrounded by the medium. It integrates both appearance and motion statistics into a cost functional, that is seeded with occluded regions and minimized efficiently by solving a linear programming problem. Where a short observation time is insufficient to determine whether the object is detachable, the results of the minimization can be used to seed a more costly optimization based on a longer sequence of video data. The result is an entirely unsupervised scheme to detect and segment an arbitrary and unknown number of objects. We test our scheme to highlight the potential, as well as limitations, of our approach.

1 Introduction

A "detached object" was defined by Gibson [17] as *"a layout of surfaces completely surrounded by the medium."* He argued that a "topologically closed surface can be moved without breaking its surface." This property is functionally important as it gives objects *"typical affordances like graspability"*. Unfortunately, unless they are floating in midair, most objects are attached to something. Absent the ability to actively intervene by attempting to grasp an object, the most we can determine from passive imaging data is whether it is *partially* surrounded by the medium. Therefore, we call *"detachable object"* a *(compact and simply-connected) subset of the domain of an image that back-projects onto a layout of surfaces that is partially surrounded by the medium*. These include protruding objects resting on the ground plane or hanging (Fig. 2), but not flat pictures (Fig. 1-right). Detachable objects are defined in the *image*, rather than the *scene*, since we want to detect them without having to explicitly reconstruct a spatial layout of surfaces. However, they correspond to regions of space that *may* be detached, given sufficient force (Fig. 1-left). A direct

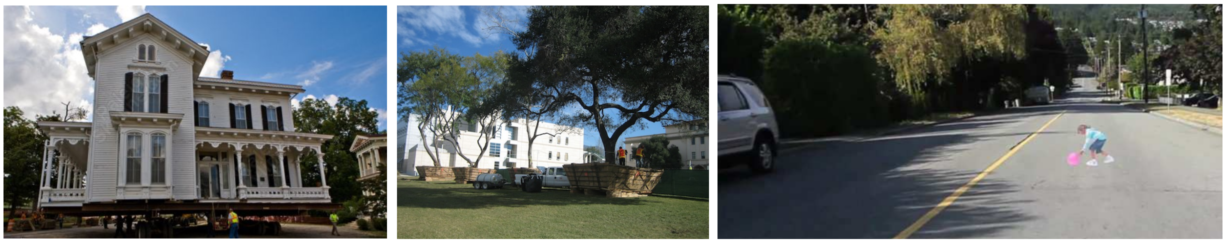

Figure 1: *Although our definition of "detachable" can be inconsistent with the dictionary notion of "something that can be detached" the two may be closer than they appear. Houses and trees satisfy both definitions, as these images illustrate. The girl figure painted on the road, on the other hand, fails both definitions: It can be erased, but not detached without breaking its surfaces. While it is in contact with the medium (air), it is not partially surrounded by it.*

consequence of the definition is that detachable objects yield *occlusion phenomena* in response to either object or viewer *motion.* A single image is not sufficient to determine whether an object is detachable. At least *two* are necessary (three, in our approach), and more are beneficial.

In this paper, therefore, we detect detachable objects in two stages: First, we detect *occlusion regions.* They provide local (and sparse) evidence of detachable objects, as well as *local* depth ordering constraints. In a second stage, these constraints are integrated into a partition of the entire image into different depth layers (Sect. 2).

The first stage requires motion. While in some cases two adjacent video frames may be sufficient to generate occlusion regions at the interface with the medium, there is typically no motion discontinuity at the support region where the object is attached (e.g., the right foot in Fig. 2-left). Photometric statistics (color, texture, intensity) may provide evidence of the boundary of the object, but in general extended observation is needed to determine whether the contact region changes, and therefore the object is detachable. For instance, the right foot in Fig. 2-right, is eventually lifted, the bench in Fig. 6 (top row) is seen against a varying background during the short sequence, and a moving car changes its point of contact with the ground, so they are all detachable, even though at no point are they actually detached.

The partition of the overall detection task into two stages sacrifices end-to-end optimality. However, it comes with a considerable benefit: The first stage is known to admit a convex relaxation [5]; we show that the second reduces to a *linear programming* problem. The **key idea** of our approach is to use occlusion regions as “seeds” in a supervised segmentation scheme, so the overall system is entirely unsupervised, and does not require any manual labeling or annotation. We also perform model selection, that is the determination of the number of detachable objects, *all by solving a linear program.* What we pay in end-to-end optimality we gain in computational efficiency. If optimality and long-term temporal integration is a concern, we can always use our results to seed a global variational optimization scheme [24].

1.1 Related work

This paper relates to a vast literature on video-based segmentation, for instance [24, 32, 14, 22, 7, 38]. Such approaches work well with few independently moving objects and require a reasonable initialization that is increasingly difficult as the scene becomes more cluttered. Our method does not require knowledge of the number of objects, nor any user input, initial bounding boxes, or scribbles [40, 7]. Instead, occluded regions, detected with [5], provide the “supervision mechanism” to seed to our method. For this reason, we refer to “short-baseline video” in our title, although the method is agnostic to whether one has short- or wide-baseline motion, video or unordered snapshots, ego-motion or object motion, so long as they yield occlusion evidence. In principle we could forgo a two-step approach and directly segment a video by processing it in a batch as in [22, 12, 10]; however, we find that the algorithmic benefits outweigh the loss of end-to-end optimality [39].

Occlusions have of course been used before as a cue for layering: [11], however, assumes a static camera; [28] classifies three kinds of occlusions to prime motion segmentation; [15] perform local analysis using spatio-temporal filters to estimate depth ordering. Other methods for layered motion segmentation [23, 25, 29, 35] also take occlusions into account, but use more restrictive parametric motion models, and do not scale well as the number of object increases beyond few. Similarly, [37, 3] use occlusion boundaries inferred using appearance, motion and depth cues [36, 31, 21] or T-junctions [2] to segment image sequences; however, they require the number of segments to be known *a priori.* Our work is rather different in spirit from those attempting to obtain a depth map from a single image [27, 1].

We capitalize on efficient *supervised segmentation* algorithms that, starting from labeled seeds (“scribbles” for background/foreground [18, 9]), produce a segmentation by solving a linear program. However, we *use the occluded regions as “local scribbles” in a supervised segmentation scheme, Fig. 2.* This evidence is globally integrated into a graph partitioning algorithm that can be solved by linear programming. It assigns each pixel to a depth ordering label that provides a putative segmentation of detachable objects, even in the presence of *multiple objects* that otherwise cause standard segmentation schemes to lose convexity [13]. This idea is formalized in the next section where we introduce our optimization scheme.

In our implementation, for computational convenience we solve our linear program on a superpixel graph, rather than the image lattice. However, this step is not conceptually necessary, and can be foregone.

2 From local occlusion ordering to global consistency

Let $I : D \subset \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}^+;\ (x,t) \mapsto I_t(x)$ be a grayscale time-varying image sequence defined on a domain D. Under the assumption of Lambertian reflection, constant illumination and co-visibility, $I_t(x)$ is related to its (forward and backward) neighbors $I_{t+dt}(x)$, $I_{t-dt}(x)$ by the usual brightness-constancy equation

$$I_t(x) = I_{t\pm dt}(x + v_{\pm t}(x)) + n_\pm(x), \quad x \in D\backslash\Omega_\pm(t) \tag{1}$$

where v_{+t} and v_{-t} are the forward and backward motion fields and the additive residual lumps together all unmodeled phenomena. *In the co-visible regions*, such a residual is typically small (in some norm) and spatially and temporally uncorrelated. However, in the presence of parallax motion, there generally are regions in the current image that are not visible in the forward (backward) neighbor, $\Omega_+(t)$ ($\Omega_-(t)$). In these regions, one cannot find a motion field v_{+t} (v_{-t}) that maps the image onto its neighbors. Therefore, the residual is typically large and not uncorrelated (unless the occluded region has identical statistics of the occluder, in which case we cannot tell that there has been an occlusion in the first place). One can use this observation to simultaneously determine the motion fields, $v_{\pm t}(x)$ and the occluded regions $\Omega_\pm(t)$ by solving a convex optimization problem [5]. From now on, therefore, we assume to be given, at each time t, the forward (occlusion) and backward (un-occlusion) time-varying regions $\Omega_+(t)$, $\Omega_-(t)$, possibly with errors, and drop the subscript ± for simplicity. The local complement of Ω, i.e. a subset of $D\backslash\Omega$ in a neighborhood of Ω, is indicated by Ω^c and can be obtained by using morphological dilation operators, or simply by duplicating the occluded region on the opposite side of the occlusion boundary (Fig. 2).

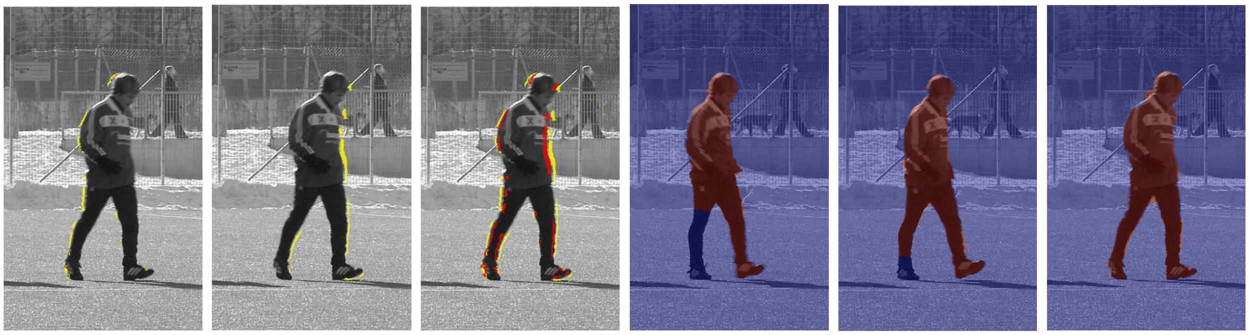

Figure 2: *Left to right: $\Omega_-(t)$ (yellow); $\Omega_+(t)$ (yellow); Ω (yellow) and Ω^c (red) on the 168th frame of the* Soccer *sequence [8]. Segmentation based on short-baseline motion does not allow determining whether the right foot and leg are detachable; however, extended temporal observation enables eventually to associate the entire leg with the body, and therefore detecting the person as a whole detachable object.*

It is important to note that these regions are in general *multiply-connected*, so $\Omega = \cup_{k=1}^K \Omega_k$, and each connected component Ω_k may correspond to a different occluded region. However, occlusion detection is a *binary classification* problem because each region of an image is either *co-visible* (visible in a temporally adjacent image) or not, regardless of how many detachable objects populate the scene. In order to detect the *(multiple)* detachable objects, we must aggregate local depth-ordering information into a global depth-ordering model. To this end, we define a *label field* $c : D \times \mathbb{R}^+ \to \mathbb{Z}^+; x \mapsto c(x,t)$ that maps each pixel x at time t to an integer indicating the depth order, $c(x,t)$. For each connected component k of an occluded region Ω, we have that if $x \in \Omega_k$ and $y \in \Omega_k^c$, then $c(x,t) > c(y,t)$ (larger values of c correspond to objects that are closer to the viewer). If x and y belong to the same object, then $c(x,t) = c(y,t)$. To enforce label consistency within each object, we therefore want to minimize $|c(x,t) - c(y,t)|$, but we want to integrate this constraint against a data-dependent measure that allows it to be violated across object boundaries. Such a measure, $d\mu(x,y)$, depends on both motion and texture statistics, for instance, for the simplest case of grayscale statistics, we have $d\mu(x,y) = W(x,y)dxdy$ where

$$W(x,y) = \begin{cases} \alpha e^{-(I_t(x)-I_t(y))^2} + \beta e^{-\|v_t(x)-v_t(y)\|_2^2} \\ 0 \quad \text{otherwise}; \qquad\qquad \|x-y\|_2 < \epsilon, \end{cases} \tag{2}$$

where ϵ identifies the neighborhood, α and β are the coefficients that weight the intensity and motion components of the measure. We then have

$$\begin{aligned} \hat{c} = \arg \min_{c:D\to\mathbb{Z}} \int_D |c(x,t) - c(y,t)| d\mu(x,y) \\ \text{s. t. } c(x,t) < c(y,t) \ \forall\ x \in \Omega_k(t), y \in \Omega_k^c(t),\ k = 1,..,K, \end{aligned} \tag{3}$$

and $\|x-y\|_2 < \epsilon$. This problem would be solved trivially by a constant, e.g., $c(x) = 0$, if it were not for the boundary conditions imposed by occlusions.

To translate this into a linear program, we quantize D into an $M \times N$ grid-graph $G = (V,E)$ with the vertex (node) set V (pixels or super-pixels), and the edge set $E \subseteq V \times V$ denoting adjacency of two nodes $i,j \in V$ via $i \sim j$. We then identify i,j with x_i, x_j, their corresponding depth ordering $c_i = c(x_i,t),\ c_j = c(x_j,t)$, and the measure $d\mu(x_i,x_j)$ is a symmetric positive-definite matrix $w_{ij} = W(x_i,x_j)$ that measures the *affinity* between two nodes i,j. The problem (3) then becomes the search for the discrete-valued function $c : V \to \mathbb{Z}^+$

$$\begin{aligned} \{\hat{c}_i\}_{i=1}^{MN} = \arg\min_c \quad \sum_{i\sim j} w_{ij}|c_i - c_j| \\ \text{s. t.} \quad c_i < c_j, \quad i \sim j,\ i \in \Omega_k(t),\ j \in \Omega_k^c(t), \end{aligned} \tag{4}$$

with $1 \leq c_i \leq L$. In the case of $L = 2$, this problem can be interpreted as binary graph cut [34]. Unfortunately, for $L > 2$ this is an NP-hard problem so, as customary, we relax it by dropping the integer constraint and allowing $c : V \to \mathbb{R}^+$.

3 Automatic model selection

A natural criterion for model selection is to trade off model complexity with data fidelity, as customary in *minimum-description length* (MDL) [20]. In our case, an obvious complexity cost is the number of objects, that is the largest value taken by the *label field*, $\|c\|_\infty \doteq \max\{|c_i|\}_{i=1}^{MN}$. This leads to the straightforward modification of the problem (4) into

$$\begin{aligned} \{\hat{c}_i\}_{i=1}^{MN} = \arg\min_c \quad \sum_{i\sim j} w_{ij}|c_i - c_j| + \gamma\|c\|_\infty \\ \text{s. t.} \quad c_i < c_j, \quad i \sim j,\ x_i \in \Omega_k(t),\ x_j \in \Omega_k^c(t), \end{aligned} \tag{5}$$

with $1 \leq c_i$ where γ is the cost for adding a new layer. While this problem preserves the convexity properties of the original model, it is not amenable to being solved using linear programming (LP). Therefore, we introduce auxiliary variables $\{u_{ij} | i \sim j\}$ and σ, so that (5) can be written as

$$\begin{aligned} \min_{u_{ij},c_i,\sigma} & \sum_{i\sim j} w_{ij}u_{ij} + \gamma\sigma \\ \text{s. t.} \quad & 1 \preceq c \preceq \sigma, \\ & c_j - c_i \geq 1, \quad i \sim j,\ x_i \in \Omega_k(t),\ x_j \in \Omega_k^c(t) \\ & -u_{ij} \leq c_i - c_j \leq u_{ij}. \end{aligned} \tag{6}$$

This makes the problem amenable to deployment of a vast arsenal of efficient numerical methods. Note that we have relaxed the integer constraint by allowing the difference between label values to be *at least* one. As customary, we will quantize the label map after solving the optimization problem by rounding its values to the nearest integer.

4 Seeding extended temporal observations

As we have anticipated, one could wrap occlusion detection, motion estimation, and depth ordering into one large optimization problem. This can be done by combining (3) with the cost functional in [5] and summing over time through an entire video sequence. The result is equation (2) of [24], and the ensuing optimization is unwieldy. Therefore, for simplicity and modularity, we prefer to keep the two stages separate, and describe the simplest form of temporal integration, that is to use the results of (5) at each instant as initialization to the optimization at the subsequent time, using the field $v_{-(t)}$. We redefine the measure to incorporate the previous layer estimate by replacing $W(x, y)$ with

$$W(x,y) + \frac{1}{\tau} H(c(x + v_{-t}(x), t-1), c(y + v_{-t}(y), t-1)), \tag{7}$$

where $H : \mathbb{R} \times \mathbb{R} \to \{0, 1\}$ is defined by

$$H(a,b) = \begin{cases} 1, & a = b, a > 1, b > 1, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

and τ is a forgetting factor. We show in the experimental section that this simple model is sufficient to aggregate parts detachable objects in the great majority of cases.

5 Revisiting occlusion errors

Since we have decomposed the original problem into two separate stages, it is important for the second to handle the inevitable errors made by the first. To model errors in occlusion detection, we introduce slack variables $\{\xi_k\}_{k=1}^K$ to relax the hard constraints, to

$$\begin{aligned} \min_{u_{ij}, c_i} \; & \sum_{i \sim j} w_{ij} u_{ij} + \lambda \sum_{k=1}^{K} \xi_k \\ \text{s. t.} \quad & 1 \preceq c \preceq L, \\ & c_j - c_i \geq 1 - \xi_k, \quad i \sim j, \; i \in \Omega_k(t), \; j \in \Omega_k^c(t) \\ & 0 \leq \xi_k \leq 1 \; \forall \; k, \\ & -u_{ij} \leq c_i - c_j \leq u_{ij}. \end{aligned} \tag{9}$$

where λ is the penalty for violating the ordering constraints. The problem of detachable object detection is finally in a form that is suitable for a standard numerical solver, for instance [19]. The (forward-backward) occluded regions $\Omega_\pm(t)$ are given from [5]. Note that layers obtained may consist of multiple objects, so to enforce simple connectivity of a detachable objects we can isolate each connected component using standard morphological operators within each depth level on c.

6 Experiments

Rather than solving (9) on the pixel grid, we pre-compute a partition of the domain into N non-overlapping superpixels $\{s_i\}_{i=1}^N$ such that $\bigcup_{i=1}^N s_i = D, \; s_i \cap s_j = \emptyset, \; \forall i \neq j$, as done by [36], using a watershed-based approach driven by a statistical multi-cue edge detector [26]. However, since superpixels may cross occlusion boundaries, we subdivide them to ensure that each superpixel is a subset of one of the three regions: Ω, Ω^c and $D\backslash(\Omega \cup \Omega^c)$. This superpixelization is not necessary, but it reduces the size of the linear program while enabling integration of simple low-level cues. Since the minimization (9) is already written for a general graph, it does not matter whether it is performed on the pixel grid or the superpixel graph. The only change is the weight matrix w_{ij}, that in the case of superpixels is given by

$$w_{ij} = |\partial s_i \cap \partial s_j|[\alpha e^{-(\bar{I}(s_i)-\bar{I}(s_j))^2} + \beta e^{-\|\bar{v}(s_i)-\bar{v}(s_j)\|_2^2} + \kappa(1 - \overline{Pb}(s_i, s_j))], \tag{10}$$

where $\bar{I}(s) = \frac{1}{|s|}\int_s I_t(x)dx$, $\bar{v}(s) = \frac{1}{|s|}\int_s v_t(x)dx$ and

$$\overline{Pb}(s, s') = \frac{1}{\partial s \cap \partial s'} \int_{\partial s \cap \partial s'} Pb(x)dx, \tag{11}$$

and $Pb : D \to [0, 1]$ is the probability of a location being a (material, occlusion, or illumination) boundary. Note that the edge features are incorporated into the computation of the weights since the domain D is partitioned based on Pb. In our experiments, we have assigned the parameters α, β and κ to 0.25, 0.5 and 0.25 respectively.

We have used the CMU Occlusion/Object Boundary Dataset[1] [36] to evaluate our approach. It includes 16 test sequences with a variety of indoor and outdoor scenes, mostly seen under small camera motion. It provides ground truth segmentation for a single reference frame in each sequence. We also report our results on the Soccer sequence [8], to illustrate the typical effect of extended temporal observations, and on a sequence portraying a drawing of a girl playing ball near an intersection in West Vancouver (Fig. 4).

6.1 Qualitative performance

Representative examples of successful detection are shown in Fig. 3, 5 and 6. In Fig. 3, a hiker and his hand are detected as separate detachable objects. The support region (left foot) is attributed to the ground plane, as expected, since there is insufficient evidence (photometric discontinuity) to separate it from the ground. The same goes for the squirrel (Fig. 3 bottom row), and the cat in Fig. 5 (fifth row). Note that the arm in Fig. 3 appears as a detached object, since its support region (where it is attached) is outside the field of view, and therefore it appears to be floating in midair. The same goes for the chair, couch, hand, and horse in Fig. 5 (first to fourth row, respectively).

The sequence in Fig. 3, from [36], is too short to capture an entire walking cycle, so the person cannot be positively identified as detachable. Using a longer sequence, such as the Soccer scene in Fig. 2, shows that we can successfully aggregate the entire person into one segment, and therefore positively detect him as a detachable object. The sequence in Fig. 4 is taken at an intersection where a child figure is painted on the road. Unlike a real pedestrian or a car, this does not trigger occlusions, and is therefore not detected as a detachable object, unlike the nearby car.

6.2 Failure modes

Our method does not always work. Representative examples are shown in Fig. 3, where the closest box (second row) is not detected as a detachable object. This is because its support region is large and its occluding boundaries are not very salient. In order to detect it as a detachable object, one would have to see the box under sufficient parallax, for instance by moving around it, or to see it moved relative to its support base. One could also determine that the object is detachable by considering different images of the same object in different contexts, without any temporal continuity, but this would require (wide-baseline) co-segmentation [30], that is beyond the scope of this paper. The handles on the toy horse in Fig. 5 (fourth row) are also not detected because the motion is too small and there is no significant motion signal around its boundaries. Longer sequences would easily disambiguate these objects.

[1] http://www.cs.cmu.edu/∼stein/occlusion_data/

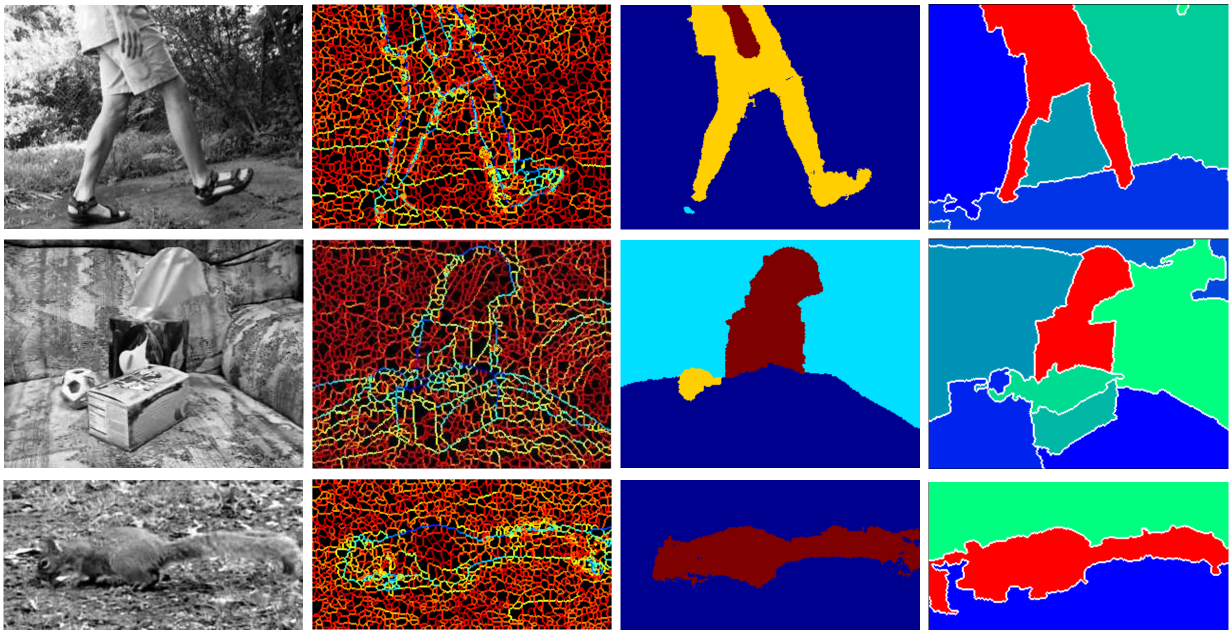

Figure 3: *Sample frames from* Walking Legs, Couch Color *and* Squirrel4 *(first column), affinities between superpixels on the reference frame (second column), output of our algorithm (third column) and the result of the segmentation method described in [37] (fourth column) (Figures at fourth column are borrowed from [37]) under Andrew Stein's authorization.'.*

Another failure mode can be seen in Fig. 5 (top row), where the inside of the bowl behind the chair is

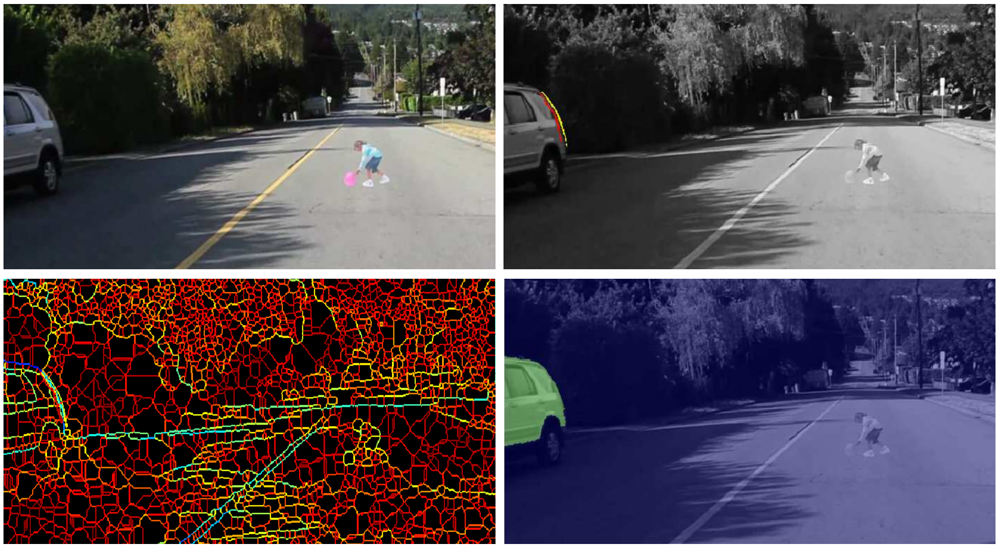

Figure 4: *A child figure is painted on the road in West Vancouver (a). Unlike a real pedestrian or a car, this drawing does not cause any occlusion (b). Therefore, given the image and motion features (c), our algorithm does not detect it as a detachable object while it segments the nearby car (d). The original sequence can be seen at http://reviews.cnet.com/8301-13746_7-20016169-48.html.*

detected as a detached object, because the water reflection violates the Lambertian assumption implicit in the occlusion detection functional.

Perhaps the most obvious failure mode is shown in the bottom line of Fig. 5, where only three of the objects (the Coffee mate container, a pen and the box in the background) are detected as detached. Again, this can be attributed to the small motion that does not elicit significant enough signal to trigger an occlusion detection.

Some of the failure modes are alleviated by automatic model selection (Fig. 7).

6.3 Quantitative assessment

Our quantitative evaluation follows the lines of [4]. The covering score of a set of ground truth segments S' by a set of segments S can be defined as

$$Score(S', S) = \frac{1}{\sum_{s' \in S'} |s'|} \sum_{s' \in S'} \max_{s \in S} \frac{|s \cap s'|}{|s| + |s'|} \tag{12}$$

Note that comparing our approach to [37] is not straightforward, and possibly unfair, since the latter is an over-segmentation method where the number of segments are predetermined; our algorithm, on the other hand, performs automatic model selection. To be fair to [37], we have selected the cases where their algorithm yields a single segment, discarding all others that would negatively bias their outcome. [37] reports segmentation covering scores of 0.72 for the pedestrian, 0.84 for the tissue box and 0.71 for the squirrel which are depicted in red in Fig 3. By comparison, our algorithm achieves scores of 0.90, 0.95 and 0.90 respectively.

We have also compared our method to normalized cut [33], as the superpixel graphs depicted at Fig. 4 can be partitioned using this technique. However, normalized cut also requires the number of segments to be known *a priori*, therefore, in our experiments, we have used self-tuning spectral clustering proposed by [41] which addresses this limitation. Our performance on the whole dataset considering all the ground truth objects is shown in Table 1, which shows that our algorithm outperforms [41] in most of the sequences.

	Bench	Car2	Chair1	Coffee Stuff	Couch Color	Couch Corner	Fencepost	Hand3
Score with model selection	0.89	0.52	0.78	0.43	0.63	0.95	0.42	0.74
Score [41]	0.67	0.52	0.66	0.40	0.40	0.93	0.42	0.65
Score with true L	0.89	0.53	0.78	0.40	0.72	0.96	0.41	0.73
Score when forcing L = 2	0.89	0.52	0.67	0.41	0.32	0.76	0.38	0.73

	Intrepid	Post	Rocking Horse	Squirrel4	Trash Can	Tree	Walking Legs	Zoe1
Score with model selection	0.85	0.98	0.78	0.90	0.75	0.69	0.92	0.72
Score [41]	0.55	0.98	0.70	0.75	0.73	0.89	0.64	0.71
Score with true L	0.66	0.98	0.77	0.91	0.75	0.74	0.91	0.72
Score when forcing L = 2	0.66	0.98	0.77	0.91	0.75	0.74	0.80	0.72

Table 1: *Performance of our approach on the CMU dataset computed based on the covering score (12) and compared to [41], [6] in case the correct number of layers is provided and [6] when L is set to* 2.

As seen in Table 1, testing with and without automatic model selection yields comparable results when the *correct* number of layers L is given. However, automatic model selection significantly improves performance when the assumed number of layers is incorrect.

In terms of running time, once occluded regions are detected, it takes 6.3 seconds for CVX [19] to solve the linear program (9) with 310 depth ordering constraints on a frame over-segmented to 4012 superpixels. The run-time for occlusion detection in [5] can be reduced to a few seconds per frame using Split Bregman, depending on the size of the images; superpixelization is not a necessary step, but recent recent work has shown that it too can be performed at a rate of a few frames per second [16].

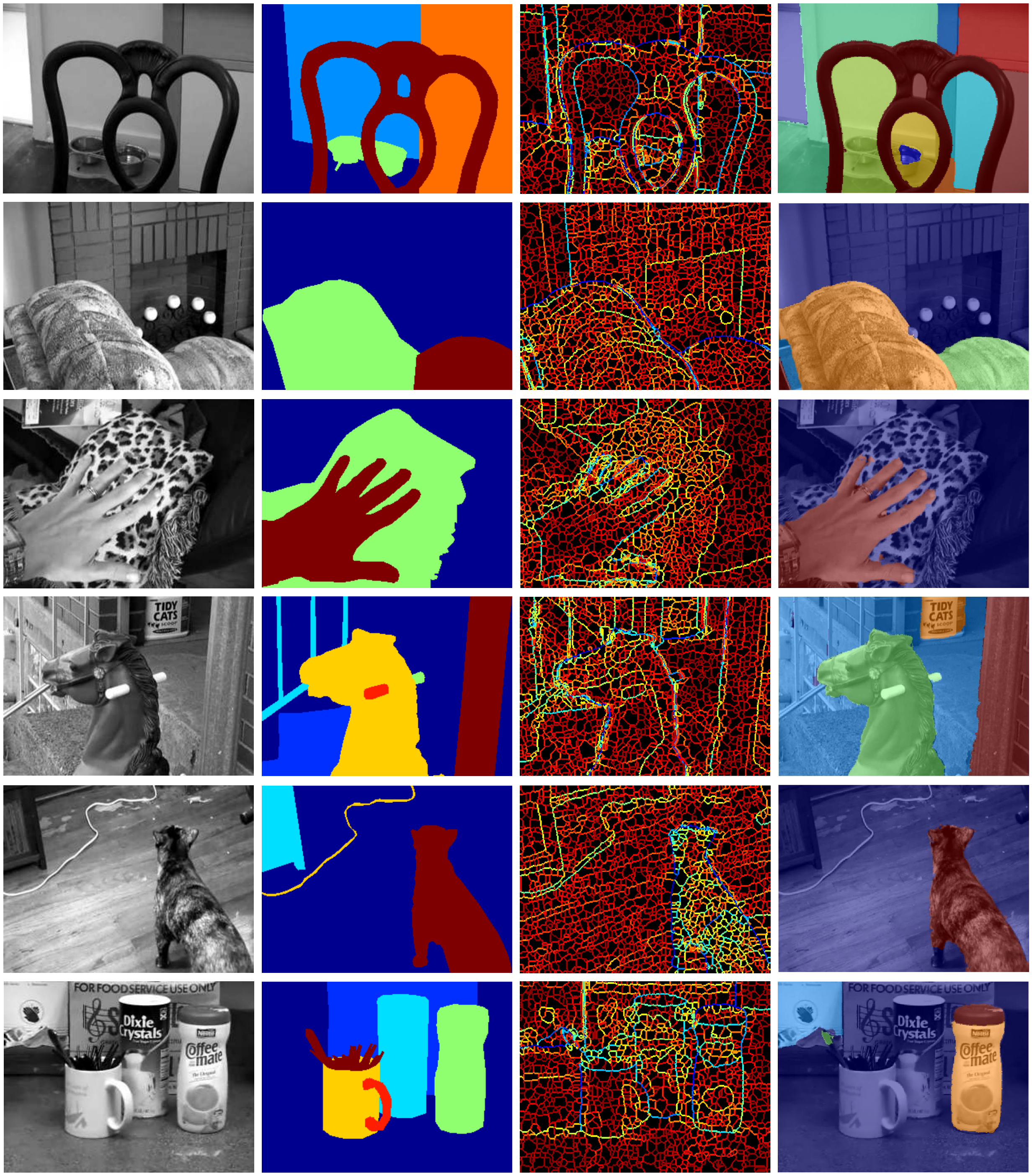

Figure 5: *Sample results from the CMU dataset (first column), ground truth objects on these sequences (second column) affinities between superpixels computed on the reference frames (third column), output of our algorithm (fourth column). Note that color coding does not represent the layers rather the distinct components on the layer map.*

7 Discussion

We have presented a method for detecting "objects" in a scene. While functionally important properties such as graspability cannot be ascertained from passive imaging data, we have defined properties that a *moving image* of an object must have in order to correlate to topological properties of the *scene*, such as being partially surrounded by the medium. We have defined these objects as *detachable*, conscious that this may be a misnomer in some cases, for instance houses and trees, despite Fig. 1.

Occlusions play a key role in the detection of detachable objects. Leveraging prior work, we show that once (binary) occlusion regions are available, integrating local ordering information into a coherent depth ordering map can be achieved by simple linear programming. The key to our approach is to convert a supervised segmentation problem into an unsupervised one using occlusions as the supervision mechanism. As a result, we have a fully unsupervised method for detecting and segmenting an unknown number of objects, and estimating their number in the meantime, all by solving a linear program.

Our method is not panacea. Despite our efforts to manage errors in the occlusion detection stage, we still suffer from complete failures of the occlusion detection mechanism. In many cases, this is due to insufficient motion in the scene, and the results improve under extended temporal observation.

Nevertheless, our results can still be useful as initialization of a more involved optimization over an extended temporal observation, that we and others have already developed.

Our approach has a few tuning parameters, but fewer than most competing schemes since we perform model selection. Of course, even model selection requires tuning the tradeoff between complexity and fidelity, and there is no "right" choice of parameters.

Our approach also shares the limitation of all schemes that break down the original problem (detached object detection, in our case) into a number of sequential steps, whereby failure of the early stages of processing cause failure of the entire pipeline. This predicament comes with the benefit of solving an otherwise very complex computational problem using efficient numerical schemes.

Acknowledgment

This work was supported by ONR N000141110863. This technical report represents a revised version of a manuscript originally submitted for review on Feb. 22, 2011, that in turn was a revision of an earlier manuscript submitted on Nov. 19, 2010, and registered on that date as Technical Report UCLA-CSD-100036.

References

[1] M. Amer, R. Raich, and S. Todorovic. Monocular Extraction of 2.1D Sketch. In *Proc. of the International Conference on Image Processing*, September 2010.

[2] N. Apostoloff and A. Fitzgibbon. Learning Spatiotemporal T-Junctions for Occlusion Detection. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, 2005.

[3] NE Apostoloff and AW Fitzgibbon. Automatic video segmentation using spatiotemporal T-junctions. In *Proc. of the Britih Machine Vision Conference*, 2006.

[4] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, 2009.

[5] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *International Journal of Computer Vision*, (in press) 2011.

[6] A. Ayvaci and S. Soatto. Detachable object detection. Technical Report CSD100036, UCLA Computer Science Department, November Nov. 19, 2010.

[7] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video SnapCut: robust video object cutout using localized classifiers. In *ACM SIGGRAPH*, 2009.

[8] S. Boltz, A. Herbulot, E. Debreuve, M. Barlaud, and G. Aubert. Motion and appearance nonparametric joint entropy for video segmentation. *International Journal of Computer Vision*, 2007.

[9] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2002.

[10] W. Brendel and S. Todorovic. Video object segmentation by tracking regions. In *Proc. of the International Conference on Computer Vision*, 2009.

[11] G.J. Brostow and I.A. Essa. Motion based decompositing of video. In *Proc. of the International Conference on Computer Vision*, 1999.

[12] T. Brox and J. Malik. Object Segmentation by Long Term Analysis of Point Trajectories. In *Proc. of the European Conference on Computer Vision*, pages 282–295, 2010.

[13] T.F. Chan and S. Esedoglu. Aspects of Total Variation Regularized L'Function Approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817, 2005.

[14] D. Cremers and S. Soatto. Motion competition: a variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005.

[15] D. Feldman and D. Weinshall. Motion segmentation and depth ordering using an occlusion detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1171–1185, 2008.

[16] B. Fulkerson and S. Soatto. Really quick shift: Image segmentation on a gpu. In *Workshop on Computer Vision using GPUs, held with the European Conference on Computer Vision*, September 2010.

[17] J. J. Gibson. *The ecological approach to visual perception.* LEA, 1984.

[18] Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[19] M. Grant and S. Boyd. Cvx: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, October 2010.

[20] P.D. Grunwald and J. Rissanen. *The Minimum Description Length Principle.* The MIT Press, 2007.

[21] X. He and A. Yuille. Occlusion Boundary Detection using Pseudo-Depth. In *Proc. of the European Conference on Computer Vision*, 2010.

[22] Y. Huang, Q. Liu, and D. Metaxas. Video object segmentation by hypergraph cut. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, pages 1738–1745, 2009.

[23] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4:324–324, 1993.

[24] J. Jackson, A. J. Yezzi, and S. Soatto. Dynamic shape and appearance modeling via moving and deforming layers. *Intl. J. of Comp. Vision*, 79(1):71–84, August 2008.

[25] A. Jepson, D. Fleet, and M. Black. A layered motion representation with occlusion and compact spatial support. In *Proc. of the European Conference on Computer Vision*, pages 692–706, 2002.

[26] D.R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[27] J.M. Morel and P. Salembier. Monocular Depth by Nonlinear Diffusion. In *Proc. of the Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.

[28] A.S. Ogale, C. Ferm, and Y. Aloimonos. Motion segmentation using occlusions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 988–992, 2005.

[29] M. Pawan Kumar, P.H.S. Torr, and A. Zisserman. Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76(3):301–319, 2008.

[30] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, 2006.

[31] ME Sargin, L. Bertelli, BS Manjunath, and K. Rose. Probabilistic Occlusion Boundary Detection on Spatio-Temporal Lattices. In *Proc. of the International Conference on Computer Vision*, 2009.

[32] T. Schoenemann and D. Cremers. High resolution motion layer decomposition using dual-space graph cuts. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, 2008.

[33] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2002.

[34] Ali Kemal Sinop and Leo Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Proc. of the International Conference on Computer Vision*, 2007.

[35] P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):479–494, 2004.

[36] A.N. Stein and M. Hebert. Occlusion boundaries from motion: low-level detection and mid-level reasoning. *International Journal of Computer Vision*, 82(3):325–357, 2009.

[37] Andrew Stein, Thomas Stepleton, and Martial Hebert. Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, June 2008.

[38] M. Unger, T. Mauthner, T. Pock, and H. Bischof. Tracking as segmentation of spatial-temporal volumes by anisotropic weighted TV. In *Proc of the Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2009.

[39] Amelio Vazquez-Reina, Shai Avidan, Hanspeter Pfister, and Eric Miller. Multiple hypothesis video segmentation from superpixel flows. In *Proc. of the European Conference on Computer Vision*, 2010.

[40] J. Wang, Y. Xu, H.Y. Shum, and M.F. Cohen. Video tooning. In *ACM SIGGRAPH*, 2004.

[41] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, 2004.

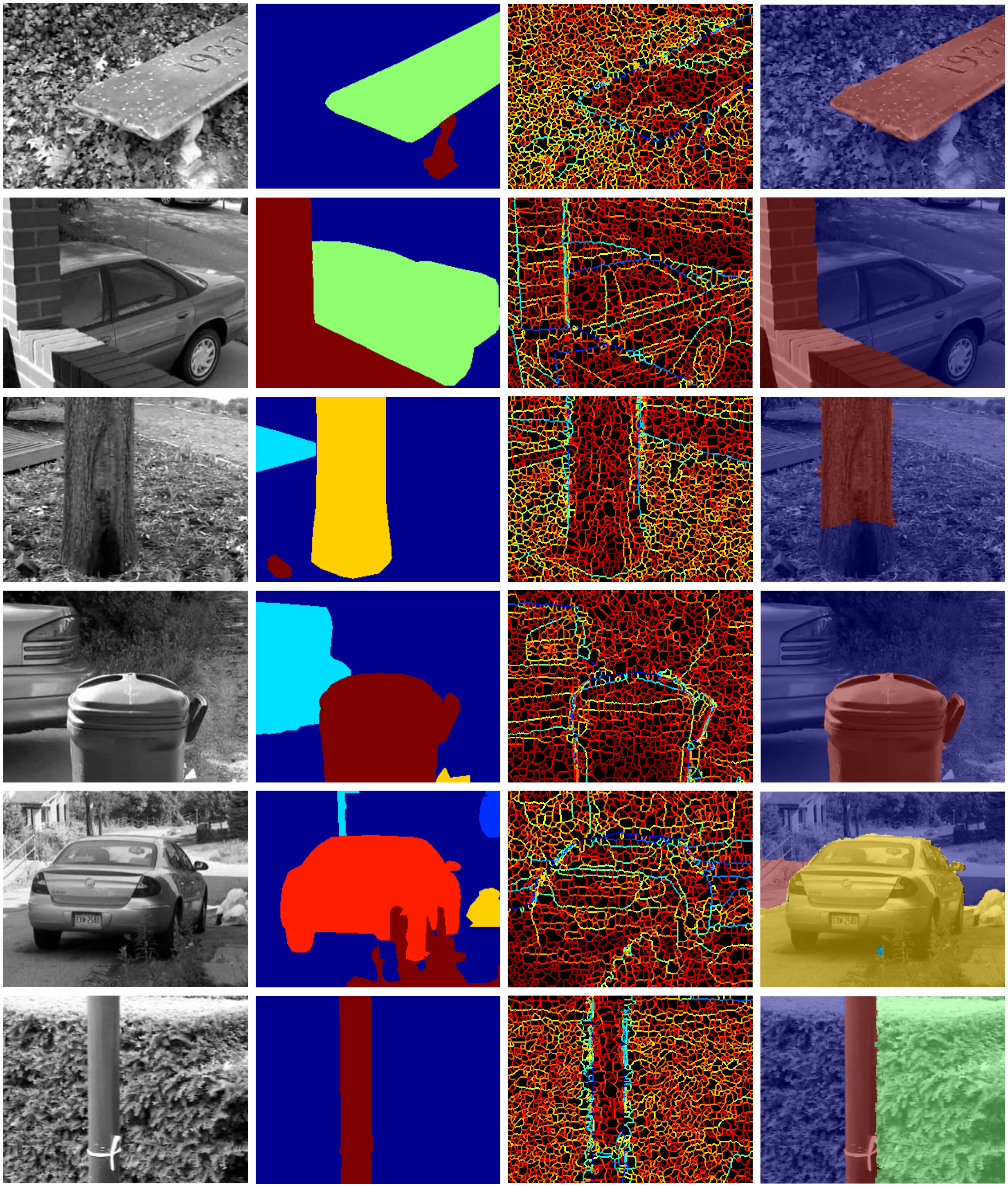

Figure 6: *Additional samples from the CMU dataset (first column), ground truth objects on these sequences (second column) affinities between superpixels computed on the reference frames (third column), output of our algorithm (fourth column). Note that color coding does not represent the layers rather the distinct components on the layer map. Failures are related to small motion and miss detection of occluded regions.*

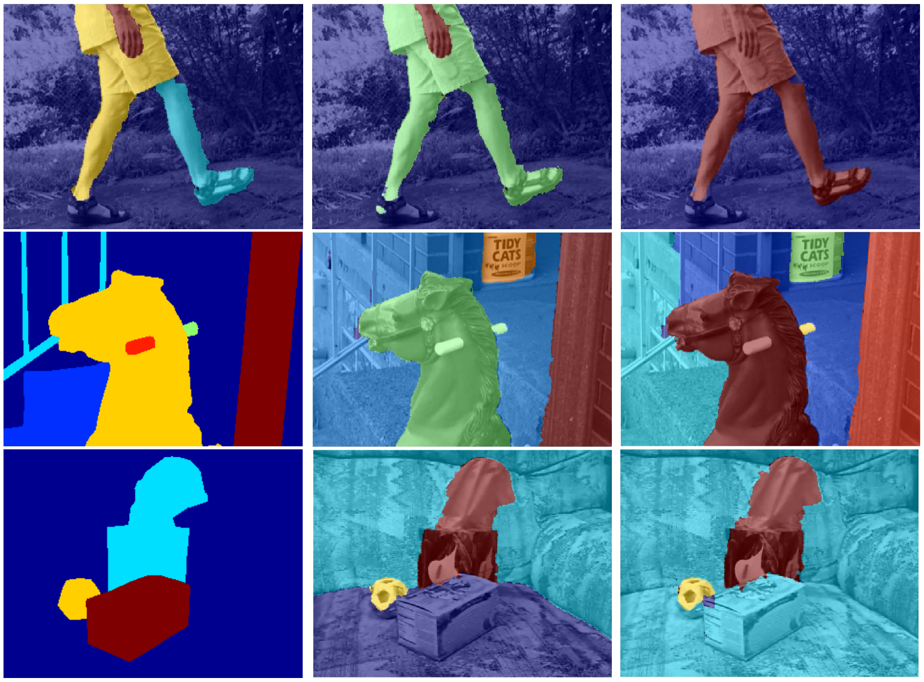

Figure 7: *Top: Effects of increasing layer cost γ on model selection. From left to right, the number of regions $\hat{\sigma}$ is estimated as 4 (background, hand-arm, body and pivot leg, swinging leg), 3 (background, body, and swinging arm) and 2 (whole body). Note that the pivot foot is attached to the ground, and is therefore classified as such. Middle: Effects of model selection on representative samples from the CMU dataset. Ground truth is on the left, segmentation with manual setting of $L = 4$ is in the middle, and segmentation with automatic model selection is on the right. Allowing automatic model selection enables the detection of the horse handles as separate detachable objects. On the bottom row, the algorithm fails to detect the closer cleenex box as a detachable object, because of insufficient parallax, so the box is either lumped with the horizontal cushion, or with the entire couch.*