Layered Segmentation and Optical Flow Estimation Over Time

Deqing Sun[1] Erik B. Sudderth[1] Michael J. Black[1,2]
[1]Department of Computer Science, Brown University, Providence, RI 02912, USA
[2]Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany

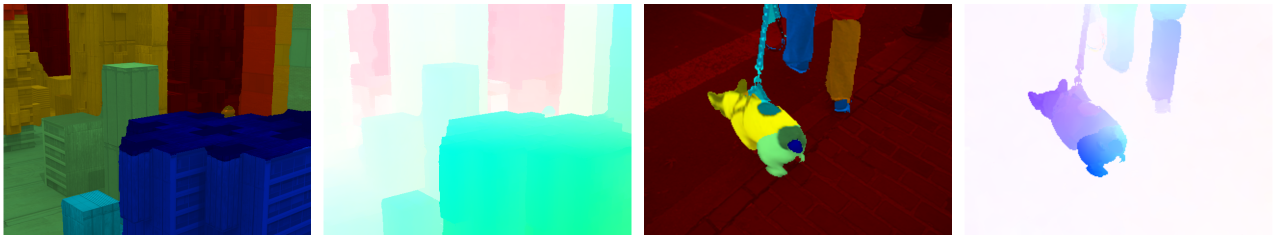


Figure 1. The proposed method segments scenes into layers (left in each pair) and estimates the flow (right) over several frames.

Abstract

Layered models provide a compelling approach for estimating image motion and segmenting moving scenes. Previous methods, however, have failed to capture the structure of complex scenes, provide precise object boundaries, effectively estimate the number of layers in a scene, or robustly determine the depth order of the layers. Furthermore, previous methods have focused on optical flow between pairs of frames rather than longer sequences. We show that image sequences with more frames are needed to resolve ambiguities in depth ordering at occlusion boundaries; temporal layer constancy makes this feasible. Our generative model of image sequences is rich but difficult to optimize with traditional gradient descent methods. We propose a novel discrete approximation of the continuous objective in terms of a sequence of depth-ordered MRFs and extend graph-cut optimization methods with new "moves" that make joint layer segmentation and motion estimation feasible. Our optimizer, which mixes discrete and continuous optimization, automatically determines the number of layers and reasons about their depth ordering. We demonstrate the value of layered models, our optimization strategy, and the use of more than two frames on both the Middlebury optical flow benchmark and the MIT layer segmentation benchmark.

1. Introduction

The segmentation of scenes into regions of coherent structure and the estimation of image motion are fundamental problems in computer vision which are often treated separately. When available, motion provides an important cue for identifying the surfaces in a scene and for differentiating image texture from physical structure. This paper addresses the principled combination of motion segmentation and static scene segmentation. We do so by introducing a new layered model of moving scenes in which the layer segmentations enable the integration of motion over time. This results in improved optical flow estimates, disambiguation of local depth orderings, and correct interpretation of occlusion boundaries. To solve the challenging inference problem we introduce a new mixed continuous and discrete optimization method that solves for the number of layers and their depth ordering. The resulting method achieves state-of-the-art results in both video segmentation and optical flow estimation (Figure 1).

Layered models offer an elegant approach to motion segmentation and have many advantages. A typical scene consists of very few moving objects and representing each moving object by a layer allows the motion of each layer to be described more simply [31]. Such a representation can explicitly model the occlusion relationships between layers making the detection of occlusion boundaries possible. Unfortunately, current layered motion models have not shown convincing layer segmentation results on challenging real-world sequences.

One key issue is that the layer-structure inference problem is difficult to optimize. Most methods adopt an expectation maximization (EM) style algorithm that is susceptible to local optima. For example, in [27] we propose a generative layered model that combines mixture models with state-of-the-art static image segmentation models [25]. This **Layers++** method estimates image motion very accurately as measured by the Middlebury optical flow benchmark [1].

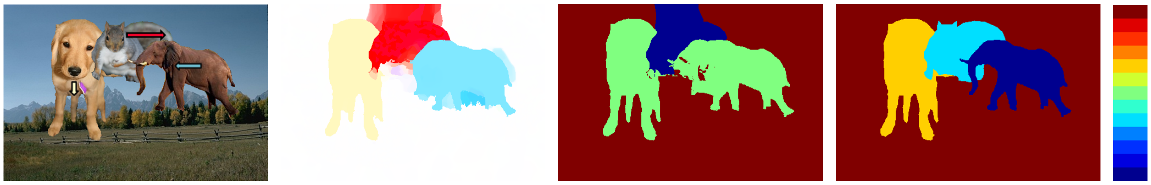


Figure 2. A failure case for the **Layers++** method [27]. Left to right: first image in a pair (arrows show motion direction and their length indicates motion magnitude); initial flow estimate, color coded as in [1]; segmentation by **Layers++**; segmentation with our proposed **nLayers** method, which automatically determines the number of layers, their depth ordering, and is able to make large changes to the initial flow field to reach a good solution. On the far right is a color key for the ordering of depth layers (blue is close and red is far).

However, our gradient-based inference algorithm is susceptible to local optima, resulting in errors in the estimated scene structure and flow field, as illustrated in Figure 2.

Overcoming such limitations requires an optimization method that can make large changes to the solution at a single step, a task more suitable for discrete optimization. Hence we propose a discrete layered model based on a sequence of ordered Markov random fields (MRFs). This model, unlike standard Ising/Potts MRFs, cannot be directly solved by "off-the-shelf" optimizers, such as graph cuts. Therefore we develop a sequence of non-standard moves that can simultaneously change the states of several binary MRFs. We also embed continuous flow estimation into the discrete framework to adapt the state space to estimate subpixel motion. The resultant discrete-continuous scheme enables us to infer the number of layers and their depth ordering automatically for a sequence.

We evaluate our layer segmentation using the MIT human-assisted motion annotation dataset [18]. Our method produces semantically more meaningful segmentations that are also quantitatively more consistent with human labeled ground truth than the continuous-only **Layers++** method. With a reliable layer segmentation and the relative depth ordering obtained with the discrete method, we initialize the more precise **Layers++** continuous model of optical flow. The discrete-continuous approach gives a concrete improvement over a purely continuous optimization that can easily become trapped in local optima.

Like many approaches, our previous work [27] considers optical flow between only two frames. Unfortunately, with only two frames, depth ordering at occlusion boundaries is fundamentally ambiguous [7]. Critically, our approach is formulated to estimate optical flow over time. By estimating layer segmentations over three or more frames we obtain a reliable depth ordering of the layers and more accurate motion estimates. At the time of writing, the proposed method is ranked first in AAE and fourth in EPE on the Middlebury optical flow benchmark.

In summary, our contributions include *a)* formulating a discrete layered model based on a sequence of ordered Ising MRFs and devising a set of non-standard moves to optimize it; *b)* formulating methods for automatically determining the number of layers and their depth ordering for a given sequence; *c)* concretely improving layer segmentation on a set of real-world sequences; *d)* demonstrating the benefits of using more frames for optical flow estimation on the Middlebury optical flow benchmark.

2. Previous Work

Layered optical flow. Most layered approaches assume a parametric motion for each layer [11, 13, 16, 31] which is too restrictive to capture the motion of natural scenes. Weiss [32] addresses this by allowing smooth motions in the layers. In [27] we extend this to impose global coherence via an affine motion field while modeling local non-smooth deformation from affine with a robust MRF. Both methods adopt continuous optimization methods that do not reason about the number or ordering of layers.

Like us, Jepson *et al.* [12] decompose a scene into overlapping layers, reasoning about the number of layers, and determining the depth order. While their method models layer support with parametric regions we allow much more varied layers. Weiss and Adelson [33] incorporate static image cues into the layered segmentation and estimate the number of layers under fairly weak assumptions. Torr *et al.* [28] use a Bayesian decision making framework to determine the number of approximately planar layers but do not infer the depth ordering. Our depth ordering formulation is similar to flexible sprites [13] and their extensions [16], but we use more flexible motion models.

Occlusion. Reasoning about occlusion in image sequences dates to the mid 1970's and early 1980's; a full review is beyond our scope. Early authors (*e.g.* [21]) note that occlusion boundaries move with the occluding surface but the first explicit statement that this requires three frames to compute seems to be by Darrell and Fleet [7]. We illustrate this in Figure 3 because we can not find a clear description in the literature.

This simple fact is a key reason why two-frame optical flow estimation is fundamentally limited. In a layered model, inferring the wrong depth order results in significant errors at motion boundaries. The idea of using three or more frames has been embodied in recent methods for computing motion boundaries and depth order [3, 8] but appears missing from recent dense flow estimation methods.

Estimating flow over time. Again, estimation of flow

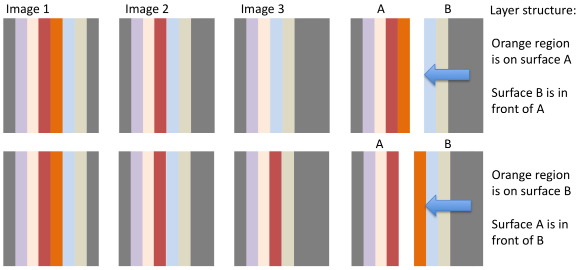


Image 1
Image 2
Image 3
A
B
Layer structure:
Orange region is on surface A
Surface B is in front of A
A
B
Orange region is on surface B
Surface A is in front of B

Figure 3. Relative depth ordering can be ambiguous in 2 frames. A third frame enables the motion of the occlusion boundary to be computed. This motion is consistent with the occluding surface, removing the ambiguity. Two cases are shown where image 1 and 2 are the same. In both, surface B moves to the left. With image 3 the ambiguity is resolved because the motion of the occluding contour is known.

over time has a long history [2, 20] yet few methods have demonstrated improved accuracy through temporal consistency of flow. Volz *et al*. [29] recently propose a multi-frame optical flow method that shows improvement using 5 frames over their 2-frame baseline. However their 5-frame method is still less accurate than the top performing 2-frame methods [27, 37]. We argue that, while flow fields are not always temporally consistent, the scene structure represented by a layered segmentation is.

Segmentation and optimization. Ising/Potts MRFs are popular for layer segmentation [35, 36] because of the availability of efficient optimizers. Unfortunately they assign low probability to typical segmentations of natural scenes [19] and do not capture the relationship between layers, such as occlusion. Sudderth and Jordan [25] develop a segmentation model based on thresholded Gaussian processes (similar to level set methods) and obtain realistic segmentations of static scenes. In [27] we exploit this model for image sequences using continuous support functions. Here we formulate a novel discrete version using a sequence of ordered Ising MRFs and develop non-standard moves to optimize it. The motion competition framework [5, 6] uses level sets to model the scene segmentation in a variational setting, but does not address occlusion reasoning.

It is common to alternate optimization between segmentation and motion estimation [23, 27, 32]. However, previous methods change the flow and segmentation separately, while we argue that they must be coupled. An object may appear in the wrong layer but with the correct motion. Consequently one must change the motion and layer segmentation simultaneously to avoid local optima.

Discrete optimization techniques, such as belief propagation [15] and graph cuts [4, 14], have been used in single-layer robust optical flow formulations [17, 24]. Particularly, Lempitsky *et al*. [17] fuse a large set of candidate flow fields to minimize a robust energy function. These methods can reach good local optima but tend to produce large errors in occlusion regions. This can partly be remedied by explicit occlusion detection and post processing [37]. In contrast, we exploit the layered model to explicitly model the occlusion process during continuous flow refinement. Graph cuts have also been used for segmentation and tracking. For example Kumar *et al*. [16] alternate the optimization of motion and segmentation and use affine motion models for each layer. Additionally, Wang *et al*. [30] require the manual segmentation of objects in the first frame and use parametric motion models.

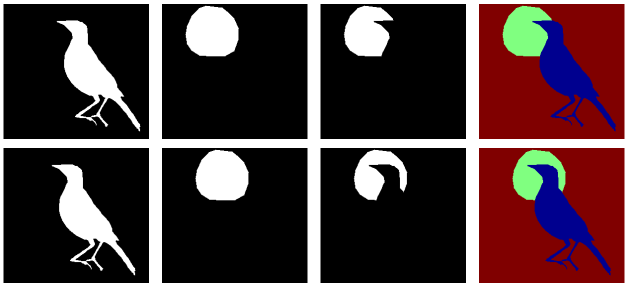


Figure 4. Left to right: support functions for the first (front) and second layers, visibility mask for the second layer, and the layer segmentation. Top: frame t; bottom: frame $t + 1$. The "bird" layer is in front of and occludes the second "apple" layer; Bottom row: The binary support functions at time $t + 1$ are temporally consistent with those at time t according to the flow field for each layer, resulting in temporally consistent layer segmentation. The two support functions jointly determine the layer segmentation.

The power of layered models is as much about segmentation as motion estimation, and we thus compare to a contemporary graph-based [9] video segmentation method. A complete review of video segmentation is beyond our scope.

3. Models and Inference

We first define a discrete generative layered model based on an ordered sequence of binary, Ising MRFs. We then introduce a family of "cooperative" discrete optimization moves, as well as methods to determine the number of layers and their depth ordering.

3.1. A Discrete Layered Model for Optical Flow

Consider an image sequence presumed to have K depth-ordered motion layers. In our previous work, we model the spatial support of these layers by thresholding a sequence of smooth, continuous layer support functions [27]. Unfortunately, our previous gradient-based inference scheme is susceptible to local optima, especially in determining the scene segmentation and depth ordering.

In contrast to gradient-based methods, discrete optimization methods like graph cuts [4, 14] can substantially change the configuration of the solution in a single move. To exploit such methods we need a discrete (approximate) formulation of the layered flow estimation problem. The

$$E(\mathbf{u},\mathbf{v},\mathbf{g},\theta) = \sum_{t=1}^{T-1} \Bigg\{ E_{\text{data}}(\mathbf{u}_t,\mathbf{v}_t,\mathbf{g}_t,\mathbf{g}_{t+1}) + \sum_{k=1}^{K} \lambda_a \big(E^{\text{flow}}_{\text{space}}(\mathbf{u}_{tk},\theta_{tk}) + E^{\text{flow}}_{\text{space}}(\mathbf{v}_{tk},\theta_{tk})\big)$$
$$+ \sum_{k=1}^{K-1} \lambda_b E^{\text{sup}}_{\text{space}}(\mathbf{g}_{tk}) + \lambda_c E_{\text{time}}(\mathbf{g}_{tk},\mathbf{g}_{t+1,k},\mathbf{u}_{tk},\mathbf{v}_{tk}) \Bigg\} + \sum_{k=1}^{K-1} \lambda_b E^{\text{sup}}_{\text{space}}(\mathbf{g}_{Tk}). \qquad (1)$$

overall energy function is given by Eq. (1) in which $\mathbf{u}_{tk}, \mathbf{v}_{tk}$ are flow fields for each of the K layers at time t, and $\mathbf{g}_{tk}$ are *binary* support functions for the first $K-1$ layers (in contrast to the smooth support functions in [27]). We also associate every layer with an affine motion field $\mathbf{u}_{\theta_{tk}}, \mathbf{v}_{\theta_{tk}}$ parameterized by θ_{tk}; we motivate this choice shortly.

As shown in Figure 4, we can determine binary visibility masks $\mathbf{s}_{tk}$ for each layer from $\mathbf{g}_{tk}$ by sequentially multiplying the support functions and their complements.

$$s^p_{tk} = \begin{cases} g^p_{tk} \prod_{k'=1}^{k-1}(1-g^p_{tk'}), & 1 \le k < K \\ \prod_{k'=1}^{K-1}(1-g^p_{tk'}), & k = K, \end{cases} \qquad (2)$$

where $p = (i,j)$ denotes a pixel at frame t. The visibility masks provide a segmentation of the scene into layers. Given the visibility mask, the occlusion reasoning (data likelihood term) is the same as in [27] $E_{\text{data}}(\mathbf{u}_t,\mathbf{v}_t,\mathbf{g}_t,\mathbf{g}_{t+1}) =$

$$\sum_{k=1}^{K}\sum_{p}\Big(\rho_d(\mathbf{I}^p_t - \mathbf{I}^q_{t+1}) - \lambda_d\Big) s^p_{tk} s^q_{t+1,k}, \qquad (3)$$

where $q = (i + u^p_{tk}, j + v^p_{tk})$ denotes the corresponding pixel at frame $t+1$, and ρ is a robust penalty function. Temporal consistency of the support functions, as aligned by the inferred flow field, is encouraged by an Ising MRF:

$$E_{\text{time}}(\mathbf{g}_{tk},\mathbf{g}_{t+1,k},\mathbf{u}_{tk},\mathbf{v}_{tk}) = \sum_p (1-\delta(g^p_{tk}, g^q_{t+1,k})).(4)$$

For non-integer flow vectors, sub-pixel interpolation introduces high-order temporal terms. We round these flow vectors to obtain an approximation with only pairwise terms. As shown in Figure 4, these temporally consistent support functions ensure the layer structures persist over time.

We capture the spatial coherence of the binary support functions by a conditional Ising MRF with weights determined by image color differences:

$$E^{\text{sup}}_{\text{space}}(\mathbf{g}_{tk}) = \sum_p \sum_{q\in\mathcal{N}_p} w^p_q (1-\delta(g^p_{tk}, g^q_{tk})). \qquad (5)$$

Here the weight is defined as in the continuous formulation [27]. These spatially coherent support functions ensure the scene segmentation is spatially coherent and respects the local image evidence. Note that the visibility term in Eq. (2) implies high-order interaction terms among several layer-specific spatio-temporal Ising MRFs.

We model the motion of each layer by a pairwise MRF with a unary term. The energy term is $E^{\text{flow}}_{\text{space}}(\mathbf{u}_{tk},\theta_{tk}) =$

$$\sum_p \sum_{q\in\mathcal{N}_p} \rho_{\text{mrf}}\,(u^p_{tk} - u^q_{tk}) + \lambda_{\text{aff}} \sum_p \rho_{\text{aff}}(u^p_{tk} - u^p_{\theta_{tk}}), \quad (6)$$

where the unary term encourages the flow field of each layer to be close to its affine flow (with weight λ_{aff}), and the affine motion is computed as in [27]. Note that this semiparametric model still allows deviation from the affine motion and is more flexible than parametric models. In automatically determining the number of layers, there is an important balance between Equations (5) and (6): the former penalizes support discontinuities, while the latter favors additional layers so that each layer's flow is closer to affine.

3.2. "Cooperative" Discrete Optimization Moves

Optimization of Equation (1) is challenging. A common strategy is to alternate the optimization of the support functions and the flow fields for each individual layer. Unfortunately, this approach is susceptible to local optima (see Figure 2). We thus develop optimization moves that can simultaneously change the flow fields and segmentation.

The standard moves of graph cuts are not directly applicable to the discrete model, because of the high-order interaction terms in the data term. We therefore define a set of "cooperative" moves that can *a)* change a group of pixels to be visible at a particular layer while also selecting their flow fields; *b)* change a group of pixels to be visible at a particular layer; *c)* select the flow fields of a particular layer from a candidate set. Each move solves a binary problem via the QPBO algorithm [10, 14], where the auxiliary binary variable, $\mathbf{b}$, encodes the states of several model variables. Next we explain the most complicated simultaneous segmentation and flow move and provide the details of other moves in the **Supplemental Material**.

Simultaneous segmentation and flow move. Sometimes a region may be assigned to a wrong layer with the correct motion. To escape this local optimum, we typically need to simultaneously change the segmentation and flow fields.

Consider a pixel p in frame t, for which layer k' is currently visible. We define a binary decision variable b^p_t such

$$\sum_{t=1}^{T-1}\sum_{p}\left(\sum_{q\in\mathcal{N}^0_{p,\text{time}}}\phi^0_{\text{time}}(b^p_t,b^q_{t+1})+\sum_{q\in\mathcal{N}^1_{p,\text{time}}}\phi^1_{\text{time}}(b^p_t,b^q_{t+1})\right)+\sum_{t=1}^{T}\sum_{p}\left[\sum_{q\in\mathcal{N}_{p,\text{space}}}\phi_{\text{space}}(b^p_t,b^q_t)+\phi_{\text{affine}}(b^p_t)\right]. \quad (7)$$

that the configuration is unchanged when $b^p_t = 0$, and an alternative layer $\hat{k}$ becomes visible when $b^p_t = 1$. Revealing this new layer may alter all the support functions for the first $\hat{k}$ layers: when $b^p_t = 1$, $g^p_{t\hat{k}}(1) = 1$ and $g^p_{tk}(1) = 0, k < \hat{k}$. In this case, we also set the motion for layer $\hat{k}$ to that of the formerly visible layer ($u^p_{t\hat{k}}(1) = u^{p,\text{old}}_{tk'}$), and the motion for layer k' to its affine mean ($u^p_{tk'}(1) = u^p_{\theta_{tk'}}$).

This segmentation and flow move involves many terms from the overall model, which depend on the binary decision variables as summarized in Eq. (7). The choice of b^p_t influences the flow vector at pixel p, and thus determines which of two candidate pixels it is linked to at the next frame. When $b^p_t = 0$, the temporal neighbors are

$$\mathcal{N}^0_{p,\text{time}} = \{(i + [u^p_{tk}(0)], j + [v^p_{tk}(0)]), 1 \le k \le K\}, \quad (8)$$

and the corresponding potential function will only “fire” when $b^p_t = 0$, so that $\phi^0_{\text{time}}(b^p_t, b^q_{t+1}) =$

$$\Big[\Big(\rho_d(\mathbf{I}^p_t - \mathbf{I}^{q'}_{t+1}) - \lambda_d\Big)s^p_{tk}(b^p_t)s^q_{t+1,k}(b^q_{t+1}) \quad (9)$$
$$+ \lambda_c(1 - \delta(g^p_{tk}(b^p_t), g^q_{t+1,k}(b^q_{t+1})))(1 - \delta(k, K))\Big](1 - b^p_t),$$

which incorporates both the data and temporal terms for the $K - 1$ support functions. We evaluate the warped image $I^{q'}_{t+1}$ at subpixel positions and the visibility mask $s^q_{t+1,k}$ and the warped support function $g^q_{t+1,k}$ at integer positions. Similarly $\phi^1_{\text{time}}(b^p_t, b^q_{t+1})$ is defined to only “fire” when $b^p_t = 1$ (see **Supplemental Material**).

For the spatial term, the set $\mathcal{N}_{p,\text{space}}$ contains the four nearest neighbors of pixel p. The binary selection variable changes the states of several binary support functions and flow fields. The effects sum together as

$$\phi_{\text{space}}(b^p_t, b^q_t) = \sum_{k=1}^{K-1}\lambda_b w^p_q\big(1 - \delta(g^p_{tk}(b^p_t), g^q_{tk}(b^q_t))\big)$$
$$+ \sum_{k=1}^{K}\lambda_a\rho_{\text{mrf}}\big(u^p_{tk}(b^p_t) - u^q_{tk}(b^q_t)\big). \quad (10)$$

The unary term can be obtained from Eq. (6) as

$$\phi_{\text{affine}}(b^p_t) = \sum_{k=1}^{K}\lambda_a\lambda_{\text{aff}}\rho_{\text{aff}}(u^p_{tk}(b^p_t) - u^p_{\theta_{tk}}). \quad (11)$$

Visibility move. Given the current flow estimate, we decide whether to make a pixel p visible for the selected layer $\hat{k}$ by modifying the previous layer support $\mathbf{g}^{\text{old}}$. When $b^p_t = 0$, all the support functions retain their previous value at p, *i.e.* $g^p_{tk}(0) = g^{p,\text{old}}_{tk}$. When $b^p_t = 1$, we need to adjust the support functions of the first $\hat{k}$ layers so that layer $\hat{k}$ is visible at p, *i.e.* $g^p_{tk}(1) = 0$ if $k < \hat{k}$, and $g^p_{tk}(1) = 1$ if $k = \hat{k}$. When $\hat{k}$ is the last layer, all the support functions of the first $K - 1$ layers are set to be 0 at p.

Flow selection move and continuous refinement. Given the current layer assignment, the pixel of each layer can retain its current motion or take the motion of the affine mean flow field. This is a binary segmentation problem similar to the FusionFlow [17] work. The main difference is that our model uses the segmentation information to handle occlusions. After this step, we refine the output flow field by continuous optimization [26] to adaptively change the candidate flow fields for the discrete optimization.

3.3. Layer Number Determination and Depth Order Reasoning

We initialize with an upper bound on the number of layers. During optimization, when a layer has no visible pixels associated with it, we remove it from the solution. The new solution can equally explain the image data, pays no penalty for the removed layer, and so has lower energy. Inferring the depth ordering of layers requires testing all the possible combinations and is computationally prohibitive. We instead use heuristics to reduce the search space. We first order the layers from fast to slow by their average motion. We then perform the moves above to estimate the support functions and the flow fields in both the fast-to-slow and the slow-to-fast ordering. The ordering with the lower energy is further refined as follows. For each pair of neighboring layers, we propose to switch their ordering, and optimize their visibility mask and support functions. If the new solution has a lower energy than its previous one, we accept this new depth ordering and proceed to other pairs (see **Supplemental Material** for the detailed algorithm). In practice, we find that this local greedy search scheme is fairly robust.

4. Experimental Results

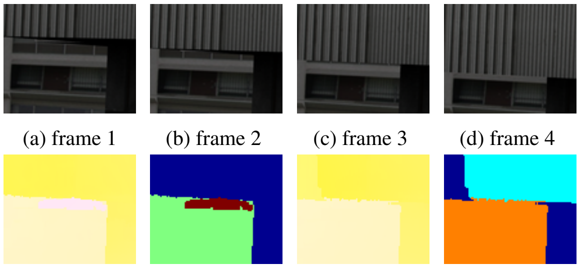
We evaluate the proposed layered model on both motion estimation and layer segmentation tasks. Throughout this section, the proposed method is called **nLayers**, since it can automatically determine the number of layers. **Layers++** refers to the continuous method developed in [27] which uses a fixed number of 3 layers. For layer segmentation, we also compare our method to a state-of-the-art, hierarchical graph-based video segmentation algorithm [9], referred to

Table 1. Average end-point error (EPE) on the Middlebury *training* set. Using four frames and the new optimization improves accuracy.

	Avg.	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3
Classic+NL (2 frames)	0.221	0.238	0.131	0.152	0.073	0.103	0.468	0.220	0.384
Layers++ (2 frames)	0.195	0.211	0.150	0.161	0.067	0.086	0.331	0.210	0.345
Layers++ (4 frames)	0.190	0.211	0.151	0.157	0.067	0.084	0.330	0.207	0.311
nLayers (4 frames)	0.183	0.191	0.126	0.175	0.062	0.080	0.336	0.175	0.316

Table 2. Average end-point error (EPE) and angular error (AAE) on the Middlebury optical flow benchmark *test* set. The discrete-continuous optimization (**nLayers**) obtains similar EPE and better AAE than the continuous-only inference method (**Layers++**).

		Rank	**Avg.**	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
EPE	**Layers++**	8.0	0.27	0.08	0.19	0.20	0.13	0.48	0.47	0.15	0.46
	nLayers	8.5	0.28	0.07	0.22	0.25	0.15	0.53	0.44	0.13	0.47
AAE	**Layers++**	9.2	2.56	3.11	2.43	2.43	2.13	2.35	3.81	2.74	1.45
	nLayers	5.7	2.38	2.80	2.71	2.61	2.30	2.30	2.62	2.29	1.38



(a) frame 1 (b) frame 2 (c) frame 3 (d) frame 4

(e) flow (2 to 3) (f) seg. (2) (g) flow (2 to 3) (h) seg. (2)

Figure 5. Occlusion reasoning using frames 2 and 3 (e-f) is hard (detail from Urban3); enforcing temporal coherence of the support functions using 4 frames significantly reduces the errors in both the flow field and the segmentation (g-h). The flow field is from frame 2 to frame 3 and the segmentation is for frame 2.

as **HGVS** in the comparison below. **HGVS** uses the output from a recent optical flow estimation method [34].

Implementation details and parameter settings. We start with the single-layered output from "Classic+NL" [26] and cluster the flow field into 10 layers. We then run the discrete method to estimate the scene structure and the flow fields to initialize the more precise continuous layered model. It takes **nLayers** about 10 hours in total to compute three forward and three backward flow fields from the four-frame 640 × 480 "Urban" sequence in MATLAB with a C++ mexed QPBO solver. It takes **Layers++** about 5 hours to compute one forward and one backward flow field from two frames. **HGVS** uses ten frames, or all the frames if a sequence has fewer than ten frames. **HGVS** has three different outputs for the same video. We show the segmentation results produced at 90 percent of highest hierarchy level, because it gives the best visual and numeric results.

4.1. Motion Estimation

We use the Middlebury optical flow benchmark to evaluate the motion estimation results. We manually set $\lambda_{\text{aff}} = 0.3, \lambda_{\text{b}} = 80$, and $\lambda_c = 10$ for the discrete model, use the provided values for the other parameters from [27], and fix them for all the motion estimation experiments. We set all the robust functions to be the generalized Charbonnier penalty function $\rho(x) = (x^2 + \epsilon^2)^a$ with $\epsilon = 0.001$ and $a = 0.45$ [26].

Results on the Middlebury *training* set are shown in Table 1. Changing from 2 to 4 frames improves results for the **Layers++** model supporting our hypothesis that longer sequences are important. More improvement comes from using a discrete model to obtain a good segmentation of the scene and then use the inferred structure for flow estimation (**nLayers**, 4 frames). Figure 5 shows a case where using 4 frames resolves ambiguity in the layer assignment and reduces errors in the estimated motion.

On the *test* set, **nLayers** obtains EPE similar to **Layers++** but better AAE, as shown in Table 2. At the time of writing (April 2012), **nLayers** is ranked first in AAE and fourth in EPE (see **Supplemental Material** for the screen shot). AAE measures the angle between the estimated motion vector and the ground truth and EPE is the Euclidean difference between the two. The results suggest that **nLayers** estimates motion directions more accurately.

Figure 6 shows the estimated segmentation and flow fields on some test sequences. Nearly all the major structures of "Urban" are correctly recovered, resulting in the best boundary EPE and AAE performance. The higher overall error results from the bottom left building. A major part of the building moves out of the image boundary and has no data term to estimate the motion. **nLayers** uses the affine model to interpolate the motion of the out-of-boundary pixels, but the building's motion violates the affine assumption.

4.2. Layer Segmentation

The Middlebury dataset does not have motion segmentation ground truth and so we use the MIT human annotated dataset [18] to evaluate segmentation performance. Segmentation accuracy is computed using the RandIndex measure [22] (larger is better). Because the MIT dataset is different in nature from the Middlebury dataset and has more rigidly moving, distant objects, we use a larger weight on the affine unary term as $\lambda_{\text{aff}} = 1$, and $\lambda_c = 3$ for the dis-

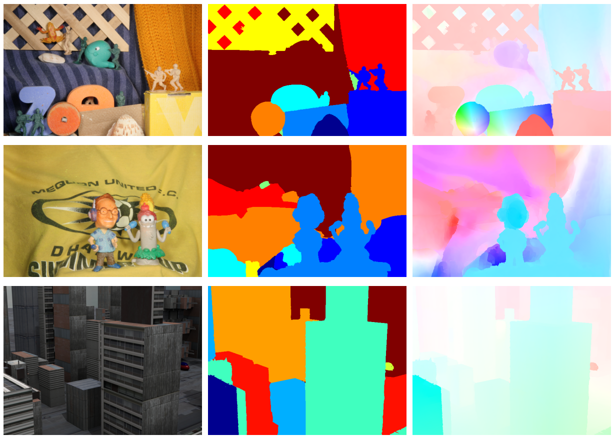


Figure 6. Estimated flow fields and scene structure on some Middlebury test sequences. Top: the proposed **nLayers** method separates the claw of the frog from the background and recovers the fine flow structure. Middle: **nLayers** separates the foreground figures of "Mequon" from the background and produces sharp motion boundaries; However, the non-rigid motion of the cloth violates the layered assumption and causes errors in estimated flow fields. Bottom: by correctly recovering the scene structure, **nLayers** achieves the lowest motion boundary errors on the "Urban" sequence; a part of the building in the bottom left corner moves out of the image boundary and its motion is predicted by the affine model. However the building's motion violates the affine assumption, resulting in errors in the estimated motion.

crete model while keeping the other parameters unchanged.

Table 3 summarizes the RandIndex measure on all the 9 sequences. On several sequences, **nLayers** (default: 10 layers and 4 frames) outperforms **Layers++** by a large margin. A bootstrap significance test is used between the **nLayers** and other methods. Small p values suggest that the improvement by **nLayers** is significant. **nLayers** with the maximum number of layers being 8 or 12 produces results similar to the baseline 10-layer model suggesting the method is not highly sensitive to the maximum number of layers. **nLayers** with only 2 frames is more accurate than the 2-frame **Layers++** method, demonstrating the benefits of discrete optimization. The improvement with 4 frames over 2 frames shows the benefits of using more frames to recover scene structure. Also note that the performance of **Layers++** drops with the number of layers used because its local inference scheme has to deal with more local optima as the number of layers increases.

Figure 7 shows some segmentation results. On "table", the layer segmentation by **nLayers** roughly matches the structure of the scene and is close to the human labeled ground truth. Although **HGVS** uses optical flow, it tends to merge foreground objects and background when their appearance is similar. **nLayers** tends to fail when motion cues are weak, such as the "phone" sequence.

5. Conclusions and Future Work

We have formulated a discrete layered model based on a sequence of ordered Ising MRFs and developed non-standard moves to optimize the model. In particular, our moves can simultaneously change the layer assignment together with the flow field, which helps avoid local optima common to schemes that alternate between optimizing flow and segmentation. The discrete optimizer enables us to adapt the number of layers to each sequence and decide their depth ordering automatically. Our method produces meaningful segmentations on the Middlebury and the MIT datasets, and achieves better quantitative results w.r.t. the human labeled ground truth than a corresponding continuous model. Our flow estimation results show the benefits of using more frames and discrete optimization to resolve depth-ordering ambiguities. Our work advances the state of the art in layered motion modeling and suggests that layered models can provide a rich and flexible representation of complex scenes.

Acknowledgments. This work was supported in part by a National Science Foundation CRCNS program grant IIS–0904875 and a gift from Intel Corp.

References

[1] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, Mar. 2011.

[2] M. Black and P. Anandan. A model for the detection of motion over time. *ICCV*, pp. 33–37, 1990.

[3] M. Black and D. Fleet. Probabilistic detection and tracking of motion boundaries. *IJCV*, 38(3):231–245, July 2000.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, Nov. 2001.

[5] T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. *ECCV*, v. I, pp. 471–483, 2006.

[6] D. Cremers and S. Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *IJCV*, 62(3):249–265, May 2005.

[7] T. Darrell and D. Fleet. Second-order method for occlusion relationships in motion layers. Technical report, MIT, 1995.

[8] D. Feldman and D. Weinshall. Motion segmentation and depth ordering using an occlusion detector. *PAMI*, 30(7):1171–1185, July 2008.

[9] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. *CVPR*, pp. 2141–2148, 2010.

[10] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0-1 optimization. *Math. Prog.*, 28:121–155, 1984.

[11] A. Jepson and M. J. Black. Mixture models for optical flow computation. *CVPR*, pp. 760–761, 1993.

[12] A. Jepson, D. Fleet, and M. Black. A layered motion representation with occlusion and compact spatial support. *ECCV*, v. I, pp. 692–706, 2002.

[13] N. Jojic and B. Frey. Learning flexible sprites in video layers. *CVPR*, v. I, pp. 199–206, 2001.

[14] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts - A review. *PAMI*, 7:1274–1279, July 2007.

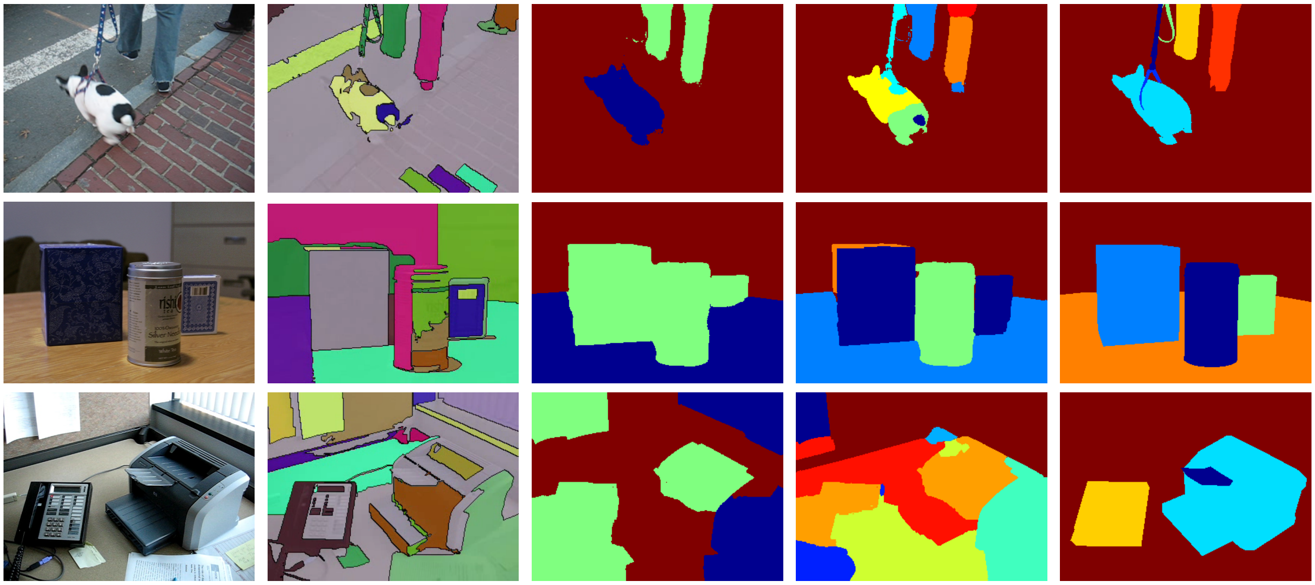


Figure 7. **MIT dataset**. Left to right first frame, segmentation results by **HGVS** [9], **Layers++** [27], **nLayers**, and human labeled ground truth. Top to bottom: “dog”, “table”, and “phone”. On “dog” and “table”, the segmentation results by **nLayers** are close to the human labeled ground truth. Motion cues in the “phone” sequence are weak, causing problem to **nLayers**. See **Supplemental Material** for more.

Table 3. RandIndex measures on the MIT human labeled dataset for the three methods (and variants).

	Avg.	p**-value**	car	car2	car3	dog	phone	table	toy	hand	person
HGVS [9]	0.550	0.008	0.602	0.401	0.689	0.260	0.493	0.766	0.809	0.499	0.430
Layers++ [27]	0.775	0.050	0.612	0.512	0.778	0.964	0.567	0.909	0.832	0.814	0.986
Layers++ (8 layers)	0.690	0.021	0.711	0.510	0.802	0.531	0.564	0.842	0.874	0.590	0.790
Layers++ (10 layers)	0.675	0.027	0.661	0.517	0.799	0.613	0.551	0.853	0.846	0.640	0.597
nLayers	0.823	*	0.836	0.589	0.766	0.974	0.578	0.979	0.858	0.881	0.944
nLayers (8 layers)	0.808	0.275	0.611	0.590	0.821	0.975	0.575	0.981	0.852	0.920	0.951
nLayers (12 layers)	0.800	0.204	0.603	0.574	0.810	0.974	0.575	0.944	0.876	0.889	0.952
nLayers (2 frames)	0.793	0.124	0.608	0.535	0.755	0.970	0.578	0.979	0.841	0.923	0.951

[15] F. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. IT*, 47(2):498–519, Feb. 2001.

[16] M. Kumar, P. Torr, and A. Zisserman. Learning layered motion segmentations of video. *IJCV*, 76(3):301–319, Mar. 2008.

[17] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. *CVPR*, pp. 1–8, 2008.

[18] C. Liu, W. Freeman, E. Adelson, and Y. Weiss. Human-assisted motion annotation. *CVPR*, pp. 1–8, 2008.

[19] R. Morris, X. Descombes, and J. Zerubia. The Ising/Potts model is not well suited to segmentation tasks. *Proc. IEEE Digit. Sig. Process. Workshop*, pp. 263–266, 1996.

[20] D. Murray and B. Buxton. Scene segmentation from visual motion using global optimization. *PAMI*, 9(2):220–228, Mar. 1987.

[21] K. Mutch and W. Thompson. Analysis of accretion and deletion at boundaries in dynamic scenes. *PAMI*, 7(2):133–138, Mar. 1985.

[22] W. Rand. Objective criteria for the evaluation of clustering methods. *J. Amer. Statistical Assoc.*, 66(336):846–850, Dec. 1971.

[23] T. Schoenemann and D. Cremers. High resolution motion layer decomposition using dual-space graph cuts. *CVPR*, pp. 1–7, 2008.

[24] A. Shekhovtsov, I. Kovtun, and V. Hlavac. Efficient MRF deformation model for non-rigid image matching. *CVPR*, pp. 1–6, 2007.

[25] E. Sudderth and M. Jordan. Shared segmentation of natural scenes using dependent Pitman-Yor processes. *NIPS*, pp. 1585–1592, 2009.

[26] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. *CVPR*, pp. 2432–2439, 2010.

[27] D. Sun, E. Sudderth, and M. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. *NIPS*, pp. 2226–2234, 2010.

[28] P. Torr, and R. Szeliski, and P. Anandan An integrated Bayesian approach to layer extraction from image sequences. *PAMI*, 23(3):297–303, Mar. 2001.

[29] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. Modeling temporal coherence for optical flow. *ICCV*, pp. 1116–1123, 2011.

[30] C. Wang, M. de La Gorce, and N. Paragios. Segmentation, ordering and multi-object tracking using graphical models. *ICCV*, pp. 747–754, 2009.

[31] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. IP*, 3(5):625–638, Sep. 1994.

[32] Y. Weiss. Smoothness in layers: Motion segmentation using non-parametric mixture estimation. *CVPR*, pp. 520–526, 1997.

[33] Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. *CVPR*, pp. 321–326, 1996.

[34] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. *BMVC*, pp. 108.1–108.11, 2009.

[35] J. Wills, S. Agarwal, and S. Belongie. What went where. *CVPR*, v. 1, pp. 37–454, 2003.

[36] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *PAMI*, 27(10):1644–1659, Oct. 2005.

[37] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *CVPR*, pp. 1293–1300, 2010.