Local Layering for Joint Motion Estimation and Occlusion Detection

Deqing Sun[1] Ce Liu[2] Hanspeter Pfister[1]
[1]Harvard University [2]Microsoft Research

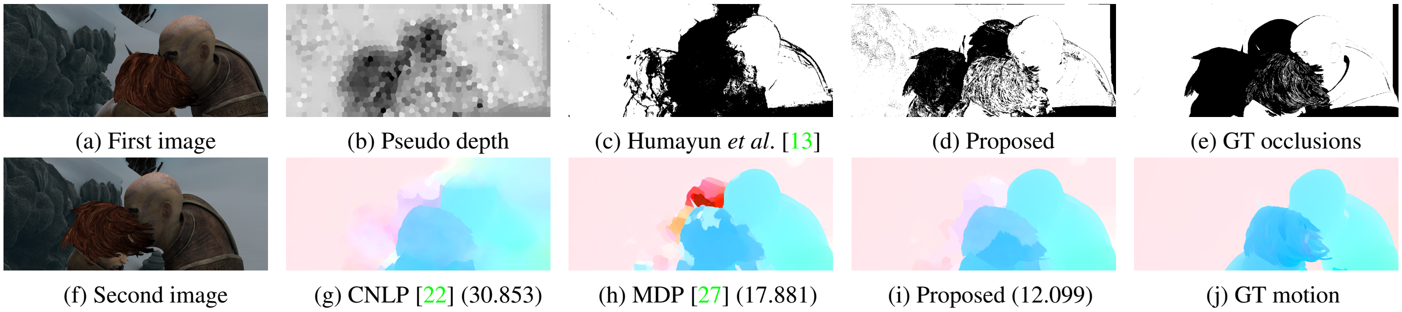

(a) First image (b) Pseudo depth (c) Humayun *et al.* [13] (d) Proposed (e) GT occlusions
(f) Second image (g) CNLP [22] (30.853) (h) MDP [27] (17.881) (i) Proposed (12.099) (j) GT motion

Figure 1. The proposed approach detects occlusions locally on a per-occurrence basis and retains uncertainty on motion and occlusion relationship during inference. It improves two baseline optical flow methods on motion estimation and has occlusion detection results comparable with a learning-based method. The pseudo depth is a visualization of the local occlusion relationship. Local layers with brighter values are likely to occlude those with darker ones. The numbers in parenthesis are the average end-point error (EPE).

Abstract

Most motion estimation algorithms (optical flow, layered models) cannot handle large amount of occlusion in textureless regions, as motion is often initialized with no occlusion assumption despite that occlusion may be included in the final objective. To handle such situations, we propose a local layering *model where motion and occlusion relationships are inferred jointly. In particular, the uncertainties of occlusion relationships are retained so that motion is inferred by considering all the possibilities of local occlusion relationships. In addition, the local layering model handles articulated objects with self-occlusion. We demonstrate that the local layering model can handle motion and occlusion well for both challenging synthetic and real sequences.*

1. Introduction

Despite recent advances, reliable motion estimation remains a challenging problem especially in the presence of large amounts of occlusion and textureless regions. It is well-known that correspondence and segmentation are chicken-and-egg problems. When the local grouping is known, correspondence becomes much easier. Vice versa, when correspondence is known, grouping can be reliably inferred by grouping pixels based on their motion. Existing methods, however, tend to treat each problem separately.

Optical flow [12], for example, assigns a flow vector for each pixel and completely ignores occlusion relationships between pixels. To handle occlusion, robust functions were introduced so that pixels to be occluded in the next frame may choose the right motion although the brightness constancy assumption is violated [6].

Furthermore, the aperture problem [2], namely local matching can be ambiguous, causes optical flow to fail miserably when the scene is completely textureless and occlusions prevail, such as the "two bars" sequence [26] in Figure 2. The only deterministic features such as X- and T-junctions, which are often caused by occlusions, may misguide optical flow algorithms to propagate erroneous flow vectors to the rest of the image.

Although recent advances in optical flow adopt sparse feature matching to handle large-displacement [8], large amount of occlusions can easily corrupt even the state-of-the-art optical flow algorithm [9], as shown in Figure 1.

To explicitly model the grouping of pixels and the fact that some pixels will be occluded, layered models were invented and widely studied to decompose the scene into several moving layers, each of which consists of appearances, mask and motion [3, 10, 14, 25]. Occlusion reasoning becomes trivial once the layers are given: a front layer occludes those behind it in every overlapping region in the image. Given the generative model, we need to infer the

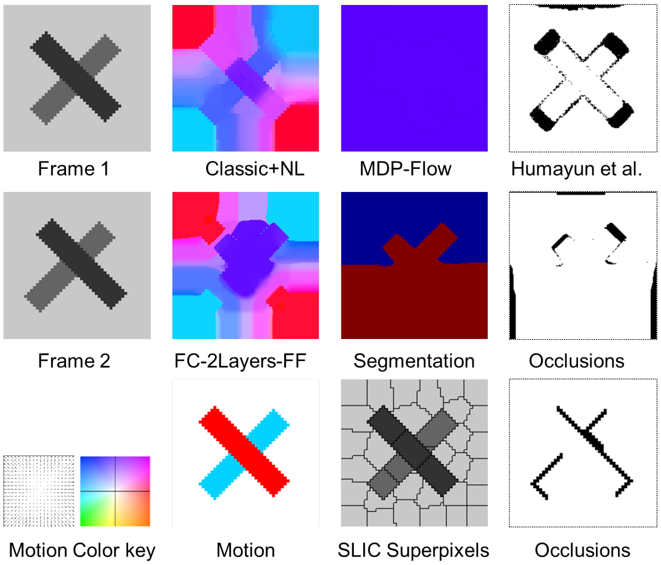


Figure 2. Results on the textureless "two bars" sequence [26] by different representations. Top row: classical optical flow [22] or feature embedded optical flow methods [27] fail to produce reasonable results. The occlusion detection method by Humayun *et al.* [13] uses output by several optical flow methods but fail to reliably detect the occlusions. Middle row: the global layered approach [23] gets stuck at the optical flow initialization by Classic+NL. Bottom row: by using superpixels [1] that well respect the static image boundaries, our local layering method correctly recovers the occlusion and motion. Motion is encoded using the color key from [5].

number of layers, the layer ownership, depth ordering, and the motion for each layer. Inferring the number of layers and the relative depth ordering among all the layers is very challenging as more layers are expected.

Although promising results have been obtained from advances in layered models [23], automatically extracting layers from a video sequence is still a challenging problem. Previous work on layers relies on either motion or color cues to initialize layer segmentation. For example, an optical flow algorithm is applied first to extract motion vectors, and then clustering is used to get layer primitives [23]. Nevertheless, the optical flow in the first step can be unreliable as we have analyzed. In addition, these global layers are limited in capturing mutual or self occlusions, and often only contain a few number of layers because the complexity explodes as the number of layers increases. Therefore, existing layered models are more suitable for characterizing scenes with several big plane motions but are fundamentally limited in capturing scenes containing articulated objects.

Some occlusion detection methods use optical flow as a feature, although the flow is estimated with the assumption of no occlusion. Humayun *et al.* [13] treat occlusion detection as a binary classification problem. They propose to use optical flow estimated by a number of different algorithms to generate features for the random forest classifiers. Training takes significant amount of time and errors in optical flow estimation may propagate to occlusion detection. Stein and Hebert [21] extract various mid-level features to predict occlusion boundaries. Sundberg *et al.* [24] compute features from optical flow and obtain significant improvement in predicting occlusion boundaries. However, all methods are separate from motion estimation and neither uses the detected occlusion (boundaries) to improve optical flow.

Ayvaci *et al.* [4] use a sparse prior to model occlusion and jointly solve optical flow and occlusion. Their occlusion reasoning is mainly based on the data matching error and a sparse prior, similar to the outlier process. The model has no notion of foreground and background to geometrically reason about occlusions using the motion. Black and Fleet [7] propose to locally track an individual patch over time to reason about occlusion and motion boundaries. They use the velocity on both sides of the occlusion boundaries to predict the (dis)occlusions of pixels. However, their method does not work for textureless scenes, such as the "two bars" example [26] in Figure 2. Yamaguchi *et al.* [28] jointly estimate the motion of superpixels and the occlusion boundaries between superpixels for epipolar-constrained optical flow estimation [11]. While the occlusion boundaries are used for locating motion boundaries, the method does not use the motion of superpixels to detect occlusion.

In this paper, we attempt to jointly infer motion and occlusion by means of a *local layering* model[1]. Our model works on superpixel representations obtained from over-segmentation. We not only model the motion for each superpixel like previous work, but also explicitly model the occlusion relationships between neighboring super-pixels. The motion of every superpixel, therefore, depends on its occlusion relationships with its neighbors. Compared with global layered models, our approach decides occlusion locally and does not require strictly ordering all the layers in depth. In the inference, we keep the uncertainties of both motion and occlusion relationships so that motion is inferred by considering all the possibilities of local occlusion relationships and vice versa, the occlusion relationship is inferred by considering all possible motion vectors. To reduce the huge solution space for the motion, our method uses the output of optical flow methods as candidates and improves the baseline methods, particularly in occlusion regions.

We have tested our model on both toy and real examples. Our local layered model solves the challenging two-bar s-

[1] McCann and Pollard [18] introduce local layering to model the complex occlusions in natural, challenging scenes. The users decide the depth ordering per overlap and a list graph is used to ensure the consistency of the local depth ordering; we borrow their idea of deciding occlusions locally on a per-occurrence basis. Our method, however, is fully automatic.

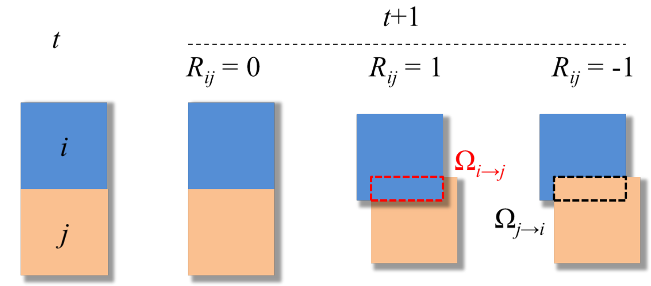


Figure 3. Three possible occlusion relationships between two local layers i and j. The set $\Omega_{i\to j}$ contains the pixels in i that bump into pixels in j at the next frame and similarly for $\Omega_{j\to i}$. Note that $\Omega_{i\to j}$ and $\Omega_{j\to i}$ depend on the unknown motion.

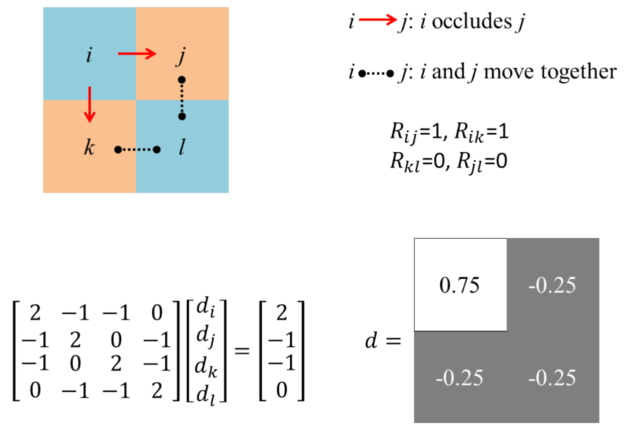
timulus. We have also evaluated our method on the MPI Sintel database [9] and demonstrated that our method improves the baseline algorithms that provide the motion candidate for our method, and also performs comparably with one learning-based occlusion detection algorithm [13].

2. Representation and Visualization

The local layering representation. Consider the scene in Figure 6(a). Even though we can have a semantic segmentation of the scene into apple and hand layers, global layers still have difficulty in modeling the mutual occlusion between the hand and the apple and self-occlusion of the hand. However, occlusion reasoning becomes feasible if we model the phenomenon locally on a per occurrence basis [18]. We propose to model the scene using a set of local layers to handel complex occlusions. Each local layer explains a small local region and has its own motion. Mutual occlusion is unlikely because of the small size of the local layers. The local layering representation only requires the segmentation boundaries to be consistent with the motion boundaries.

Given a pair of images $\mathbf{I}_t$ and $\mathbf{I}_{t+1}$, the unknowns include a segmentation of the first frame into local layers, the motion of each local layer, the occlusion relationship between spatially close local layers, and the occlusion map for the first frame. To simplify the problem, we pre-segment the first frame using the SLIC superpixel algorithm [1] with static image (and motion) cues. In the rest of the paper, we use local layers and superpixels exchangeably. Now we need to infer the motion $\mathbf{m}$ of every local layer, the occlusion relationship between layers $\mathbf{R}$, and a per-pixel occlusion map $\mathbf{o}$. There are three relationships between two local layers i and j: i occludes j ($R_{ij}=1$), i and j move together ($R_{ij}=0$), and j occludes i ($R_{ij}=$–1), as shown in Figure 3. A pixel p is occluded if $o_p=1$ and visible if $o_p=0$.

The pseudo depth visualization. To better visualize the local occlusion relationship, we derive a global pseudo depth map from the occlusion relationship, as shown in Figure 1(b). The brighter a local layer's pseudo depth is, the more likely will this local layer occlude other ones. Let d_i be the pseudo depth for the ith superpixel. We compute the pseudo depth to satisfy the following constraints


i
j
k
l
$i \rightarrow j$: i occludes j
$i \cdots j$: i and j move together
R_{ij}=1, R_{ik}=1
R_{kl}=0, R_{jl}=0
d =
0.75
-0.25
-0.25
-0.25

$$\begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} d_i \\ d_j \\ d_k \\ d_l \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

Figure 4. A toy example to explain the computation of pseudo depth from the local occlusion relationship. The brightest top left patch occludes its two neighboring patches, while the rest three patches move together.

$$d_i - d_j = R_{ij}. \tag{1}$$

If superpixel i occludes j, *i.e.* $R_{ij} = 1$, we enforce the pseudo depth of i to be larger than that of j. If i is occluded by j, we enforce the pseudo depth of i to be smaller than that of j. If i and j move together, we encourage their pseudo depths to be the same. We can obtain a system of linear equations for the pseudo depth

$$\mathbf{A}\mathbf{d} = \mathbf{b}, \tag{2}$$

where the set $\mathcal{N}_i^+$ contains all the spatially close superpixels that may bump into i in the next frame, $A(i,i) = |\mathcal{N}_i^+|$, $A(i,j) =$–$1, j \in \mathcal{N}_i^+$ and 0 otherwise, and $b(i) = \sum_{j\in\mathcal{N}_i^+} R_{ij}$. Solving the linear equation system gives the pseudo depth (for singular $\mathbf{A}$, we compute the pseudo inverse). Figure 4 shows the constraints and results for a toy example. The occluding local layers are assigned larger pseudo depth values than the occluded ones, while local layers moving together are assigned similar values.

3. Probabilistic Model

We adopt a probabilistic approach to model the dependence between the observed and the unknowns and their priors. Figure 5 shows our graphical model. We use an EM algorithm to maximize the posterior probability density function (p.d.f.) of the motion and the occlusion relationship, while marginalizing over the per-pixel occlusion map

$$\{\hat{\mathbf{m}}, \hat{\mathbf{R}}\} = \arg\max_{\mathbf{m},\mathbf{R}} \sum_{\mathbf{o}} p(\mathbf{m}, \mathbf{o}, \mathbf{R} | \mathbf{I}_t, \mathbf{I}_{t+1}), \tag{3}$$

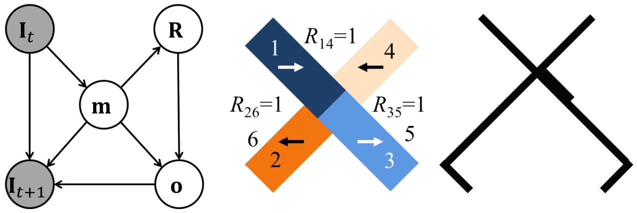


Figure 5. Left: the graphical model of our local layering approach. Middle: the explanation for the "two bars" sequence with the motion for every local layer and the occlusion relationship between local layers ($R_{ij}=1$ means that i occludes j). Right: inferred occlusion from the motion and occlusion relationship in the middle.

where the posterior factorizes as $p(\mathbf{m},\mathbf{o},\mathbf{R}|\mathbf{I}_t,\mathbf{I}_{t+1}) \propto$

$$p(\mathbf{I}_{t+1}|\mathbf{o},\mathbf{m},\mathbf{I}_t)p(\mathbf{o}|\mathbf{R},\mathbf{m})p(\mathbf{R}|\mathbf{m})p(\mathbf{m}|\mathbf{I}_t), \qquad (4)$$

where the first (data) term describes how to generate the next frame given the current frame, the motion, and the occlusions, the second and third terms encodes the constraints among motion, occlusion, and occlusion relationship, and the last one is the conditional motion prior. We will explain each term as follows.

Data term. The data term tells how we can generate the next frame from the current frame, the motion, and the occlusion map. Generally, if a pixel is visible, the appearance of this pixel at the current frame gives strong constraint on the appearance of the corresponding pixel at the next frame. However, an occluded pixel has no corresponding pixel at the next frame and should not be used for generating the next one. Hence $-\log p(\mathbf{I}_{t+1}|\mathbf{o},\mathbf{m},\mathbf{I}_t) \propto$

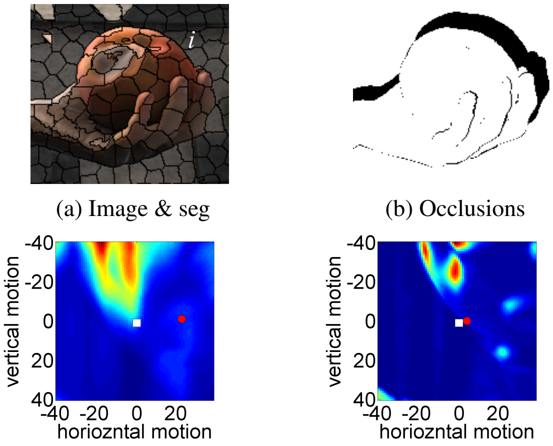
$$\sum_i \sum_{p\in\Omega_i} [\rho_D(I_t^p - I_{t+1}^{p+m_p})\bar{o}_p + \lambda_O o_p], \qquad (5)$$

where the set Ω_i contains all the pixels in the ith local layer, ρ_D is a robust penalty function, λ_O is a constant penalty for occlusion, and $\bar{o}_p = 1 - o_p$. This data term has been used in previous global layered models [23] on a per-pixel basis.

To gain some intuition, we plot the sum of squared difference (SSD) surface for a superpixel with occlusions in Figure 6. The minimum of the SSD surface is far from the true motion in the presence of large occlusions. If we disable the occluded pixels, the minimum of the modified SSD surface is close to the ground truth motion. For superpixels with occluded region, the data term in Eq. (5) mainly relies on their visible pixels.

Motion prior. Our prior for the motion of local layers is similar to that for optical flow. Both encode the fact that motion of real-world objects is smooth and slow, but occasionally abrupt [20]. We assume that abrupt motion tends to happen at object boundaries, across which appearances often change. Our conditional motion prior is $-\log p(\mathbf{m}|\mathbf{I}_t) \propto$

$$\sum_i \Big\{ \sum_{p\in\Omega_i} \lambda_S \rho_S(m_p) + \sum_{j\in\mathcal{N}_i} \lambda_F w_{ij} \rho_F(\bar{m}_i - \bar{m}_j) \Big\}, \qquad (6)$$



(a) Image & seg

(b) Occlusions

(c) SSD surface

(d) Modified SSD surface

Figure 6. A large portion of the marked superpixel i in (a) is occluded, as indicated by the occlusion map in (b). The minimum of the sum squared difference (SSD) surface (red circle) is far from the true motion (white rectangle) in (c), while the minimum of the occlusion-modified SSD surface in (d) is close to the true motion.

where the set $\mathcal{N}_i$ contains all the spatially neighboring local layers of i, $\bar{m}_i$ is the average motion of pixels in the local layer i, and $w_{ij} = \max\Big\{\exp\{-\frac{||\bar{I}_t^i - \bar{I}_t^j||^2}{\sigma_I^2}\}, T\Big\}$, in which $\bar{I}_t^i$ is the mean color of the local layer i, σ_I is the standard deviation of the Gaussian kernel, and T is a threshold. The weights allow motion boundaries to lie between superpixels with different appearances. Again λ_* is a constant and ρ_* is a robust penalty function.

Motion and occlusions. If motion is known, it provides constraints to the occlusion relationship, as shown in Figure 3. Specifically, if two local layers overlap each other at the next frame according to their motion, then their occlusion relationship should indicate the occurrence of occlusion; otherwise, the occlusion relationship should prefer the "moving together" explanation. The conditional distribution of the occlusion relationship given the motion encodes the constraints as

$$-\log p(\mathbf{R}|\mathbf{m}) \propto \sum_i \sum_{j\in\mathcal{N}_i^+} \lambda_P \Big\{ \delta(|\Omega_{i\to j}| \neq 0)\delta(R_{ij}=0) + \delta(|\Omega_{i\to j}|=0)\delta(R_{ij}\neq 0) \Big\}, \qquad (7)$$

where the first term enforces that, when two local layers bump into each other at the next frame (the $R_{ij} = 1$ and $R_{ij} = -1$ images in Figure 3), one layer should occlude another, while the second term enforces that, when two local layers have no overlap (the $R_{ij} = 0$ image in Figure 3), they should move together. The indicator function $\delta(x) = 1$ if x is true and 0 otherwise. λ_P is a constant penalty to penalize the "forbidden" states. The set $\Omega_{i\to j}$ contains the pixels in i that overlap j at the next frame, as shown in Figure 3.

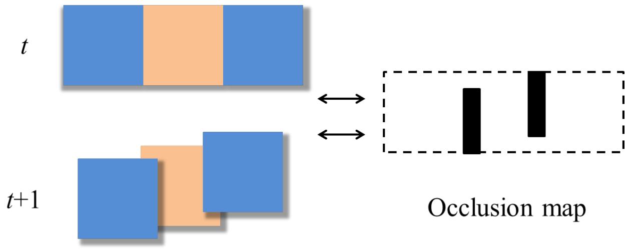


Figure 7. We need both the motion and occlusion relationships for the center orange layer and the two neighboring blue layers to jointly determine the occlusion for the center layer. To avoid the high-order interactions, we introduce the auxiliary occlusion map, which indicates occlusion by black. The motion and occlusion relationships should be consistent with the occlusion map.

The number of pixels in $\Omega_{i\to j}$, denoted by $|\Omega_{i\to j}|$, indicates whether i and j overlap at the next frame. The set $\mathcal{N}_i^+$ contains all local layers that may overlap with the ith layer at the next frame. Note that $\mathcal{N}_i^+$ is usually a superset of $\mathcal{N}_i$ that contains spatial neighbors of i, because occlusion may happen between non-spatially neighboring local layers. For example, the fast moving persons in Figure 1 occlude far-away background.

The motion $\mathbf{m}$ and the occlusion relationship $\mathbf{R}$ also provide strong constraints on the occlusion map $\mathbf{o}$, as shown in Figure 7 . Our model enforces that the predicted occlusion according to $\mathbf{m}$ and $\mathbf{R}$ should be consistent with the occlusion map. $-\log p(\mathbf{o}|\mathbf{R},\mathbf{m}) \propto$

$$\sum_i \sum_{j\in\mathcal{N}_i^+} \sum_{p\in\Omega_{i\to j}} \lambda_{\mathrm{C}} \Big\{ o_p \delta(R_{ij}\geq 0) + \bar{o}_p \delta(R_{ij}\leq 0) \Big\}. \quad (8)$$

Note that this term applies only to pixels that bump into pixels in other layers at the next frame. We encourage pixels in the occluding layer to be visible and pixels in the occluded layers to be occluded. Readers may wonder why we need the additional occlusion map $\mathbf{o}$, because we can determine $\mathbf{o}$ from $\mathbf{m}$ and $\mathbf{R}$. Actually inferring $\mathbf{o}$ from $\mathbf{m}$ and $\mathbf{R}$ may require high-order interactions among several local layers and makes the inference intractable, as shown in Figure 7. Introducing the occlusion map and the coupling term avoids the high-order terms.

4. Inference

Our method alternates between computing the probability of a pixel being occluded and inferring for the motion and occlusion relationship, as summarized in Table 1.

Per-pixel occlusion probability. Given an estimate of the motion and the occlusion relationship, we compute the probability of a pixel being occluded as

$$\Pr(o_p = 1) = \frac{\exp\{-\alpha E^p_{\text{occ}}\}}{\exp\{-\alpha E^p_{\text{occ}}\} + \exp\{-\alpha E^p_{\text{vis}}\}}, \quad (9)$$

where α is a scaling constant to convert the energy to probabilities. The energy for being occluded and visible are respectively

$$E^p_{\text{occ}} = \sum_{j\in\mathcal{N}_i^+ : p\in\Omega_{i\to j}} \lambda_{\mathrm{C}} \delta(R_{ij} = 1) + \lambda_{\mathrm{O}}, \quad (10)$$

$$E^p_{\text{vis}} = \sum_{j\in\mathcal{N}_i^+ : p\in\Omega_{i\to j}} \lambda_{\mathrm{C}} \delta(R_{ij} = -1) + \rho_{\mathrm{D}}(I_t^p - I_{t+1}^{p+m_p}). \quad (11)$$

Both the matching cost and the comparability with the motion and occlusion relationship contribute to the energies. A pixel is more likely to be occluded if its matching cost is large and it belongs to an occluded superpixel.

Joint motion and occlusion relationship reasoning. Given the probability of pixels being occluded, we jointly estimate the motion and the occlusion relationship by minimizing Eq. (12). The motion state space is huge even when we assume a single integer translational motion for each local layer. Hence we restrict the motion space to a few candidate motion fields for large images. Given fixed motion candidates, we run min sum algorithm on the loopy graph [15, 19] to minimize Eq. (12). We propose several schemes to generate candidate motion fields. One scheme is to pre-select several optical flow fields as candidates. Each local layer can take the motion at the corresponding position from these candidate flow fields. Another scheme is to use one optical flow field, cluster the flow vectors by kmeans to construct additional constant flow fields as candidates [16]. Finally, we also test sampling the motion space around the current solution during the inference. We then adaptively keep the top motion candidates and sample around them. When the iteration stops, we threshold the occlusion probability to decide the occlusion state of every pixel. Figure 8 shows the change of the unknowns during the iteration for the textureless "two bar" sequence in Figure 2. With more iterations, the solution by the proposed method converges to the correct occlusion and motion.

Table 1. The algorithm for the local layering algorithm. The *sampling* step is omitted if we use a fixed motion space.

```
Input: frames $\mathbf{I}_t$ and $\mathbf{I}_{t+1}$
• Initialize: compute SLIC superpixels [1] and optical flow [22, 27]; sample integer motion around optical flow
• Loop until convergence (outer iteration)
    - Compute the probability of being occluded (Eq.(9))
    - Solve for motion and occlusion relationship: loop
      until convergence (inner iteration)
        - Select and sample around top motion states
        - Perform loopy belief propagation (Eq. (12))
• Threshold the occlusion probability
• Refine motion with detected occlusion
Output: motion, occlusion relationship, and occlusions
```

$$E(\mathbf{m},\mathbf{R}) = \sum_i \left\{ \Big\{ \sum_{p\in\Omega_i} \{\rho_D(I_t^p - I_{t+1}^{p+m_p})\big(1-\Pr(o_p=1)\big) + \lambda_O \Pr(o_p=1) + \lambda_{\mathrm{S}}\rho_S(m_p)\} + \sum_{j\in\mathcal{N}_i} \lambda_F w_{ij}\rho_F(\bar{m}_i - \bar{m}_j)\Big\} \right. \tag{12}$$

$$\left. + \sum_{j\in\mathcal{N}_i^+} \Big\{\lambda_P\{\delta(|\Omega_{i\to j}|\neq 0)\delta(R_{ij}=0) + \delta(|\Omega_{i\to j}|=0)\delta(R_{ij}\neq 0)\} + \sum_{p\in\Omega_{i\to j}} \lambda_{\mathrm{C}}\{\Pr(o_p=1)\delta(R_{ij}\geq 0) + \big(1-\Pr(o_p=1)\big)\delta(R_{ij}\leq 0)\}\Big\} \right\}.$$

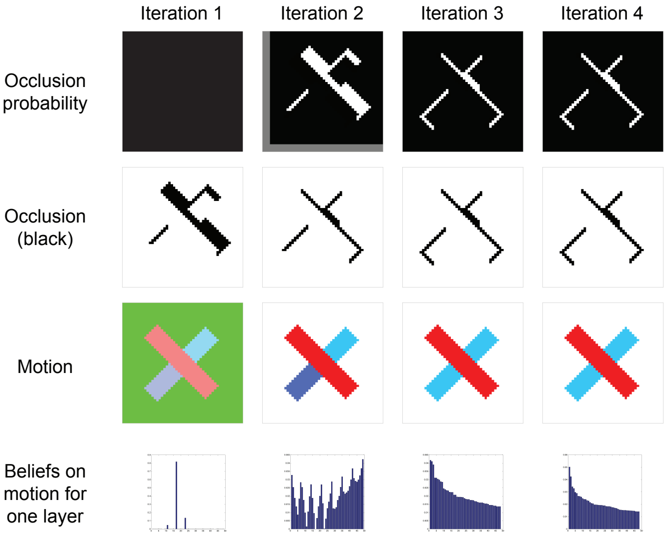


Figure 8. Convergence of the inference algorithm on the "two bars" sequence in Figure 2. By retaining uncertainty on both motion and occlusion relationship, the proposed method converges to the correct motion and occlusion.

The motion prior for local layers may not accurately describe the interaction within each local layer. To further improve the motion, we use the estimated motion and occlusion to initialize a modified optical flow method. Specifically, we disable the data term of the "Classic+NLP" method [22] in the detected occlusion regions to refine the estimated motion by our local layering method.

5. Results

We evaluate the proposed method on both synthetic data and the MPI Sintel dataset. For motion estimation, we compare the proposed method to two widely used optical flow methods: Classic+NL-FastP [22] and MDP-flow2 [27]. For occlusion detection, we compare the proposed method to one state-of-the-art learning-based occlusion detector [13]. We use the code from the authors' website. For [13], we add additional data from Sintel to train the "lean" version of its classifier. The code [13] outputs the probability of every pixel being occluded. We select the best threshold according to the GT occlusion to obtain the hard occlusion map.

Synthetic data. We first test on the classical "two bars" sequence [26], as shown in Figure 2. The sequence is challenging because it is textureless, has false T junctions, and contains occlusions. Weiss and Adelson [26] propose a layered method for the "two bars" sequences. Their method requires a perfect segmentation of the scene as input. Liu *et al*. [17] analyze the contour motion particularly for textureless sequences. The output is sparse motion for the contours and cannot be directly interpolated to be dense motion field. The MDP-Flow2 [27] method fails because the false T junctions breaks down the feature matching component of the method. The "Classic+NLP" [22] method also fails because of the textureless surfaces and the occlusions. We use a full solution space for the motion that covers the ground truth for this toy example. The proposed method correctly recovers the motion and the occlusions by locally reasoning about occlusions among over-segmented superpixels. Note that the slow motion prior is important to recover the background motion. Adding texture noise to the synthetic sequences helps the optical flow methods obtain better results, while the proposed method performs as well.

MPI Sintel. We test the proposed method using the MPI Sintel dataset [9]. These sequences contain complex occlusions that challenge the state of the art. Figure 9 shows the results on several challenging sequences by the proposed method and the baseline optical flow methods. By analyzing the occlusions locally on a per-occurrence basis, the proposed method detects most of the occlusions. The detected occlusions help recover the motion in the large occlusion regions, such as the background occluded by the hand, the leg, and the small dragon.

As summarized in Table 2, the proposed method improves the optical flow methods [22, 27] that serve as candidates, particularly in the unmatched (occlusion) regions. We further test some variants of the proposed method using a representative subset of the training clean set (the 1$^{\text{st}}$, 5$^{\text{th}}$, and 23$^{\text{rd}}$ image pairs from each of the 23 sequence). We find that adding motion cue to the SLIC algorithm results in slightly more accurate results (MotionSLIC in Table 3), suggesting benefits to jointly solve for segmentation, motion, and occlusions. We test two simple methods to construct diverse motion candidates but find no overall improvement (MDP-Kmeans3 and MDP-Sampling in Table 3).

We also evaluate the occlusion detection results in Ta-

Table 2. **Average end-point error (EPE)** results on the MPI Sintel *test* set. Unmatched correspond to the occlusion regions.

	Clean		Final	
	all	*unmatched*	*all*	*unmatched*
CNL-fastP [22]	6.940	37.866	8.439	41.014
MDP-Flow2 [27]	5.837	38.158	8.445	43.430
Proposed	**5.820**	**35.784**	**8.043**	**40.879**

Table 3. **Average end-point error (EPE)** results by two baseline optical flow methods, the proposed method and its variants on 69 *clean* image pairs from the MPI Sintel *training* set.

	all	*unmatched*
CNL-fastP [22]	5.149	11.155
MDP-Flow2 [27]	4.002	12.182
Proposed (CNL+MDP)	3.763	10.287
Proposed (MotionSLIC)	3.719	10.252
Proposed (MDP-Kmeans3)	4.492	11.990
Proposed (MDP-Sampling)	4.362	12.002

Table 4. **Average F-measure** (larger better) for occlusion detection, averaged over 69 MPI Sintel sequences. The *oracle* threshold is determined using the GT occlusion.

Threshold	*oracle*	*fixed* (0.5)
Humayun *et al.* [13]	0.535	0.448
Proposed	0.474	0.376

ble 4. The proposed method performs closely to the-state-of-the-art learning based approach [13], which has been trained to maximize the classification accuracy and uses many features, including output from a few optical flow methods. Visually the detected occlusion boundaries of the proposed method appear consistent with the ground truth, as shown in Figures 9 and 10.

6. Conclusions

We have introduced the local layering representation for motion estimation and occlusion detection. This flexible representation enables us to capture complex occlusions without resorting to a fully 3D model. We find that locally deciding the depth ordering on a per-occlusion basis is feasible when we jointly infer motion and occlusion relationship and retain the uncertainty on both during inference. Our simple representation achieves promising results on both the “two bars” sequence and the MPI Sintel dataset. The detected occlusions are close to the ground truth, even for complex occlusions. Our method improves over the baseline optical flow methods particularly in occlusion regions. Our work opens new avenues to jointly modeling motion and occlusions and suggests that exploring richer and more flexible representations can be fruitful for this challenging problem.

Acknowledgement This work has been partially supported by NSF grant OIA 1125087. DS would like to thank Jason Pacheco for helpful discussions on inference algorithms.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 34(11):2274–2282, 2012. 2, 3, 5

[2] E. Adelson and J. Movshon. Phenomenal coherence of moving visual patterns. *Nature*, 300:523–525, 1982. 1

[3] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *ICCV*, pages 777–784, Jun 1995. 1

[4] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *IJCV*, 97(3), May 2012. 2

[5] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92(1):1–31, March 2011. 2

[6] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *CVIU*, 63:75–104, 1996. 1

[7] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion boundaries. *IJCV*, 38(3):231–245, July 2000. 2

[8] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE TPAMI*, 33(3):500–513, Mar. 2011. 1

[9] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, IV, pages 611–625, 2012. 1, 3, 6

[10] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE TPAMI*, 17(5):474–487, 1995. 1

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2

[12] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 16:185–203, Aug. 1981. 1

[13] A. Humayun, O. Mac Aodha, and G. J. Brostow. Learning to Find Occlusion Regions. In *CVPR*, 2011. 1, 2, 3, 6, 7, 8

[14] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *CVPR*, pages 760–761, 1993. 1

[15] F. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE TIT*, 47(2):498–519, Feb. 2001. 5

[16] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, pages 1–8, 2008. 5

[17] C. Liu, W. T. Freeman, and E. H. Adelson. Analysis of contour motions. In *NIPS*, pages 913–920, 2006. 6

[18] J. McCann and N. S. Pollard. Local layering. *Siggraph*, 28(3), Aug. 2009. 2, 3

[19] J. M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, Aug. 2010. 5

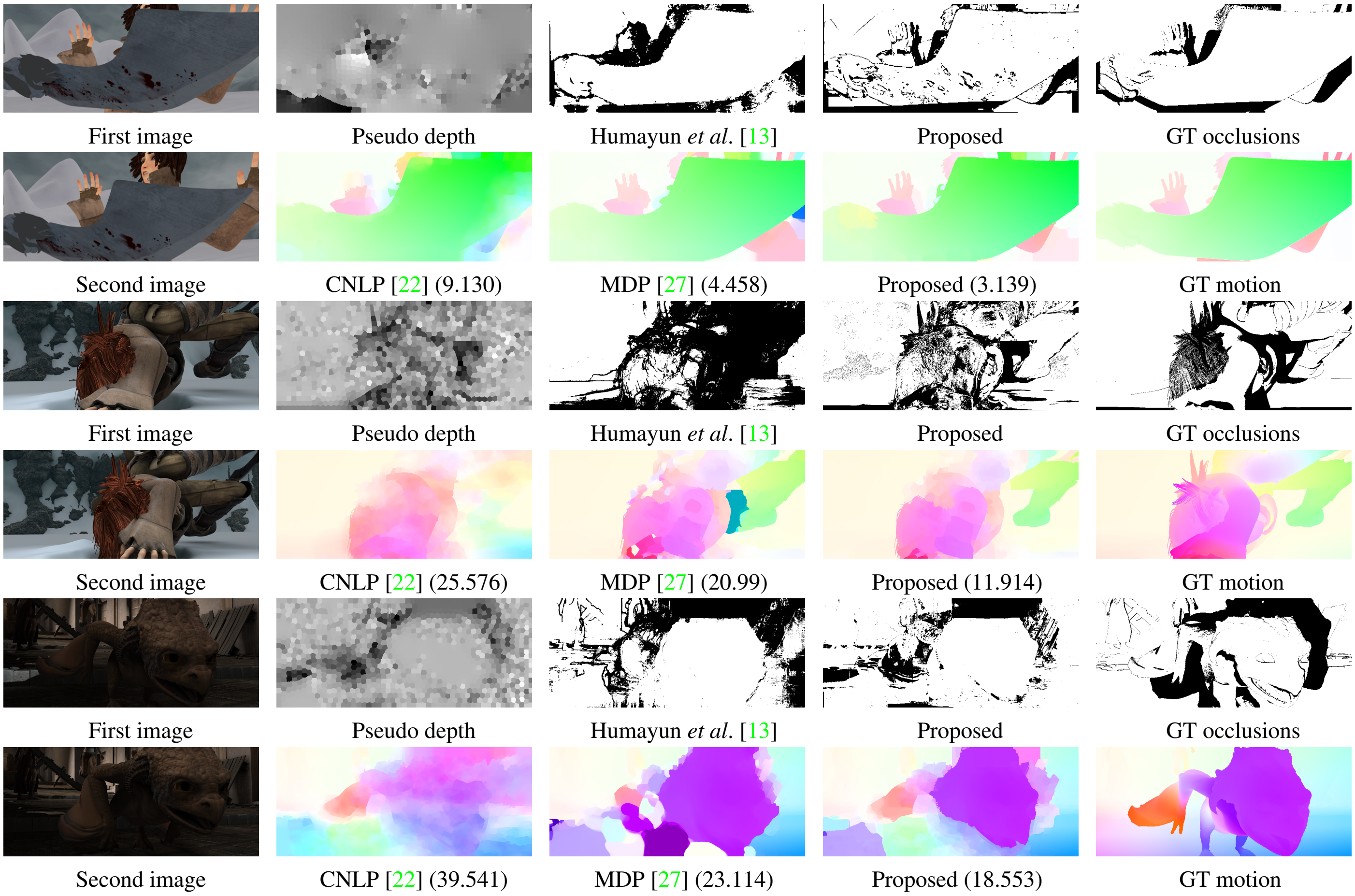
[20] S. Roth and M. J. Black. On the spatial statistics of optical flow. *IJCV*, 74(1):33–50, Aug 2007. 4

[21] A. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *IJCV*, 82(3):325–357, May 2009. 2

[22] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *IJCV*, 106(2):15–137, 2014. 1, 2, 5, 6, 7, 8

[23] D. Sun, J. Wulff, E. B. Sudderth, H. Pfister, and M. J. Black. A fully-connected layered model of foreground and background flow. In *CVPR*, pages 2451–2458, 2013. 2, 4

[24] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, pages 2233–2240, 2011. 2


First image | Pseudo depth | Humayun *et al*. [13] | Proposed | GT occlusions

Second image | CNLP [22] (9.130) | MDP [27] (4.458) | Proposed (3.139) | GT motion

First image | Pseudo depth | Humayun *et al*. [13] | Proposed | GT occlusions

Second image | CNLP [22] (25.576) | MDP [27] (20.99) | Proposed (11.914) | GT motion

First image | Pseudo depth | Humayun *et al*. [13] | Proposed | GT occlusions

Second image | CNLP [22] (39.541) | MDP [27] (23.114) | Proposed (18.553) | GT motion

Figure 9. The proposed method produces reasonable occlusion detection results, which helps recover the motion of the region occluded by the hand in the top row, the leg in the middle row, and the dragon in the bottom row. The numbers in parenthesis are the average EPE.

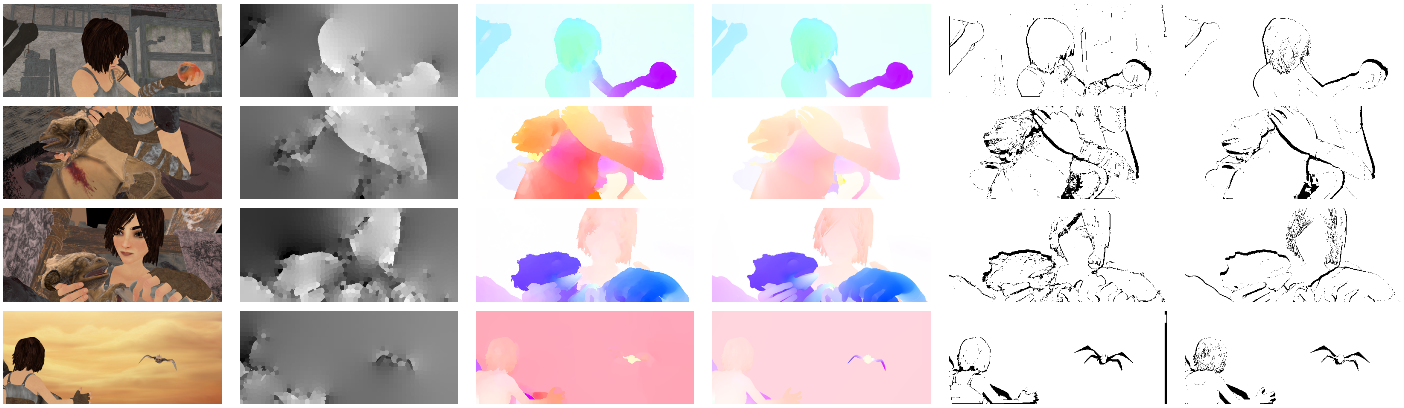

(a) First image (b) Pseudo depth (c) Motion (d) GT Motion (e) Occlusions (f) GT occlusions

Figure 10. More motion estimation and occlusion detection results. The detected occlusion boundaries by the proposed method are close to the ground truth. The brighter its pseudo depth is, the local layer is more likely to occlude other local layers.

[25] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE TIP*, 3(5):625–638, Sept. 1994. 1

[26] Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *CVPR*, pages 321–326, 1996. 1, 2, 6

[27] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE TPAMI*, 34(9):1744–1757, 2012. 1, 2, 5, 6, 7, 8

[28] K. Yamaguchi, D. A. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. In *CVPR*, pages 1862–1869, 2013. 2