Local Layering for Joint Motion Estimation and Occlusion Detection

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters

<table>
<thead>
<tr>
<th>Citation</th>
<th>Sun, D., C. Liu, and H. Pfister. 2014. Local Layering for Joint Motion Estimation and Occlusion Detection. In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), Columbus, Ohio, June 24-27, 2014: 1098-1105.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citable link</td>
<td><a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:21150406">http://nrs.harvard.edu/urn-3:HUL.InstRepos:21150406</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP">http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP</a></td>
</tr>
</tbody>
</table>
Local Layering for Joint Motion Estimation and Occlusion Detection

Deqing Sun\textsuperscript{1} Ce Liu\textsuperscript{2} Hanspeter Pfister\textsuperscript{1}
\textsuperscript{1}Harvard University \textsuperscript{2}Microsoft Research

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
The occlusion detection method by Humayun et al. [27] fail to produce reasonable results. The 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].

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. Our model works on superpixel representations obtained from oversegmentation. We not only model the motion for each superpixel like previous work, but also explicitly model the occlusion relationships between neighboring superpixels. 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-
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 handle 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 \( I_t \) and \( I_{t+1} \), 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 \( \hat{m} \) of every local layer, the occlusion relationship between layers \( \hat{R} \), and a per-pixel occlusion map \( o \). 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 4. 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 \( i \)th superpixel. We compute the pseudo depth to satisfy the following constraints

\[
\begin{align*}
    &d_i - d_j = R_{ij}, \\
    &d_i - d_j = R_{ij}, \\
    &d_i - d_j = R_{ij}, \\
\end{align*}
\]

where the set \( N_i^+ \) contains all the spatially close superpixels that may bump into \( i \) in the next frame, \( A(i, i) = |N_i^+|, A(i, j) = -1, j \in N_i^+ \) and 0 otherwise, and \( b(i) = \sum_{j \in N_i^+} R_{ij} \). Solving the linear equation system gives the pseudo depth (for singular \( A \), 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

\[
\{\hat{m}, \hat{R}\} = \arg\max_{m,R} \sum_o p(m, o, R|I_t, I_{t+1}),
\]
The data term tells how we can 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 $-\log p(I_{t+1}|o, m, I_t)$ is a robust penalty to penalize images in which $\rho_t$ contains all the pixels in the $t$th local layer, $\rho_D$ is a robust penalty function, $\lambda_O$ is a constant penalty for occlusion, and $\hat{\phi}_p = 1 - \phi_p$. 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 (white rectangle) in (c), while the minimum of the occlusion-modified SSD surface in (d) is close to the true motion.

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

$$-\log p(m|I_t) \propto \sum \{ \sum_{p \in \Omega_i} \lambda_S \rho_S(m_p) + \sum_{j \in N_i} \lambda_F \rho_F(\hat{m}_i - \hat{m}_j) \},$$

where $\lambda_F$ 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 $\lambda_o$ is a constant and $\rho_o$ is a robust penalty function.

**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 $\rho(m, o, R|I_t, I_{t+1}) \propto p(I_{t+1}|o, m, I_t)p(o|R, m)p(R|m)p(I_t)$,

$$\sum \sum_{p \in \Omega_i} \rho_D(I_i^m - I_{t+1}^p \rho_p) + \lambda_o \rho_o,$$

where the set $\Omega_i$ contains all the spatially neighboring local layers of $i$, $\hat{m}_i$ is the average motion of pixels in the local layer $i$, and $w_{ij} = \max \{ \exp(-||\hat{m}_i - \hat{m}_j||_2^2/T), 0 \}$, in which $\hat{m}_i$ is the mean color of the local layer $i$, $\sigma_j$ 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 $\lambda_o$ is a constant and $\rho_o$ is a robust penalty function.

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 $-\log p(m|I_t) = \sum \{ \sum_{p \in \Omega_i} \lambda_S \rho_S(m_p) + \sum_{j \in N_i} \lambda_F \rho_F(\hat{m}_i - \hat{m}_j) \}$.

**Figure 6.** A large portion of the marked superpixel 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.
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 $\Omega_{i\rightarrow j}$, denoted by $|\Omega_{i\rightarrow j}|$, indicates whether $i$ and $j$ overlap at the next frame. The set $\mathcal{N}_i^+$ contains all local layers that may overlap with the $i$th 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. 

$$-\log p(\mathbf{O} | \mathbf{R}, \mathbf{m}) \propto \sum_{i} \sum_{j \in \mathcal{N}_i^+} \sum_{p \in \Omega_{i\rightarrow j}} \lambda_c \left\{ o_p \delta(R_{ij} \geq 0) + \bar{o}_p \delta(R_{ij} \leq 0) \right\}. \quad (8)$$

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

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

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

$$E_{\text{occ}}^p = \sum_{j \in \mathcal{N}_i^+} \sum_{p \in \Omega_{i\rightarrow j}} \lambda_c \delta(R_{ij} = 1) + \lambda_o,$$

$$E_{\text{vis}}^p = \sum_{j \in \mathcal{N}_i^+} \sum_{p \in \Omega_{i\rightarrow j}} \lambda_c \delta(R_{ij} = -1) + \rho_D(I^p - I^p_{i+1}). \quad (11)$$

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.

<table>
<thead>
<tr>
<th>Input: frames $I_t$ and $I_{t+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Initialize: compute SLIC superpixels [1] and optical flow [22, 27]; sample integer motion around optical flow</td>
</tr>
<tr>
<td>- Loop until convergence (outer iteration)</td>
</tr>
<tr>
<td>- Compute the probability of being occluded (Eq.(9))</td>
</tr>
<tr>
<td>- Solve for motion and occlusion relationship: loop until convergence (inner iteration)</td>
</tr>
<tr>
<td>- Select and sample around top motion states</td>
</tr>
<tr>
<td>- Perform loopy belief propagation (Eq. (12))</td>
</tr>
<tr>
<td>- Threshold the occlusion probability</td>
</tr>
<tr>
<td>- Refine motion with detected occlusion</td>
</tr>
</tbody>
</table>

| Output: motion, occlusion relationship, and occlusions |
\[ E(m, R) = \sum_i \left\{ \sum_{p \in \Omega_i} \left\{ \rho_D(I_t^p - I_{t+1}^{p+m_p})(1 - \Pr(o_p = 1)) + \lambda_0 \Pr(o_p = 1) + \lambda_s \rho_S(m_p) \right\} + \sum_{j \in N_i} \lambda_F \rho_F(\bar{m}_i - \bar{m}_j) \right\} \]

\[ + \sum_{j \in N_i^+} \{ \lambda_p \left\{ \delta(\Omega_{i\to j} \neq 0) \delta(R_{ij} = 0) + \delta(\Omega_{i\to j} = 0) \delta(R_{ij} \neq 0) \right\} + \sum_{p \in \Omega_{i\to j}} \lambda_c \{ \Pr(o_p = 1) \delta(R_{ij} \geq 0) + (1 - \Pr(o_p = 1)) \delta(R_{ij} \leq 0) \right\} \} \]
Table 2. **Average end-point error (EPE)** results on the MPI Sintel *test* set. Unmatched correspond to the occlusion regions.

<table>
<thead>
<tr>
<th>Method</th>
<th>Clean all</th>
<th>Clean unmatched</th>
<th>Final all</th>
<th>Final unmatched</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNL-fastP [22]</td>
<td>6.940</td>
<td>37.866</td>
<td>8.439</td>
<td>41.014</td>
</tr>
<tr>
<td>MDP-Flow2 [27]</td>
<td>5.837</td>
<td>38.158</td>
<td>8.445</td>
<td>43.430</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>5.820</strong></td>
<td><strong>35.784</strong></td>
<td><strong>8.043</strong></td>
<td><strong>40.879</strong></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>all</th>
<th>unmatched</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNL-fastP [22]</td>
<td>5.149</td>
<td>11.155</td>
</tr>
<tr>
<td>MDP-Flow2 [27]</td>
<td>4.002</td>
<td>12.182</td>
</tr>
<tr>
<td>Proposed (CNL+MDP)</td>
<td>3.763</td>
<td>10.287</td>
</tr>
<tr>
<td>Proposed (MotionSLIC)</td>
<td>3.719</td>
<td>10.252</td>
</tr>
<tr>
<td>Proposed (MDP-Kmeans3)</td>
<td>4.492</td>
<td>11.990</td>
</tr>
<tr>
<td>Proposed (MDP-Sampling)</td>
<td>4.362</td>
<td>12.002</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Threshold</th>
<th>oracle fixed (0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Humayun et al. [13]</td>
<td>0.535</td>
<td>0.448</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.474</td>
<td>0.376</td>
</tr>
</tbody>
</table>

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

[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
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.

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.