Revisiting Depth Layers from Occlusions

Adarsh Kowdle
Cornell University
apk64@cornell.edu

Andrew Gallagher
Cornell University
acg226@cornell.edu

Tsuhan Chen
Cornell University
tsuhan@ece.cornell.edu

Abstract

In this work, we consider images of a scene with a moving object captured by a static camera. As the object (human or otherwise) moves about the scene, it reveals pairwise depth-ordering or occlusion cues. The goal of this work is to use these sparse occlusion cues along with monocular depth occlusion cues to densely segment the scene into depth layers. We cast the problem of depth-layer segmentation as a discrete labeling problem on a spatio-temporal Markov Random Field (MRF) that uses the motion occlusion cues along with monocular cues and a smooth motion prior for the moving object. We quantitatively show that depth ordering produced by the proposed combination of the depth cues from object motion and monocular occlusion cues are superior to using either feature independently, and using a naïve combination of the features.

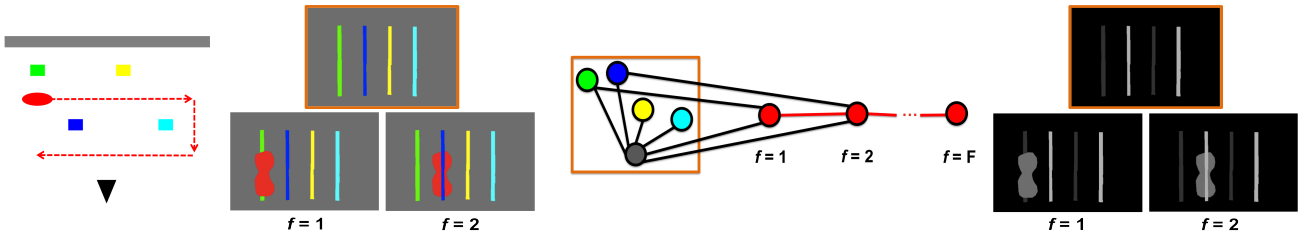
1. Introduction

We consider a time-series of images of a scene with moving objects captured from a static camera, and our goal is to exploit occlusion cues revealed as the objects move through the scene to segment the scene into depth layers. Recovering the depth layers of a scene from a 2D image sequence has a number of applications. Video surveillance often has a fixed camera focused on a scene with one or more moving objects. As objects move through the scene over time, we recover a layered representation of the scene. This aides tasks such as object detection and recognition in the presence of occlusions since one can reason about partial observations of an occluded object with a better 3D understanding of the scene [6, 15, 22]. In addition, a layered representation of the scene is useful in video editing applications, such as composing novel objects into the scene with occlusion reasoning [30] and changing the depth of focus [24].

An image sequence captured from a dynamic (moving) camera allows one to leverage powerful stereo matching cues to recover the depth and occlusion information of the scene. However, these cues are absent in the case of a static camera. For single images, monocular cues help reveal useful depth information [8, 10, 12, 13, 23, 28, 31, 32]. In this work, we consider a set of images with moving objects captured from a static camera. As the object moves it is either occluded by or occludes a portion of the scene, consequently revealing sparse pairwise ordering relationships [3, 29] between the moving object and the scene, and reveals long-range pairwise cues between the two regions of the scene it simultaneously interacts with. These pairwise cues are powerful, but sparse, which makes our goal of extracting dense pixel-level depth layers a hard problem.

In this work, we cast the problem of depth-layer segmentation as a discrete labeling problem on a spatio-temporal MRF over the video. We accumulate the pairwise ordering cues revealed as the object moves through the scene and include monocular cues to propagate the sparse occlusion cues through the scene. We over-segment the background scene (which has no moving objects) and construct a region-level MRF with edges between adjacent regions. In each frame, we identify the pixels corresponding to the moving object and add a node corresponding to each moving object for every frame of the video. We add temporal edges between the corresponding moving object nodes across frames, allowing us to encode a smooth motion prior for the moving object. As the object moves about the scene, we detect *motion occlusion events* and add edges between the background scene node and the corresponding moving object node, including long range edges between two background scene nodes to encode the pairwise depth-ordering or occlusion cues. An overview of our proposed formulation for a single moving object is shown in Figure 1, with the extension to handle multiple objects in Section 3.4.

Contributions. Our paper, for the first time, proposes a framework for recovering depth layers in static camera scenes by combining depth-ordering cues from moving objects and cues from monocular occlusion reasoning. Our approach works with any moving object (human or otherwise) and extends to multiple objects moving in the scene. We show that this depth layer reasoning out-performs the current state-of-the-art in terms of depth-layer recovery.



(a) (b) (c) (d)

Figure 1: Overview. (a) Ground-truth top view, black triangle shows the camera looking up at a scene with the red moving object region following the path shown in the red arrow; (b) Shows the background scene in the orange box and two frames from the input sequence where the red object interacts with the background regions to reveal pairwise depth-ordering cues such as red occludes green, blue occludes red; (c) A graph constructed over the background regions is shown in the orange box. Each colored node corresponds to the respective colored region in (b). The red nodes correspond to the moving object with a node for every frame f in the input sequence ($\{1, 2, \ldots, F\}$). The black edges enforce the observed pairwise depth-ordering, for instance between the green-red nodes at $f = 1$, and blue-red nodes at $f = 2$. The red edges enforce a smooth motion model for the moving object; (d) Shows the inferred depth layers, white = near and black = far.

2. Related work

Research in cognitive science has shown that humans rely on occlusion cues to obtain object boundaries and depth discontinuities even in the absence of strong image cues such as edges and lighting [17, 25]. Recovering occlusion boundaries in a scene is a classic problem that has been a topic of wide interest. We focus on prior work with the similar setting of *static camera* scenarios. We broadly classify these works into learning-based approaches and approaches that purely rely on motion occlusion cues revealed by the moving object.

Learning-based approaches. Prior work has explored learning-based approaches for estimating the depth of the scene [8,10,12,14,23,28,31,32] and estimating depth ordering [13, 16] from a single image for 3D scene understanding. Recent work has shown objects (clutter) in the scene to aid better depth estimation of the scene [9, 11] through affordances.

Moving beyond single image scenarios to image sequences, Fouhey *et al*. [5] showed that the pose of people interacting with a cluttered room can be used to obtain functional regions and recover a coarse 3D geometry of the room. Our work is complementary to this work, and in particular is agnostic to priors about the type of moving object and the type of scene (indoor or outdoor). In other words, we do not require a human as the moving object. We relate back to prior research in cognitive science that show that occlusion cues we observe are agnostic to any prior about the object. We use these sparse, yet strong occlusion cues revealed by the moving object to aid the dense depth layer segmentation of the scene.

Depth layers from motion occlusion. We work with a single static camera image sequence that precludes us from using algorithms for multiview occlusion reasoning using a moving object [7]. We focus on segmenting a scene captured by a single static camera into depth layers using occlusion cues revealed by the moving objects. Our work is inspired by the work of Brostow *et al*. [3] and Schodl *et al*. [29] who use pairwise occlusion cues to "push" and "pop" the regions of the scene affected by the moving object to obtain depth layers at each frame. A limitation of these works is that they reason only about the portion of the scene the object interacts with, leaving behind huge portions of the scene at an unknown depth. In addition, since the interaction with each region is treated independently it leads to excessive fragmentation of the scene as we show in Section 4. This fragmentation can be partially avoided [29] by making the (possibly over-restrictive) strong assumption that the moving object stays at a constant depth. Our model includes a more reasonable model of object motion.

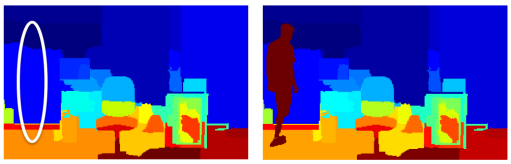
In summary, we revisit depth layers from occlusions and address limitations of prior work via a unified framework that leverages sparse depth-ordering cues revealed by the moving object and gracefully propagates them throughout the whole scene.

3. Algorithm

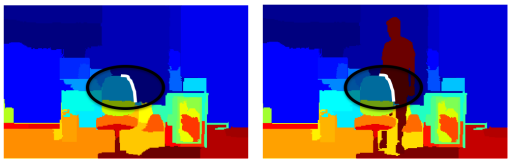
We formulate the task of segmenting the scene into depth layers as a discrete labeling problem. In this section, we first describe our formulation as applied to a scene with a single moving object and then extend the same framework to handle multiple moving objects in the scene.

3.1. Spatio-temporal graph

Background scene segmentation. We refer to the scene without any moving objects as the background scene. We use a calibration stage to obtain a clean background image without any moving objects. In the absence of the calibration stage we take advantage of the static camera scenario and obtain an estimate of the background image as the median image over the video. Given the background image we obtain an over-segmentation using mean shift segmentation [4] to give us about 300 superpixels. We treat this segmentation as a stencil of background superpixels that applies to each frame of the video.



(a) Object in-front-of background scene region



(b) Object behind background scene region

Figure 2: Pairwise depth-ordering cues. Left image shows the background scene segmentation and the right image shows an intermediate frame segmentation with the moving object segment. (a) A region in the background is covered by the moving object (white ellipse) indicating that the moving object occludes the background region; (b) Observing that the boundary corresponding to the background region (white pixels in black ellipse) does not change when the moving object comes in contact with it reveals that the moving object is occluded by the background region. It also reveals new relationships via transitivity; the chair occludes the object and at the same instant the object occludes regions on the wall; therefore the chair occludes the regions on the wall.

Moving object segmentation. Given the superpixel stencil for the background scene, we update this superpixel map for every frame by identifying the pixels corresponding to the moving object via background subtraction. We model the appearance of the background using a per-pixel Gaussian distribution (A_p) centered at the mean color (RGB space) of the pixel across the whole video. Given A_p, for every frame we estimate the likelihood for each pixel belonging to the background. We label pixels with background likelihood above 90% as confident background pixels and below 10% likelihood as confident moving object pixels. Using these as confident initial seeds, we learn an appearance model for the background (BG) and the moving object (FG). The moving object segmentation is obtained using iterative graph-cuts [1, 2, 20] updating the BG/FG color models with each iteration similar to GrabCut [27]. Figure 2 shows examples of the moving object segmentation overlaid on the background segmentation.

After this stage, we have the background scene superpixel map and the moving object segmentation for each frame. A region-level MRF is constructed over the background scene superpixels where each superpixel is a node with an edge to adjacent superpixels. We add a node corresponding to the moving object for every frame of the video and add temporal edges connecting the moving object nodes on adjacent frames. This graph is illustrated in Figure 1(c).

3.2. Pairwise depth-ordering cues

The object moving through the scene is either occluded by or occludes portions of the scene. We refer to these as *motion occlusion events*. In our superpixel representation of the scene, we accumulate the pairwise cues using a matrix we call Occlusion Matrix (O) where, $O_{i,j} \in \{-1, 0, +1\}$ indicates the relationship between superpixel i and superpixel j *i.e.*, $\{i$ occluded by j, no cue, and i occludes $j\}$, respectively. O is a skew-symmetric matrix *i.e.*, $O_{i,j} = -O_{j,i}$. The matrix is updated at every frame of the video using detected motion occlusion events or using learnt monocular cues in absence of occlusion cues.

Motion occlusion cues. Low-level cues revealed by the moving object in the scene serve as sparse, yet strong pairwise depth-ordering cues. We work with the abstract superpixel representation of each frame and use cues similar to prior work [3] to obtain pairwise relationship between the moving object segment and the superpixel it interacts with. The cues are intuitive, given a background region the moving object is interacting with, we use the moving object pixels and the boundary pixels of the background region to infer whether the object moved in-front-of this region or behind this region, respectively, as illustrated in Figure 2.

We update the corresponding entry of the occlusion matrix with $O_{i,j}$ as $+1$ to indicate that superpixel i occludes superpixel j and set $O_{j,i}$ to -1. In addition to the pairwise depth-ordering cues between the moving object and the superpixel it is interacting with, we also enforce transitivity while updating the matrix. If the object is occluded by a region of the background scene and is simultaneously occluding several regions of the background scene, via transitivity it establishes a pairwise relationship between the occluding background region and each of the other background regions as shown in Figure 2(b). More formally, if m refers to the moving object segment simultaneously involved in motion occlusion events with superpixels k and l then, $O_{k,m} = +1$ and $O_{l,m} = -1$, implies $O_{k,l} = +1$. This provides a strong depth-ordering cue between k and l. In addition, since k and l are not constrained to be adjacent superpixels, long-range edges between non-adjacent superpixels are also a result.

Monocular cues. We use monocular cues to provide evidence about occlusions for the other regions of the scene. Given the superpixel map for each frame, we use the work of Hoiem *et al.* [13] that uses learnt priors to determine which of two adjacent superpixels occludes the other. For each frame, we first update the occlusion matrix using the motion occlusion cues where available and update the matrix for all the other spatially adjacent superpixels using the monocular cues. We do not enforce transitivity here since the monocular cues are not as reliable as motion occlusion

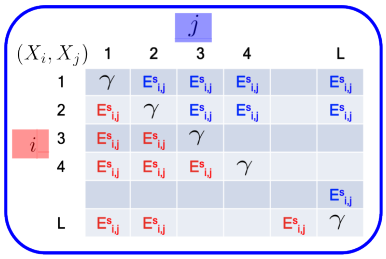


Figure 3: Spatial pairwise term E^S_{ij}. If i occludes j, the pairwise term will encourage that i takes a depth label closer (lower label) than j via a large penalty for the red terms and zero penalty for the blue terms. See Section 3.3 and Eqn 2 for details.

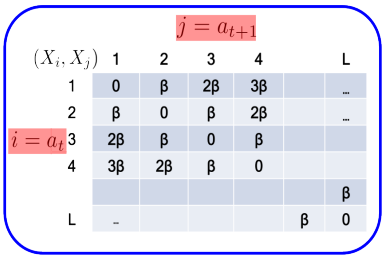


Figure 4: Temporal pairwise term E^T_{ij}. The penalty (β) increases as we go away from the diagonal encouraging a smooth motion of the object across depth layers. See Section 3.3.

cues. The occlusion matrix serves as the observations for modulating the terms of the energy function.

3.3. Energy minimization problem

The goal given the sparse pairwise depth-ordering constraints is to obtain dense depth-layers. One approach is a greedy algorithm where the whole scene starts at layer-0 and with every pairwise depth-ordering constraint regions of the scene are "pushed" and "popped" [3] to obtain the final labeling. Hoiem *et al*. [13] use a graph with boundaries between superpixels are nodes connected to adjacent boundaries to encourage continuity and closure. Jia *et al*. [16] use image junctions as nodes to obtain a globally consistent depth ordering using a minimum spanning tree. In this work, we use superpixels as nodes in the graph. This allows us to directly obtain the depth-layer labeling, and also incorporate long range edges between nodes.

We formulate depth layer segmentation as a discrete labeling problem where every superpixel is assigned a depth label $\{1, 2, \ldots, L\}$ where L is some pre-defined yet large set of discrete labels[1]. The labels are depth-ordered from closer to the camera moving away *i.e.* $\{1 < 2 < \cdots < L\}$. We formulate this multi-label segmentation problem as an energy minimization problem over the spatio-temporal graph obtained in the previous stage. The graph is a collection of $n + F$ nodes, where n nodes correspond to the background scene and F nodes correspond to the moving object with one node for the moving object for each of the F frames of the video. Our goal is to obtain a labeling $\mathcal{X} = \{X_1, X_2, \ldots, X_{n+F}\}$. We define an energy function over the graph as follows:

$$E(\mathcal{X}) = \sum_{i \in 1,\ldots,n+F} E_i(X_i) + \sum_{(i,j)\in\mathcal{N}_\mathcal{S}} E^S_{ij}(X_i, X_j) + \sum_{(i,j)\in\mathcal{N}_\mathcal{T}} E^T_{ij}(X_i, X_j) \quad (1)$$

where $E_i(X_i)$ is the unary term indicating the cost of assigning a depth layer to a node, $E^S_{ij}(X_i, X_j)$ is the spatial pairwise term updated by the motion occlusion cues and the monocular cues between interacting regions ($\mathcal{N}_\mathcal{S}$), and $E^T_{ij}(X_i, X_j)$ is the temporal pairwise term updated by the object motion model between the temporal edges ($\mathcal{N}_\mathcal{T}$).

Unary term (E_i). The unary term measures the cost of assigning a particular depth label to a node. We use a uniform likelihood across all labels since a node does not prefer one label over another. However, we note that the moving object can move between two background regions that are in adjacent depth layers. To address this, we ensure that the background regions only take odd or modulo-2 labels, which makes an intermediate layer between two depth layers available for the moving object. We do so using hard constraints where the background region pays infinite penalty for choosing an even numbered depth label.

Spatial pairwise term (E^S_{ij}). The spatial pairwise term encodes the pairwise depth-ordering observations we accumulate within the occlusion matrix. Consider two regions (nodes) i and j, using the cues we discussed in Section 3.2 let us suppose we know that region i is occludes region j *i.e.* $O_{i,j} = +1$. Intuitively, the pairwise term for the edge between i and j must encourage i to take a depth label that is smaller than (closer) j. To accomplish this, our pairwise term has the form of an lower triangular matrix where a large cost is incurred for region i taking a depth label larger than region j. We make this term contrast sensitive using the score from a coplanar classifier $(1.0 - \delta^f_{i,j})$ that indicates how likely i and j are coplanar using the relative region-level features similar to [21] for each frame f. More formally,

$$E^{S,f}_{ij}(X_i, X_j) = \begin{cases} -log\left(c^f_{ij} \times \frac{1+O^f_{i,j}+\epsilon}{2}\right) & \forall X_i < X_j \\ \gamma & X_i = X_j \\ -log\left(c^f_{ij} \times \frac{1+O^f_{j,i}+\epsilon}{2}\right) & \forall X_i > X_j \end{cases}$$

$$E^S_{ij}(X_i, X_j) = \sum_{f\in F}\left(E^{S,f}_{ij}(X_i, X_j) \times \exp\left(-\delta^f_{i,j}\right)\right) \quad (2)$$

where, $E^{S,f}_{ij}$ is the pairwise term for frame f, $O^f_{i,j}$ is the occlusion relationship between region i and j in frame f of the image sequence. c^f_{ij} is the confidence of the pairwise occlusion relationship for frame f. We set this value to 1.0 for edges that involve the moving object and use the occlusion strength [13] as the confidence score for all

[1] In all our experiments we set L = 40. An over-estimate of L allows for enough layers for the background scene. Increasing L beyond 40 did not affect performance but added to the computational complexity.

other edges. The summation over pairwise terms over all frames helps capture the evidence between two nodes over the whole sequence. The factor γ is a bias that keeps the solution away from the trivial solution of a single depth layer for the whole scene[2]. ϵ is a small value to maintain numerical precision. The form of the spatial pairwise term is illustrated in Figure 3.

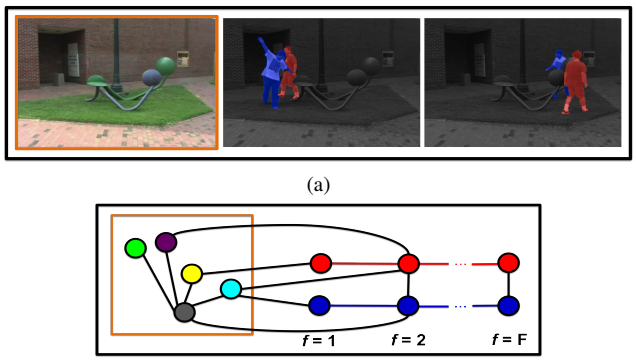
Temporal pairwise term (E_{ij}^T). The temporal pairwise term penalizes label disagreement between the moving object node across frames and encourages a smooth motion for the moving object, illustrated in Figure 4. The pairwise penalty is similar to the standard Pott's model, except with an increasing penalty (β) as we go away from the diagonal[3]. Given the depth label of the moving object in one frame, smooth motion is encouraged by making the node pay a lower cost to switch to nearby depth labels but larger penalty for more drastic changes in the depth label. Physically, this motion model assumes that the object does not abruptly change in depth as it moves through the scene.

3.4. Handling multiple moving objects

Here, we extend the formulation (single moving object) to handle multiple moving objects. Consider the example in Figure 5(a) with the region corresponding to the two moving objects in the scene shown in blue and red overlay. In case of k moving objects, we add k nodes (a node for each moving object) for each frame of the video. The resulting spatio-temporal graph for the example is shown in Figure 5(b). We have an edge between the moving objects as shown in the frame, $f = 2$ when the objects cross path. We obtain their pairwise depth-ordering using the cue that when the two objects are in contact, the taller object, *i.e.*, the object with a larger bounding box height, occludes the smaller one. This assumes that the moving objects are the same size in real world; however, more sophisticated classifiers could be used. We modify the unary term to reflect that there are multiple moving objects. In the single object case we used a modulo-2 representation of the depth labels that put hard constraints on the background regions to take only alternate depth labels allowing for the moving object to lie between two background region layers. In case of k moving objects in the scene we extend this to a modulo-(k+1) representation that allows the k objects to lie between two adjacent background region layers. Given this graph, the definition of the energy function is the same as Section 3.3.

3.5. Inference

In our energy function, each energy term by itself is weak. For instance, the unary term does not provide an affinity of a node towards a particular label but restricts the labels the background regions can take; the spatial pairwise term bounds the possible labels the adjacent node can take based on the label of the current node. However, the combination of these terms is powerful. The intuition behind the goal of inference is to find a depth labeling that satisfies as many pairwise interaction terms and motion model terms as possible. We perform inference using sequential tree-reweighted max-product message passing (TRW-S) [19]. The algorithm scales linearly with the number of frames and quadratically in the worst case with the number of superpixels (i.e., fully connected graph).



(a)

(b)

Figure 5: Multiple moving objects. (a) The background scene is shown in the orange bounding box. The two moving object segments for intermediate frames are overlaid in red and blue; (b) The spatio-temporal graph constructed. The spatial graph corresponding to the background scene is shown within the orange bounding box and the two nodes for each frame corresponding to the moving objects are shown using the red and blue nodes. See Section 3.4.

4. Experiments

In this section, we discuss the dataset, the evaluation metric, followed by our quantitative and qualitative results.

4.1. Dataset

Our first dataset (SET-A) contains 24 videos with a single moving object. 18 videos are from the publicly available multiview video dataset by Guan *et al.* [7] that include a person moving through the scene captured from multiple viewpoints. Each of these multiview videos serves as a test video for our scenario. The dataset has 6 additional videos with two clips from the movie 'Sound of Music'.

Our second dataset (SET-B) contains 9 videos from the publicly available video dataset by Guan *et al.* [7] with two people walking in the scene. In the single object scenario, the moving object segmentation and correspondence across frames was achieved using background subtraction, however, this is not trivial for multiple objects. While we believe that there is scope to leverage prior work on multiple object tracking to achieve this task automatically, in this work we provide correspondence and manually segment the moving objects on 30 frames for each video using GrabCut [27]. An

[2]We set the bias $\gamma = -log(0.5)$ for our experiments.
[3]$\beta = -log(0.5)$ for our experiments.

example is shown in Figure 5(a).

We manually obtain a pixel-level ground-truth depth layer segmentation for each of the background scenes using the depth-layer annotation tool by Hoiem *et al*. [13] and then map it to the background scene superpixel map by labeling all the pixels within a superpixel with the dominant label. An example is shown in Figure 6. We make all the data publicly available on our website[5].

It is worth pointing out that the ground surface has no clear 'ground-truth'. In particular, our instruction to the ground-truth annotator was that any object that stands on the ground surface occludes the ground surface as a basis for evaluations. Preprocessing to perform ground segmentation could be an alternate approach to add more semantics to the framework. However, this does not change the problem formulation or the improvement we obtain over the state-of-art.

4.2. Evaluation

We evaluate the performance of the algorithm as the accuracy of pairwise ordering between the regions of the background scene. Using the background superpixel map we translate the ground-truth depth layers into the ground-truth occlusion matrix (O^{gt}), which gives the pairwise depth-ordering between any pair of superpixels. Let the final occlusion matrix from the algorithm be O'. Given the two matrices, we evaluate the performance of the pairwise ordering between the superpixels by accumulating concordant pairs, discordant pairs, and compute the accuracy as[4],

$$\begin{aligned} &\text{Concordant pair } (i,j) : O^{gt}_{i,j} = O'_{i,j} \\ &\text{Discordant pair } (i,j) : O^{gt}_{i,j} \neq O'_{i,j} \\ &\text{Accuracy} = \frac{\#\text{Concordant pairs}}{\#\text{Concordant pairs} + \#\text{Discordant pairs}} \end{aligned} \tag{3}$$

The accuracy measure evaluates the performance of the algorithm over all pairs of regions in the scene. This gives an average score of 25.2% across our dataset even when the whole scene is given a single depth layer. We obtain a metric focused only on the occlusion boundaries by computing the precision and recall of the algorithm evaluating the fraction of recovered occlusion boundaries that are the true occlusion boundaries and the fraction of the true occlusion boundaries recovered by the algorithm, respectively.

Our problem is similar to that of inferring a rank ordered list of entries. We use two standard metrics to evaluate the performance of pairwise ordering, Kendall tau correlation coefficient (τ) and Kendall tau distance (τ_d) [18, 26]. In particular, we use the variant of Kendall's tau (Tau-b) that accounts for ties within the list, because pairs of superpixels can take the same depth label. τ measures the similarity between orderings and has range $[-1, +1]$, the higher the coefficient the better. τ_d is a measure of the distance between the orderings and has range $[0, 1]$, the lower the better.

[4] $\#x$ = number of x

(a) Background scene (b) Ground-truth (c) Estimated

Figure 6: (a) Background scene, (b) manually labeled ground-truth depth layers for the quantitative analysis and (c) estimated depth layers using our algorithm. White = near, black = far.

4.3. Quantitative results

We quantitatively evaluate the performance of our algorithm, comparing with several baselines. First, we compare with prior works that use only motion occlusion cues [3] or only monocular cues [13]. We then evaluate the performance of a naïve combination of the motion occlusion and monocular cues using a greedy algorithm similar to [3]. We first use all the motion occlusion cues to obtain the pairwise depth-ordering and then use the monocular cues to update the pairwise orderings only for adjacent superpixels that do not yet have a pairwise ordering constraint to obtain the final depth labeling. This baseline does not enforce a global consistency in combining the cues. In our full algorithm, we use a spatio-temporal graph to combine the two cues and enforce global consistency. In addition to evaluating the performance of the proposed algorithm (full), we evaluate a variant of the proposed algorithm where we drop the temporal links that enforce a smooth object motion.

Tables 1 and 2 summarize the results. We see that using motion occlusion cues alone (ROW-1) performs the worst, for two main reasons - fragmentation of the scene due to the greedy algorithm [3] and sparsity of the cues *i.e*., it only reasons about regions the object interacts with. Monocular cues (ROW-2) do better because it reasons about the whole scene and encourages global consistency with a graph model [13]. While the naïve combination of the cues (ROW-3) performs better than only motion occlusion cues, it performs poorly in comparison to using only monocular cues, due to fragmentation and lack of global consistency.

Even without temporal links (ROW-4), we outperform the baselines in each metric. This clearly indicates that our improvements are not based on tracking per se, and shows our algorithm is applicable to scenarios like time-lapse sequences. Finally, in both test sets, our full proposed approach (ROW-5), gives an additional boost in performance and significantly outperforms all the other algorithms in each metric. Across the datasets, the proposed algorithm achieved the best performance in 19 out of 24 videos in SET-A and 8 out of 9 videos in SET-B.

4.4. Qualitative results

We show qualitative results obtained using only motion occlusion cues, only monocular cues and the proposed algorithm in Figure 7. Figure 7(b) shows the ground-truth depth

Single moving object (SET-A)	Accuracy (%)	Precision (%)	Recall (%)	F-measure [0.0, 1.0]	Kendall tau coefficient [-1.0, 1.0]	Kendall tau distance [0.0, 1.0]
Only motion cues [3]	38.2	40.0	38.3	0.39	+0.01	0.40
Only monocular cues [13]	49.0	55.1	50.0	0.52	+0.15	0.33
Naïve [3] + [13]	42.1	46.3	38.8	0.42	+0.03	0.36
Proposed (No temporal)	**54.9**	**60.8**	**55.4**	**0.58**	**+0.33**	**0.26**
Proposed (Full)	**56.5**	**62.6**	**57.5**	**0.61**	**+0.36**	**0.24**

Table 1: Quantitative results and comparisons for the single moving object scenario (SET-A). Each measure is averaged across the videos in the dataset. ROW-1 shows the performance when we use only the motion occlusion cues [3]; ROW-2 shows the performance when we use only the learnt monocular cues [13]; ROW-3 shows the performance of a naïve combination of the motion occlusion and monocular cues; ROW-4 shows the performance of the proposed approach but without the temporal links enforcing the object motion model; Finally ROW-5 shows the performance of the full proposed approach that combines the motion occlusion and monocular cues into one framework. In summary, the proposed algorithm (in green) outperforms the other algorithms in each metric.

Multiple moving objects (SET-B)	Accuracy (%)	Precision (%)	Recall (%)	F-measure [0.0, 1.0]	Kendall tau coefficient [-1.0, 1.0]	Kendall tau distance [0.0, 1.0]
Only motion cues [3]	40.5	43.4	40.7	0.42	+0.02	0.37
Only monocular cues [13]	50.9	55.6	50.7	0.53	+0.20	0.35
Naïve [3] + [13]	45.1	54.2	43.0	0.48	+0.06	0.36
Proposed (No temporal)	**56.3**	**60.3**	**55.5**	**0.58**	**+0.30**	**0.26**
Proposed (Full)	**58.2**	**62.4**	**59.1**	**0.60**	**+0.33**	**0.24**

Table 2: Quantitative results and comparisons for the multiple moving objects scenario (SET-B). The rows are the same algorithms as Table 1. The proposed approach (in green) outperforms the other algorithms in each metric.

layers for each scene. We first observe the drawback of using only motion occlusion cues in Figure 7(c), such as the fragmentation in the labeling due to the greedy algorithm and the unknown layer for pixels untouched by the moving object (in blue). Using the monocular cues results in a better dense labeling but errors due to the image-based features exist, Figure 7(d). In contrast, the proposed algorithm achieves a better labeling of the scene as seen in Figure 7(e). In particular, we see that occlusion cues captured in the motion occlusion cues but missing in the monocular cues such as the tree occluding the background in ROW-1, the chair and box occluding the background in ROW-2, 4, the pillars in ROW-5 are all carried forward to improve the result using the proposed algorithm. Errors due to pairwise cues unseen by the moving object but present in the monocular cues are carried forward to the final result (ROW-3, 6). In ROW-6 the proposed algorithm favors smoothness instead of the excessive fragmentation found from the motion occlusion cues. The sensitive stage of the algorithm is foreground segmentation (background subtraction) especially in case of scene irregularities such as specular surfaces and thin structures (computer monitor in ROW-2 Figure 7), which can lead to errors in the sparse occlusion cues. In our work, we handle this using the MRF over all the regions and incorporate temporal dependency via smooth motion of the moving object. We make a joint solution given all the (soft) occlusion cues, reducing the errors in comparison with prior work that make hard decisions using occlusion cues.

5. Conclusions

We have presented an algorithm to combine the sparse, yet strong motion occlusion cues revealed by moving objects in a static scene along with monocular cues for occlusion reasoning in a unified framework. The proposed framework uses pairwise ordering cues that even extends to other algorithms to obtain monocular occlusion cues. The results show that the proposed approach improves the performance of prior approaches, and handles multiple objects moving in the scene. We make these manually labeled depth layers and the manual multiple object segmentation across frames publicly available, which are also useful in evaluating tasks such as multiple object co-segmentation[5] .

Acknowledgements. The authors thank Ashutosh Saxena for his useful feedback that helped improve this work. This work was partly supported by NSF DMS-0808864.

References

[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004. 3

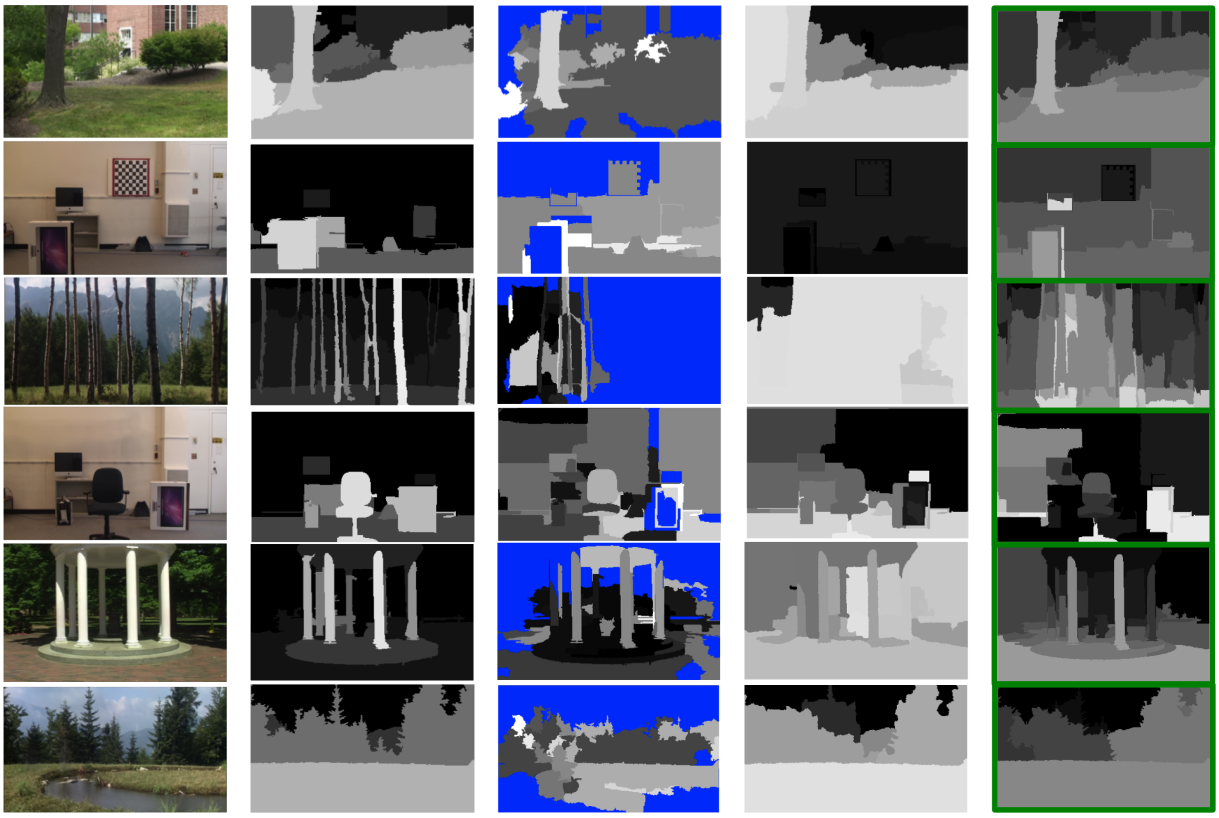
[2] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *PAMI*, 20(12):1222–1239, 2001. 3

[3] G. Brostow and I. Essa. Motion based decompositing of video. In *ICCV*, 1999. 1, 2, 3, 4, 6, 7, 8

[4] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002. 2

[5] D. Fouhey, V. Delaitre, I. Laptev, J. Sivic, A. Efros, and A. Gupta. People watching: Human actions as a cue for single view geometry. In *ECCV*, 2012. 2

[5] http://chenlab.ece.cornell.edu/projects/DepthLayersMRF/



(a) Background scene (b) Ground-truth layers (c) Only motion cues [3], pixels with no cues in blue (d) Only monocular cues [13] (e) Proposed algorithm

Figure 7: Qualitative results and comparisons. For all the above depth-layers, white = near, black = far. Discussion in Section 4.4.

[6] D. Greenhill, J. Renno, J. Orwell, and G. Jones. Occlusion analysis: Learning and utilising depthmaps in object tracking. In *BMVC*, 2004. 1

[7] L. Guan, J. Franco, and M. Pollefeys. Probabilistic multi-view dynamic scene reconstruction and occlusion reasoning from silhouette cues. In *IJCV*, 2010. 2, 5

[8] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 1, 2

[9] A. Gupta, S. Satkin, A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *CVPR*, 2011. 2

[10] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 1, 2

[11] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010. 2

[12] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *IJCV*, 2008. 1, 2

[13] D. Hoiem, A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. In *IJCV*, 2011. 1, 2, 3, 4, 6, 7, 8

[14] Y. Horry, K. Aniyo, and K. Arai. Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *SIGGRAPH*, 1997. 2

[15] Y. Huang and I. Essa. Tracking multiple objects through occlusions. In *CVPR*, 2005. 1

[16] Z. Jia, A. Gallagher, Y. Chang, and T. Chen. A learning based framework for depth ordering. In *CVPR*, 2012. 2, 4

[17] G. Kanizsa. Organization in vision: Essays on gestalt perception. In *Praeger*, 1979. 2

[18] M. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–93, 1938. 6

[19] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, Oct. 2006. 5

[20] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004. 3

[21] A. Kowdle, Y. Chang, A. Gallagher, and T. Chen. Active learning for piecewise planar 3d reconstruction. In *CVPR*, 2011. 4

[22] N. Krahnstoever and P. Mendonca. Bayesian autocalibration for surveillance. In *CVPR*, 2005. 1

[23] D. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010. 1, 2

[24] M. McGuire, W. Matusik, H. Pfister, . J. F. Hughes, and F. Durand. Defocus video matting. *SIGGRAPH.*, 2005. 1

[25] K. Nakayama. Biological image motion processing. In *Vision Research*, volume 25, pages 625–660, 1985. 2

[26] G. E. Noether. Why kendall tau? *Teaching Statistics*, 3(2):41–43, 1981. 6

[27] C. Rother, V. Kolmogorov, and A. Blake. "Grabcut" - Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 3, 5

[28] A. Saxena, S. Chung, and A. Ng. Learning depth from single monocular images. In *NIPS*, 2005. 1, 2

[29] A. Schodl and I. Essa. Depth layers from occlusions. In *CVPR*, 2001. 1, 2

[30] J. Ventura and T. H"ollerer. Depth compositing for augmented reality. In *SIGGRAPH*, 2008. 1

[31] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, 2010. 1, 2

[32] S. Yu, H. Zhang, and J. Malik. Inferring spatial layout from a single image via depth-ordered grouping. In *POCV Workshop*, 2008. 1, 2