Extending Layered Models to 3D Motion

Dong Lao and Ganesh Sundaramoorthi

KAUST, Saudi Arabia
{dong.lao,ganesh.sundaramoorthi}@kaust.edu.sa [**]

Abstract. We consider the problem of inferring a layered representation, its depth ordering and motion segmentation from video in which objects may undergo 3D non-planar motion relative to the camera. We generalize layered inference to that case and corresponding self-occlusion phenomena. We accomplish this by introducing a flattened 3D object representation, which is a compact representation of an object that contains all visible portions of the object seen in the video, including parts of an object that are self-occluded (as well as occluded) in one frame but seen in another. We formulate the inference of such flattened representations and motion segmentation, and derive an optimization scheme. We also introduce a new depth ordering scheme, which is independent of layered inference and addresses the case of self-occlusion. It requires little computation given the flattened representations. Experiments on benchmark datasets show the advantage of our method over existing layered methods, which do not model 3D motion and self-occlusion.

Keywords: motion / video segmentation, layered models

1 Introduction

Layered models are a powerful way to model a video sequence. Such models aim to explain a video by decomposing it into *layers*, which describe the shapes and appearances of objects, their motion, and a generative means to reconstructing the video. They also relate objects through their occlusion relations and depth ordering, i.e., the ordering of objects in front of each other with respect to the given camera viewpoint. Compared to dense 3D reconstruction from monocular video, which is valid for rigid scenes, layered approaches provide a computationally efficient intermediate 2D representation of (dynamic) scenes, which is still powerful enough for a variety of computer vision problems. Some of these problems include segmentation, motion estimation (e.g., tracking and optical flow), and shape analysis. Since all of the aforementioned problems are coupled, layered approaches provide a natural and principled framework to address these problems. Although such models are general in solving a variety of problems and have been successful in these problems, existing layered approaches are fundamentally limited as they are 2D and only model objects moving according to planar motions. Thus, they cannot cope with 3D motions such as rotation in depth and the

[**] Code available: https://github.com/donglao/layers3Dmotion

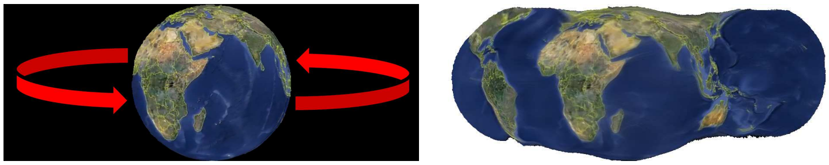


Fig. 1. Example flattened representation of the rotating earth. *The video sequence (left) shows the rotating earth. The flattened representation reconstructed by our algorithm is on the right. Notice that the representation compactly captures parts of the earth that are self-occluded in some frames, but visible in others.*

associated *self-occlusion* phenomena. Here, we define self-occlusion as the part of a 3D object surface that is not visible, in the absence of other objects, due to camera viewpoint. In this paper, we generalize layered models and depth ordering to self-occlusion generated from out-of-plane object motion and non-planar camera viewpoint change.

Specifically, our contributions are as follows. **1.** From a modeling perspective, we introduce *flattened 3D object representations* (see Fig. 1), which are compact 2D representations of the radiance of 3D deforming objects. These representations aggregate parts of the 3D object radiance that are *self-occluded* (and occluded by other objects) in some frames, but are visible in other frames into a compact 2D representation. They generalize layered models to enable modeling of 3D (non-planar) motion and corresponding self-occlusion phenomena. **2.** We derive an optimization algorithm within a variational framework for inferring the flattened representations and segmentation whose complexity grows linearly (as opposed to combinatorially) with the number of layers. **3.** We introduce a new global depth ordering method that treats self-occlusion, in addition to occlusion from other objects. The algorithm requires virtually no computation given the flattened representations and segmentation. It also allows for the depth ordering to change with time. **4.** Finally, we demonstrate the advantage of our approach in recovering layers, depth ordering and in segmentation on benchmark datasets.

1.1 Related Work

The literature on layered models for segmentation, motion estimation and depth ordering is extensive, and we highlight only some of the advances. Layers relate to video segmentation and motion segmentation (e.g., [1–6]) in that layered models provide a segmentation, and a principled means of dealing with occlusion phenomena. We are interested in more than just segmentation, i.e., a generative *explanation* of the video, which these methods do not provide. Since the problems of segmentation, motion estimation and depth ordering are related, many layered approaches are treated as a joint inference problem where the layers, motion and depth ordering are solved together. As the joint inference problem is difficult and a computationally intensive optimization procedure, early approaches (e.g., [7–15]) for layers employed low dimensional parametric motion models (e.g., translation or affine), which inherently limits them to planar motion.

Later approaches (e.g., [16–19]) to layers model motion of layers with fully non-parametric models based on optical flow (e.g., [20–24]), thus enabling 2D articulated motion and deformation. [16] formulates the problem of inferring layered representations as an extension of the classical Mumford and Shah segmentation problem [25–28], which provides a principled approach to layers. In [16] depth ordering is not formulated, but layers can still be inferred. Optimization, based on gradient descent was employed due to the non-convexity of the problem. While our optimization problem is similar to the framework there, their optimization method does not allow for self-occlusion. Later advances (e.g., [17, 18]) improved the optimization in the layer and motion inference. However the depth ordering problem, which is coupled with layered inference, is combinatorial in the number of layers, restricting the number of layers. [29, 30] aim to overcome the combinatorial problem by considering localized layers rather than a full global depth ordering. Within local regions there are typically few layers and it is feasible to solve the combinatorial problem. Further advances in optimization were achieved in [19], where the expensive joint optimization problem for segmentation, motion estimation and depth ordering are decoupled, resulting in less expensive optimization. There, depth ordering is solved by a convex optimization problem based on occlusion cues. While the aforementioned layered approaches have modeled complex deformation, they are all 2D and cannot cope with self-occlusion phenomena arising from 3D rotation in depth, which is present in realistic scenes. Thus, segmentation could fail when objects undergo non-planar motion. Our work extends layers to model such self-occlusion, and our depth ordering also accounts for this phenomena. While [31, 3] does treat self-occlusion, it only performs video segmentation not layered inference; we show out-performance against that method in experiments in video segmentation.

A recent approach to layers [30] uses semantic segmentation in images (based on the advances in deep learning) to improve optical flow estimation and hence the layered inference. Although our method does not integrate semantic object detectors, as the focus is to address self-occlusion, it does not preclude them, and they can be used to enhance our method, for instance in the initialization.

2 Layered Segmentation With Flattened Object Models

In this section, we formulate the inference of the flattened 3D object representations, and segmentation as an optimization problem.

2.1 Energy Formulation

We denote the image sequence by $\{I_t\}_{t=1}^T$ where $I_t : \Omega \to \mathbb{R}^k$ ($k = 3$ for the color channels), $\Omega \subset \mathbb{R}^2$ is the domain of the image, and T is the number of images. Suppose that there are N objects (including the “background” which includes all of the scene except the objects of interest), and denote by $R_i \subset \mathbb{R}^2$ the domain (shape) of the flattened 3D object representation for object i. We denote by $f_i : R_i \to \mathbb{R}^k$ the *radiance function* of object i defined in the flattened object

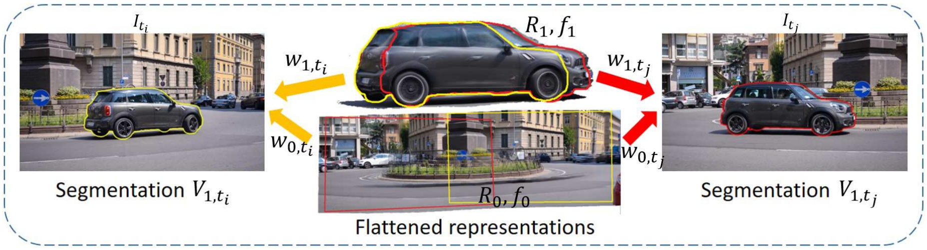


Fig. 2. Schematic of flattened representations and generation of images.

domain. f_i is a compact representation of all the appearances of the object i seen in the image sequence. The object appearance in any image can be obtained from the part of f_i visible in that frame. We define the *warps*, $w_{it} : R_i \to \Omega$, as the mapping from the flattened representation domain of object i to frame t. These will be diffeomorphisms (smooth and invertible maps) from the un-occluded portion of R_i to the segmentation of object i at time t. For convenience, they will be extended diffeomorphically to all of R_i. We denote by $V_{i,t} : \Omega \to [0, 1]$ the *visibility functions*, the relaxed indicator functions for the pixels in image t that map to the visible portion of object i. Finally, we let $\tilde{R}_{i,t} = \{V_{i,t} = 1\}$ be the domain of projected flattened object i that is visible in from t. See Figure 2.

We now define an energy to recover the flattened representation of each the objects, i.e., f_i, R_i, the warps $w_{i,t}$ and the visibility functions. The energy consists of two components, E_{app}, the appearance energy that is driven by the images, and E_{reg}, which contain regularity terms. The main term of the appearance energy aims to choose the flattened representations such that they can as close as possible reconstruct *each* of the images I_t by deforming the flattened representations by smooth warps. Thus, the appearance energy consists of a term that warps the appearances f_i into the image domains via the inverse of w_{it} and compares it via the squared error to the image I_t within $\tilde{R}_{it}$, the segmentations. The first term in the energy to be minimized is thus

$$E_{app} = \sum_{t,i} \int_{\tilde{R}_{it}} |I_t(x) - f_i(w_{it}^{-1}(x))|^2 \,\mathrm{d}x - \int_{\tilde{R}_{it}} \beta_t(x) \log p_i(I_t(x)) \,\mathrm{d}x. \qquad (1)$$

The second term above groups pixels by similarity to other image intensities, via local histograms (i.e., a collection of histograms that vary with spatial location) p_i for object i. The spatially varying weight β_t is small when the first term is reliable enough to group the pixel, and small otherwise. This term is needed to cope with noise: if a pixel back projects to a point in the scene that is only visible in few frames, the true appearance that can be recovered is unreliable, and hence more weight is placed on grouping the pixel based on similar intensities in the image. The weighting function β, will be given in the optimization section, as it will be easier to interpret there. Other terms could be used rather than the second one, possibly integrating semantic knowledge, but we choose it for its simplicity, as our main objective is in optimization of the first term.

The regularity energy E_{reg} consists of boundary regularity of the regions defined by the visibility functions and an area penalty on the domains of the flattened object models, and is defined as follows:

$$E_{reg} = \alpha \sum_{i,t} \text{Len}(\partial \tilde{R}_{i,t}) + \gamma \sum_{i} \text{Area}(R_i), \tag{2}$$

where $\alpha, \gamma > 0$ are weights, $\text{Len}(\partial \tilde{R}_{it})$ is the length of the boundary of $\tilde{R}_{it}$, which induces spatially regular regions in the images, and $\text{Area}(R_i)$ is the area of the domain of the object model. The last term, which can be thought of as a measure of compactness of the representation, is needed so that the models are compact as possible. Note that if that term is not included, a trivial (non-useful) solution to the full optimization problem is to simply choose a single object model that is a concatenation of all the images, the warps to be the identity, and the visibility functions to be 1 everywhere, which gives $E_{app} = 0$.

The goal is to optimize the full energy $E = E_{app} + E_{reg}$, which is a joint optimization problem in the shapes R_i and appearances f_i of the flattened objects, the warps w_{it}, and the visibility functions V_{it}.

Occlusion and Self-Occlusion: By formulating the energy with flattened object models, we implicitly address issues of both occlusion from one object moving in front of another, *and self-occlusion*, which are both naturally addressed and are not distinguished. The flattened model R_i, f_i contain parts of the projected object that are visible in one frame but not another. The occluded and self-occluded parts of the representation in frame t are the set $R_i \backslash w_{it}^{-1}(\{V_{it} = 1\})$. Considering only the first term of E_{app}, the occluded part of the R_i are the points that map to points x in which the squared error $|I_t(x) - f_i(w_{it}^{-1}(x))|^2$ is not smallest when compared to squared error from other flattened representations that map to the points x.

For the problem of flattened representation inference, distinguishing occlusion and is not needed. However, we eventually want to go beyond segmentation and obtain a depth ordering of objects, which requires distinguishing both occlusion (see Section 3). This separation of occlusion and self-occlusion allows one to see behind objects in images. See Fig. 6 where we visualize the flattened representation minus the self-occlusion, which shows the object(s) without other objects occluding them.

2.2 Optimization Algorithm

Due to non-convexity, our optimization algorithm will be a joint gradient descent in the flattened shapes, appearances, warps, and the visibility functions. We now show the optimization of each one of these variables, given the others are fixed and then give the full optimization procedure at the end.

Appearance Optimization: We optimize in f_i given estimates of the other variables. Notice that f_i appears only in the first term of E_{app}. We can perform

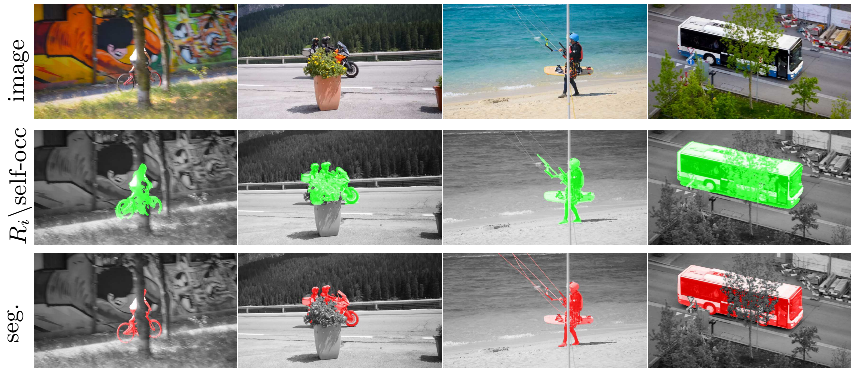


Fig. 3. Seeing Behind Occlusion from Other Objects. *From top to bottom: Original image, the flattened representation minus the self-occlusion, which removes occlusion due to other objects, and the object segmentation. Video segmentation datasets label the bottom as the segmentation, but the middle seems to be a natural object segmentation. Which should be considered ground truth?*

a change of variables of each of the integrals, and then differentiate the expression in $f_i(x)$, and solve for the global optimum of f_i, which gives that

$$f_i(x) = \frac{\sum_t I_t(w_{it}(x))V_{it}(w_{it}(x))J_{it}(x)}{\sum_t V_{it}(w_{it}(x))J_{it}(x)}, \quad x \in R_i, \tag{3}$$

where $J_{it}(x) = \det \nabla w_{it}(x)$ is the determinant of the Jacobian of the warp. The expression for f_i has a natural interpretation: the appearance at x is a weighted average of the images values at visible projections of x, i.e., $w_{it}(x)$, in the image domain. The weighting is done by area distortion of the mappings.

Shape Optimization: We optimize in the shape of the flattened region R_i by gradient descent, since the energy is non-convex in R_i. We first consider the terms in E_{app} and perform a change of variables so that the integrals are over the domains R_i. The resulting expression fits into a region competition problem [32], and we can use the known gradient computation there. One can show that the gradient with respect to the boundary ∂R_i is given by

$$\nabla_{\partial R_i} E = \sum_t \left[|\tilde{I}_{it} - f_i|^2 - |\tilde{I}_{jt} - \tilde{f}_j|^2 - \beta_t \log \frac{p_i(\tilde{I}_{it})}{p_j(\tilde{I}_{jt})} + \alpha\kappa_i \right] J_{it}\tilde{V}_i N_i + \gamma N_i, \tag{4}$$

where N_i is the unit outward normal to the boundary of R_i, $\tilde{I}_{it} = I_t \circ w_{it}$, $\tilde{V}_i = V_{it} \circ w_{it}$, $\tilde{f}_j = f_j \circ w_{jt}^{-1} \circ w_{it}$, and j, which is a function of x and t, is the layer adjacent to layer i in I_t. This optimization is needed so that the size and shape of the flattened representation can adapt to new self-occlusion discovered. This is a major distinction over [16], which although has a similar model to ours, by-passes this optimization and instead only optimizes the segmentation, which

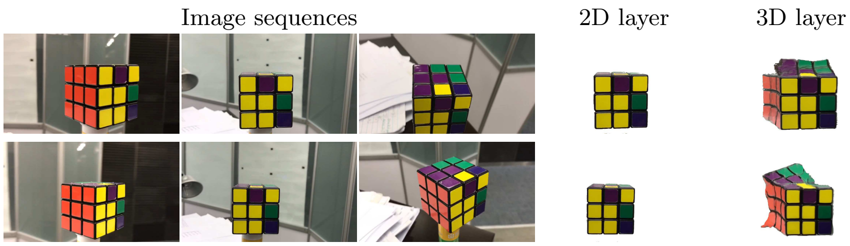


Fig. 4. Layered Inference of Rubix cube *Two different video sequences (top and bottom rows) of the same Rubix cube with different camera motion. [Last column]: Our flattened 3D representations capture information about the 3D structure (e.g., connectivity between faces of the Rubix cube) and motion, and includes parts of the object that are self-occluded. [Second last column]: Existing 2D layered models (result from a modern implementation of [16]) cannot adapt to 3D motion and self-occlusion.*

is equivalent in the case of no self-occlusion, but not otherwise. Thus, it cannot adapt to self-occlusion. See Fig. 4.

Visibility Optimization: We optimize in the visibility functions V_{it}, which form the segmentation, given the other variables. Note that the visibility functions can be determined from the corresponding projected regions $\tilde{R}_{it}$. We thus compute the gradient of the energy with respect to the boundary of the projected regions $\partial\tilde{R}_{it}$. This is a standard region competition problem. One can show that the gradient is then

$$\nabla_{\partial \tilde{R}_{it}} E = \sum_t \left[|I_t - \hat{f}_i|^2 - |I_t - \hat{f}_j|^2 - \beta_t \log \frac{p_i(I_t)}{p_j(I_t)} + \alpha\kappa_i \right] \tilde{N}_i, \quad x \in \partial \tilde{R}_{it} \tag{5}$$

where $\hat{f}_i = f_i(w_{it}^{-1}(x))$, $\tilde{N}_i$ is the normal to $\partial\tilde{R}_{it}$, and j is defined as before: it is the layer adjacent to i in I_t.

Warp Optimization: We optimize in the warps w_{it} given the other variables. Since the energy is non-convex, we use gradient descent. To obtain smooth, diffeomorphic warps, and robustness to local minima, we use Sobolev gradients [33, 34]. The only term that involves the warp w_{it} is the first term of the E_{app}. One can show that the Sobolev gradient G_{it} with respect to w_{it}, has a translation component $\text{avg}(G_{it}) = \text{avg}(F_{it})$ and a deformation component that satisfies:

$$\begin{cases} -\Delta \tilde{G}_{it}(x) = F_{it}(x) & x \in w_{it}(R_i) \\ \nabla \tilde{G}_{it}(x) \cdot \tilde{N}_i = |I_t - \hat{f}_i|^2 \tilde{V}_i \tilde{N}_i & x \in \partial w_{it}(R_i) \end{cases}, \quad F_{it} = \nabla \hat{f}_i [I_t - \hat{f}_i]^T \tilde{V}_i \tag{6}$$

where Δ denotes the Laplacian, and ∇ denotes the spatial gradient. The optimization involves updating the warp w_{it} iteratively by the translation until convergence, then one update step of w_{it} by the deformation $\tilde{G}_{it}$, and the pro-

Algorithm 1 *Layered optimization*

```
Input: Initialization for the flattened representations $R_i, f_i$
repeat // update the flattened representations, warps and segmentations
    For all $i$ and $t$, update $w_{it}$ performing gradient descent (6) until convergence
    For all $i$, compute $f_i$ by (3)
    For all $i$, update $R_i$ by one step in negative gradient direction (4)
    For all $t$, update the $V_{it}$ by one step in negative gradient direction (5)
until the energy $E$ converges
```

cess is iterated until convergence.

Initialization: The innovation in our method is the formulation and the optimization for flattened representations and self-occlusion, and we do not focus here on the initialization. Here we provide a simple scheme that we use in experiments, unless otherwise stated. From $\{I_t\}_{t=1}^T$, we compute frame-to-frame optical flow using [23] and then by composing flow, we obtain displacement $v_{t,T/2}$ between t and $T/2$. We use these as components in an edge-detector [35], which gives the number of regions and a segmentation in frame $T/2$. We then choose that segmentation as the initial flattened regions. One could use more sophisticated strategies, for instance, by using semantic object detectors.

Overall Optimization Algorithm: The overall optimization is given by Algorithm 1. Rather than evolving boundaries of regions, we evolve relaxed indicator functions of the regions, described in Supplementary. We now specify β_t in (1) as $\beta_t(x) = [\min_{j\sim i, j=i} \sum_{t'} V_{jt'}(w_{jt'} \circ w_{jt}^{-1}(x)) J_{jt'}(w_{jt}^{-1}(x))]^{-1}$ where $j \sim i$ denotes object j is adjacent to object i at x and $x \in \partial R_i$. β_t is the unreliability of the first term in E_{app}, defined as follows. We compute for each j, the number of frames t' the point x corresponds to a point in the flattened representation j that is visible in frame t'. To deal with distortion effects of the mapping, there is a weighting by $J_{jt'}$. Since the evolution depends on data from all j adjacent to i and i, we define the unreliability $\beta_t(x)$ as the inverse of the least reliable representation. Therefore, more times a point is visible, the more accurate the appearance model will be, and the more dependence on the first term in E_{app}, and the less dependence on local histograms.

3 Depth Ordering

In this section, we show how the depth ordering of the objects in the images can be computed from the segmentation and flattened models determined in the previous section. In the first sub-section, we assume that the object surfaces in 3D, their mapping to the imaging plane, and the segmentation in the image are known, and present a (trivial) algorithm to recover the depth ordering. Of course, in our problem, the objects in 3D are not available. Thus, in the next sub-section, we show how the previous algorithm can be used without 3D object surfaces by using the flattened representations and their mappings to the imaging plane as proxies for the 3D surfaces and their mappings to the image.

3.1 Depth Ordering From 3D Object Surfaces

We first introduce notation for the object surfaces and mappings to the plane, and then formalize *self-occlusion* and *occlusion* induced from other objects. These concepts will be relevant to our depth ordering algorithm, which we present following these formal concepts.

Notation and Definitions: Let $O_1, \ldots, O_N \subset \mathbb{R}^3$ denote N object surfaces in the 3D world that are imaged to form the image $I : \Omega \to \mathbb{R}^k$ at a given viewpoint at a given time. With abuse of notation we let V_i denote the segmentation (points in Ω of object i) in the image I. Based on the given viewpoint, the camera projection from points on the surface O_i to the imaging plane will be denoted w_{O_iI} and $w_{O_iI}^{-1}$ will denote the inverse of the mapping. We can now provide computational definitions for self-occlusion and occlusion induced by other objects, relevant to our algorithms. The **self-occlusion** (formed due to the viewpoint of the camera) is just the points of O_i (when all other objects are removed from the scene) that are not visible from the viewpoint of the camera. $w_{O_iI}(O_i)$ will denote the projection of non self-occluded points on O_i. The **occluded part** of object O_i induced by object O_j is $w_{O_iI}^{-1}(w_{O_iI}(O_i) \cap V_j)$. The **occlusion of** O_i induced by other objects (denoted by $O_{i,occ}$) is just the union of the occluded parts of O_i induced all other objects, which is given by ${w_{O_iI}}^{-1}(\cup_{j \neq i}(w_{O_iI}(O_i) \cap V_j))$.

Algorithm for Depth Ordering: We now present an algorithm for depth ordering. The algorithm makes the assumption that if any part of object i is occluded by object j, then any part of object j is not occluded by object i. This can be formulated as

Assumption 1 *For $i \neq j$, one of $w_{O_iI}(O_i) \cap V_j$ or $w_{O_jI}(O_j) \cap V_i$ must be empty.*

Under this assumption, we can relate the depth ordering of object i and j; indeed, $Depth(i) < Depth(j)$ (object i is in front of object j) in case $w_{O_jI}(O_j) \cap V_i \neq \emptyset$. This naturally defines the depth ordering of each objects ranging from 1 to N. Note that the depth ordering is not unique due to two cases, when both sets in the assumption above are empty. First, if the projections of two objects do not overlap ($w_{O_iI}(O_i) \cap w_{O_jI}(O_j) = \emptyset$) then no relation can be established and the ordering can be arbitrary. Second, if the overlapping part of the projections of two objects are fully occluded by another object ($w_{O_iI}(O_i) \cap w_{O_jI}(O_j) \subseteq V_k, k \neq i \text{ or } j$) then the depth relation between i and j cannot be established.

Under the previous assumption, we can derive a simple algorithm for depth ordering. Note that by definition of depth ordering, for object i satisfying $Depth(i) = 1$, we have that $\cup_{j \neq i} w_{OI}^{i}(O_i) \cap V_j = \emptyset$, which means that it is not occluded by any other object. Therefore, we can recover the object with depth 1. By removing that object from the scene, we can repeat the the same test and identify the

Algorithm 2 *Depth ordering given 3D surfaces*

```
1: Set $index = 1$;
2: Find $i$ satisfying $V_i = w^i_{OI}(O_i)$, label $Depth(i) = index$;
3: For all objects $j$ not labeled, let $V_j = V_j \cup (w^j_{OI}(O_j) \cap V_i)$;
4: $index = index + 1$, go to Step 2 until all objects are labeled
```

object with depth 2. Continuing this way, we can recover the depth ordering of all objects. One can effectively remove an object i from the scene in the image by removing V_i from the segmentation in image I and then augmenting V_j by the occluded part of object j induced by object i. Therefore we can recover the depth ordering by Algorithm 2.

3.2 Depth Ordering From Flattened Representations

We now translate the depth ordering algorithm assuming 3D surfaces in the previous section to the case of depth ordering with flattened representations. We define $w_{O_iR_i}$ to be the mapping from the surface O_i to the flattened representation R_i. Ideally, $w_{O_iR_i}$ is a one-to-one mapping, but in general it will be onto since the video sequence from which the flattened representation is constructed may not observe all parts of the object. By defining the mapping from the flattened representation to the image as $w_{R_iI} := w_{O_iI} \circ w^{-1}_{O_iR}$, the definitions of self-occlusion, occlusion induced by other objects, and the visible part of the object can be naturally extended to the flattened representation. By noting that $w^{-1}_{O_iR_i}(R_i) \subset O_i$, and under Assumption 1, we obtain the following property.

Statement 1 *At least one of $w_{R_iI}(R_i) \cap V_j$ and $w_{R_jI}(R_j) \cap V_i$ must be empty.*

This translates Assumption 1 to the mappings from flattened representations to the image. This statement allows us to similarly define a depth ordering as $w_{R_jI}(R_j) \cap V_i \neq \emptyset$ means $Depth(i) < Depth(j)$, as before. Therefore, we can apply the same algorithm in the previous section with w_{O_iI} replaced by w_{R_iI}.

In theory, the mappings w_{R_iI} only map the non-self occluded part of R_i to the image. However, in practice w_{R_iI} is computed from optical flow computation in Section 2.2, which maps the entire flattened region R_i to the image. The optical flow computation implicitly ignores data from the occluded (self-occluded as well as occlusion from other objects) part of the flattened representation through robust norms on the data fidelity, and extends the flow into occluded parts by extrapolating the warp from the visible parts. Near the self occluding boundary of the object, the mapping w_{O_iI} maps large surface areas to small ones in the image so that the determinant of the Jacobian of the warp becomes small. Since the warping w_{R_iI} from the flattened representation is a composition with w_{O_iI}, near the self-occlusion, the map w_{R_iI} maps large areas to small areas in the image. Since the optical flow algorithm extends the mapping near the self-occlusion into the self-occlusion, the self-occlusion is mapped to a small area (close to zero) in the image. Therefore, in Statement 1 rather than the condition

Algorithm 3 *Depth ordering from flattened representations*

```
1: Set $index = 1$
2: $i^* = \min_{i \text{ not labeled}} \text{Area}[w_{R_i I}(R_i) \backslash \cup_{j \text{ labeled}} V_j \backslash V_i]$
3: label $Depth(i^*) = index$
4: $index = index + 1$, go to Step 2 until all objects are labeled
```

that $w_{R_j I}(R_j) \cap V_i = \emptyset$ (object j is in front of object i), it is reasonable to assume that $w_{R_j I}(R_j) \cap V_i$ has small area (representing the area of the mapping of the self-occluded part of object j to V_i).

We can now extend the algorithm for depth ordering to deal with the case of $w_{R_i I}$ approximated with optical flow computation, based on the fact that self-occlusions are mapped to a small region in the image. To identify the object on top (depth ordering 1), rather than the condition $w_{R_i I}(R_i) \backslash V_i = \emptyset$, we compute the object i_1 such that $\text{Area}(w_{R_i I}(R_i) \backslash V_i)$ is smallest over all i. As in the previous algorithm, we can now remove the object with depth ordering 1, and again find the object i_2 that minimizes $\text{Area}(w_{R_i I}(R_i) \backslash V_{i_1} \backslash V_i)$ over all $i \neq i_1$. We can continue in this way to obtain Algorithm 3. Note that this allows one to compute depth ordering from only a single image, which allows the depth ordering to change with frames in a video sequence.

4 Experiments

In this section, we show the performance of our method on three standard benchmarks, one for layered segmentation, and the others for video segmentation.

MIT Human Annotated Dataset Results: MIT Human Annotated Dataset [36] has 10 sequences, and is used to test layered segmentation approaches and depth ordering. Results are reported visually. Both planar and 3D motion are present in these image sequences. We test our layered framework by using as initialization the human labeled ground truth segmentation of the first frame (not depth ordering). Fig. 5 presents the segmentation and depth ordering results. Our algorithm recovers the layers with high accuracy, and the depth ordering of the layers correctly in most of the cases.

DAVIS 2016 Dataset: The DAVIS 2016 dataset [37] dataset is a dataset focusing on video object segmentation tasks. Video segmentation is one output of our method, but our method goes further. The dataset contains 50 sequences ranging from 25 to 100 frames. In each frame the ground truth segmentation of the moving object versus the background is densely annotated. We run our scheme fully automatically initialized by the method described in Section 2.2.

Coarse-to-Fine Scheme for DAVIS and FBMS-59: The initialization scheme described in Section 2.2 often results in noisy results over time, perhaps missing segmentations in some frames. To clean up this noise, we first run our algorithm

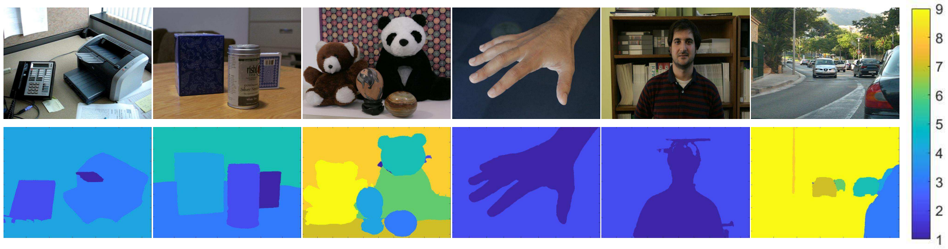


Fig. 5. Segmentation and Depth ordering in MIT dataset. *Multiple layers are extracted to obtain multi-label segmentation. Based on the segmentation result and extracted layers, Algorithm 3 is applied to compute depth ordering. In most cases the depth ordering are inferred correctly. Note that due to the ambiguity of the depth ordering, in some cases ground truth depth ordering does not exist. Layers in the front are indicated by small values of depth.*

with this initialization in small overlapping batches (of size 15 frames) of the video. This fills in missing segmentations. We then run our algorithm with this result as initialization on the whole video. This integrates coarse information across the whole video. Finally, to obtain finer scale details, we again run our algorithm on overlapping small batches (of size 7 frames). We iterate the last two steps to obtain our final result. Table 1 shows the result of these stages (labeled initialization, 1st, 2nd, 3rd, and the final result is labeled "ours") on DAVIS.

Method	Initial	1st	2nd	3rd	[16]	[19]	[3]	[38]	Ours
J mean	0.491	0.571	0.644	0.673	0.615	0.514	0.625	0.625	**0.683**
J recall	0.575	0.629	0.745	0.766	0.715	0.581	0.743	0.700	**0.777**
J decay	0.097	0.050	0.064	0.069	**0.041**	0.127	0.110	-	0.069
F mean	0.509	0.575	0.622	0.651	0.593	0.490	0.593	0.593	**0.672**
F recall	0.550	0.637	0.737	0.738	0.695	0.578	0.691	0.662	**0.759**
F decay	0.089	**0.064**	0.075	0.082	0.070	0.128	0.118	-	0.082

Table 1. Evaluation of Segmentation Results on DAVIS. *From left to right: result after our initialization, result after the 1st stage of our coarse-to-fine layered approach (see text for an explanation), result after our 2nd stage, result after our 3rd stage of coarse-to-fine, results of competing methods, and finally our final result after the last stage of our coarse-to-fine scheme.*

Comparison on DAVIS: We compare to a modern implementation of the layered method [16], which is a equivalent to our method if the shape evolution of the flattened representation is not performed. We also compare to [19], which is another layered method based on motion. We also include in the comparison non-layered approaches [3], which addresses the problem of self-occlusion in motion segmentation, and [38], which is another motion segmentation approach.

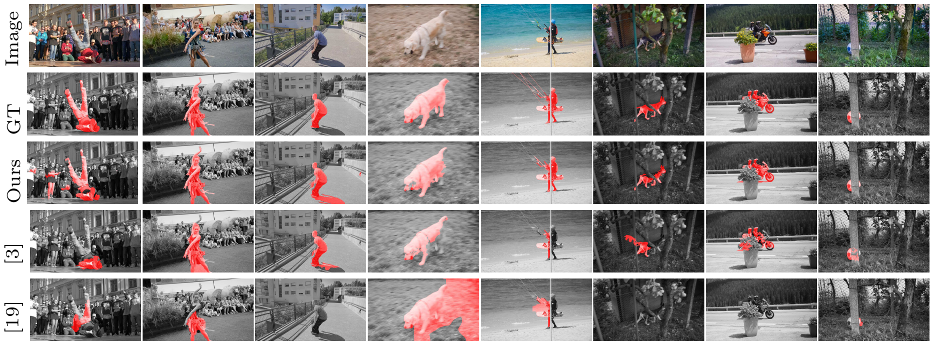
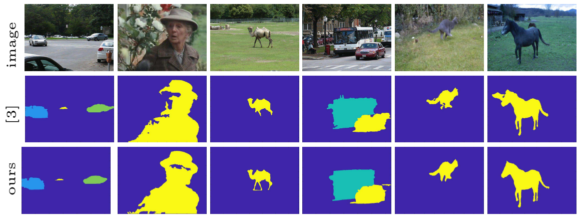


Fig. 6. Qualitative Comparison on DAVIS. *From left to right: (images 1-3): sequences with 3D motion inducing self-occlusion, (image 4): sequence with object color similarity to background, and (images 5-8): sequences with occlusion by other objects. Our layered segmentation successfully captures the object all of the sequence cases. In (1-3) [19], a layered approach, fails due lack of 3-D motion modeling; in (4) color similarity leads to wrong labeling in both [19, 3] due to reliance on intensity similarities. In (5-8) [19, 3] fail due to inability to deal with objects moving behind others.*

Qualitative comparison of the methods can be found in Fig. 6 and quantitative comparison can be found in Table 1. Quantitatively, our method outperforms all comparable motion-based approaches. Note the that the state-of-the-art approaches on this dataset use deep learning and are trained on large datasets (for instance, Pascal), however, they only perform segmentation and do not give a layered interpretation of the video and they are applicable to only binary segmentation, and they cannot be adapted to multiple objects. Our method requires no training data and is low-level, and comes close to the performance of these deep learning approaches. In fact, in 15/50 sequences, our method performs the best more than any other method.

FBMS-59 Dataset: To test our method on inferring more than two layered representations, we test our method on the FBMS-59 Dataset, which is used for benchmarking video segmentation algorithms. The test set of FBMS-59 contains 30 sequences with 69 labeled objects, and the the number of frames range from 19 to 800. Ground truth is given on selected frames. We compare to [3] that is a video segmentation that handles self-occlusion but not layers (discussed in the previous section), the layered approach [19], and other motion segmentation approaches. Quantitative results and representative results are shown in Fig. 7. They show that our method has the best results among these methods, and shows a slight improvement over [3], with the additional advantage that our method gives a layered representation, more powerful than just a segmentation.

Parameters: Our algorithm has few parameters, i.e., the parameter γ, which is the weight on penalizing the area of the flattened representation, and α, which



Methods	F	P	R	N
ours	**76.2**	**90.4**	**65.9**	**28**
[3]	75.9	89.8	65.8	**28**
[4]	74.1	86.0	65.1	23
[19]	68.3	82.4	58.4	17
[2]	66.7	74.9	60.1	20
[39]	62.0	79.6	50.7	7

Fig. 7. Results (qualitative and quantitative) and comparison on FBMS-59.

is the weight on the spatial regularity of the segmentation. These parameters are not sensitive. The values chosen in the experiments were $\gamma = 0.1$ and $\alpha = 2$.

Computational Cost: Our algorithm is linear in the number of layers (due to the optical flow computation for each layer). For 2 layers and 480p video with 30 frames, our entire coarse-to-fine scheme runs in about 10 mins with a Matlab implementation, on a standard modern processor.

5 Conclusion

We have generalized layered approaches to 3D planar motions and corresponding self-occlusion phenomena. This was accomplished with an intermediate 2D representation that concatenated all visible parts of an object in a monocular video sequence into a single compact representation. This allowed for representing parts that were self-occluded in one frame but visible in another. Depth ordering was formulated independent of the inference of the flattened representations, and is computationally efficient. Results on benchmark datasets showed that the advantage of this approach over other layered works. Further, increased performance was shown in the problem of motion segmentation over existing layered approaches, which do not account for 3D motion.

A limitation of our method is that is dependent on the initialization, which remains an open problem, although we provided a simple scheme. More advanced schemes could use semantic segmentation. Another limitation is in our representation, in that it does not account for all 3D motions and all self-occlusion phenomena. For instance, a person walking, the crossing of legs cannot be captured with a 2D representation (our method accounted for this case on datasets since the number of frames used was small enough that legs did not fully cross). A solution would be to infer a 3D representation of the object from the monocular video, but this could be expensive computationally, and it is valid for only rigid scenes. Our method trades off between complexity of a full 3D representation and its modeling power: although it does not model all 3D situations, it is a clear advance over existing layered approaches, without the complexity of a 3D representation and its limitation to rigid scenes. Another limitation is when Assumption 1 is broken (e.g., a hand grasping an object), in which our depth ordering would fail, but the layers are still inferred correctly.

References

1. Cremers, D., Soatto, S.: Motion competition: A variational approach to piecewise parametric motion segmentation. International Journal of Computer Vision **62**(3) (2005) 249–265
2. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. IEEE transactions on pattern analysis and machine intelligence **36**(6) (2014) 1187–1200
3. Yang, Y., Sundaramoorthi, G., Soatto, S.: Self-occlusions and disocclusions in causal video object segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 4408–4416
4. Keuper, M., Andres, B., Brox, T.: Motion trajectory segmentation via minimum cost multicuts. In: Computer Vision (ICCV), 2015 IEEE International Conference on, IEEE (2015) 3271–3279
5. Tokmakov, P., Alahari, K., Schmid, C.: Learning video object segmentation with visual memory. arXiv preprint arXiv:1704.05737 (2017)
6. Jain, S., Xiong, B., Grauman, K.: Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. arXiv preprint arXiv:1701.05384 (2017)
7. Wang, J.Y., Adelson, E.H.: Representing moving images with layers. IEEE Transactions on Image Processing **3**(5) (1994) 625–638
8. Darrell, T., Pentland, A.: Robust estimation of a multi-layered motion representation. In: Visual Motion, 1991., Proceedings of the IEEE Workshop on, IEEE (1991) 173–178
9. Hsu, S., Anandan, P., Peleg, S.: Accurate computation of optical flow by using layered motion representations. In: Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on. Volume 1., IEEE (1994) 743–746
10. Ayer, S., Sawhney, H.S.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In: Computer Vision, 1995. Proceedings., Fifth International Conference on, IEEE (1995) 777–784
11. Bergen, L., Meyer, F.: Motion segmentation and depth ordering based on morphological segmentation. In: European Conference on Computer Vision, Springer (1998) 531–547
12. Jojic, N., Frey, B.J.: Learning flexible sprites in video layers. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Volume 1., IEEE (2001) I–I
13. Smith, P., Drummond, T., Cipolla, R.: Layered motion segmentation and depth ordering by tracking edges. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(4) (2004) 479–494
14. Kumar, M.P., Torr, P.H., Zisserman, A.: Learning layered motion segmentations of video. International Journal of Computer Vision **76**(3) (2008) 301–319
15. Schoenemann, T., Cremers, D.: A coding-cost framework for super-resolution motion layer decomposition. IEEE Transactions on Image Processing **21**(3) (2012) 1097–1110
16. Jackson, J.D., Yezzi, A.J., Soatto, S.: Dynamic shape and appearance modeling via moving and deforming layers. International Journal of Computer Vision **79**(1) (2008) 71–84

17. Sun, D., Sudderth, E.B., Black, M.J.: Layered segmentation and optical flow estimation over time. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1768–1775
18. Sun, D., Wulff, J., Sudderth, E.B., Pfister, H., Black, M.J.: A fully-connected layered model of foreground and background flow. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE (2013) 2451–2458
19. Taylor, B., Karasev, V., Soatto, S.: Causal video object segmentation from persistence of occlusions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4268–4276
20. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3) (1981) 185–203
21. Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer vision and image understanding **63**(1) (1996) 75–104
22. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: European conference on computer vision, Springer (2004) 25–36
23. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 2432–2439
24. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (2017)
25. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Communications on pure and applied mathematics **42**(5) (1989) 577–685
26. Tsai, A., Yezzi, A., Willsky, A.S.: Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. IEEE transactions on Image Processing **10**(8) (2001) 1169–1186
27. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. International journal of computer vision **50**(3) (2002) 271–293
28. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the mumford-shah functional. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE (2009) 1133–1140
29. Sun, D., Liu, C., Pfister, H.: Local layering for joint motion estimation and occlusion detection. (2014)
30. Sevilla-Lara, L., Sun, D., Jampani, V., Black, M.J.: Optical flow with semantic segmentation and localized layers. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 3889–3898
31. Yang, Y., Sundaramoorthi, G.: Modeling self-occlusions in dynamic shape and appearance tracking. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 201–208
32. Zhu, S.C., Yuille, A.: Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. IEEE transactions on pattern analysis and machine intelligence **18**(9) (1996) 884–900
33. Yang, Y., Sundaramoorthi, G.: Shape tracking with occlusions via coarse-to-fine region-based sobolev descent. IEEE transactions on pattern analysis and machine intelligence **37**(5) (2015) 1053–1066

34. Sundaramoorthi, G., Yezzi, A., Mennucci, A.: Coarse-to-fine segmentation and tracking using sobolev active contours. IEEE Trans. Pattern Anal. Mach. Intell. **30**(5) (2008) 851–864
35. Dollár, P., Zitnick, C.L.: Fast edge detection using structured forests. IEEE transactions on pattern analysis and machine intelligence **37**(8) (2015) 1558–1570
36. Liu, C., Freeman, W.T., Adelson, E.H., Weiss, Y.: Human-assisted motion annotation. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
37. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Computer Vision and Pattern Recognition. (2016)
38. Wehrwein, S., Szeliski, R.: Video segmentation with background motion models. In: British Machine Vision Conference. (2017)
39. Ayvaci, A., Soatto, S.: Detachable object detection: Segmentation and depth ordering from short-baseline video. IEEE transactions on pattern analysis and machine intelligence **34**(10) (2012) 1942–1951