

Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes

Rui Yu, Chris Russell, and Lourdes Agapito

Abstract—Consider a video sequence captured by a single camera observing a complex dynamic scene containing an unknown mixture of multiple moving and possibly deforming objects. In this paper we propose an unsupervised approach to the challenging problem of simultaneously segmenting the scene into its constituent objects and reconstructing a 3D model of the scene. The strength of our approach comes from the ability to deal with real-world dynamic scenes and to handle seamlessly different types of motion: rigid, articulated and non-rigid. We formulate the problem as hierarchical graph-cut based segmentation where we decompose the whole scene into background and foreground objects and model the complex motion of non-rigid or articulated objects as a set of overlapping rigid parts. We evaluate the motion segmentation functionality of our approach on the Berkeley Motion Segmentation Dataset. In addition, to validate the capability of our approach to deal with real-world scenes we provide 3D reconstructions of some challenging videos from the *YouTube-Objects* and KITTI dataset.

Index Terms—Monocular, Dynamic Scenes, Piecewise Reconstruction



1 INTRODUCTION

With the emergence of video cameras on phones and laptops and the rise of video libraries (e. g. *YouTube Action*, *YouTube Objects*) the use of 3D information for recognition tasks has experienced a resurgence. While structure from motion (SfM) techniques exist that can reliably reconstruct a static scene, most scenes of interest contain multiple moving objects or even articulated or non-rigid objects. Motion segmentation and non-rigid scene reconstruction from monocular video have become more important than ever. This paper proposes a refocusing of 3D reconstruction towards reconstructing videos of dynamic scenes.

Substantial progress has been made in multibody SfM and non-rigid structure from motion (NRSfM) for dealing with dynamic scenes [25], [30] or creating vivid life-like reconstructions of deformable objects [14]. However, dynamic scene reconstruction still remains a significant challenging problem and is far from solved. On the one hand, multibody SfM approaches can segment the scene into multiple **rigid** moving objects, but they cannot deal with the presence of deformable or articulated objects in the scene. On the other hand, although NRSfM algorithms can reconstruct a single pre-segmented **deformable** surface moving in front of a camera [14], [39], they require segmentation of the scene into background and foreground objects. To be able to deal with complex dynamic scenes, which may contain a mixture of rigid objects, non-rigid motion and articulated objects, a powerful framework is needed which combines the advantages of both multibody SfM and NRSfM. In other words, a framework is needed that can segment the scene into different moving objects and reconstruct them separately, in a unified way, regardless of whether objects are rigid, non-rigid or articulated.

Recently, piecewise methods have emerged as a promising technique for seamlessly handling general dynamic scenes by modelling all objects (rigid, non-rigid or articulated) in the scene as a combination of rigid parts. Piecewise approaches to non-rigid and articulated reconstruction have been successfully applied to explain the complex motion of 2D tracks on a single non-rigid surface or an articulated object as a network of *overlapping* parts [11], [31], [39]. However, if naively applied to an entire scene with foreground/background objects occluding one another, depth boundaries between objects would not be respected and neighbouring models in the image would be forced to overlap irrespective of whether or not they belong to the same physical object.

The main contribution of this work is to offer a solution to the problem of *scene reconstruction* for real-world dynamic monocular videos that deals seamlessly with the presence of multiple non-rigid, articulated or pure rigid motion. In an entirely unsupervised approach, we reorganise/segment the scene into a constellation of object parts, recognise which parts are likely to constitute objects, join them together, and reconstruct the scene. We offer solutions to some of the problems of previous approaches to dynamic scene reconstruction: (i) Our approach is able to adapt the topology of the neighbourhood graph by breaking edges where necessary to preserve boundaries between objects. In this way our approach can deal with an entire scene where objects might occlude one another and not just pre-segmented objects; (ii) Our work results in a hierarchical approach to dynamic scene analysis. At the higher level of the hierarchy the scene is explained as a set of *objects* that are detached from the background and from each other. At the lower level of the hierarchy, each *object* can be explained as a set of overlapping *parts* that can model more complex motion.

2 RELATED WORK

Most works in dynamic scene reconstruction [8], [12], [32], [25] follow a pipeline approach where first feature point tracks or dense

- Rui Yu, Chris Russell and Lourdes Agapito are with the Department of Computer Science, University College London.
- E-mail: r.yu@cs.ucl.ac.uk
E-mail: c.russell@cs.ucl.ac.uk
E-mail: l.agapito@cs.ucl.ac.uk

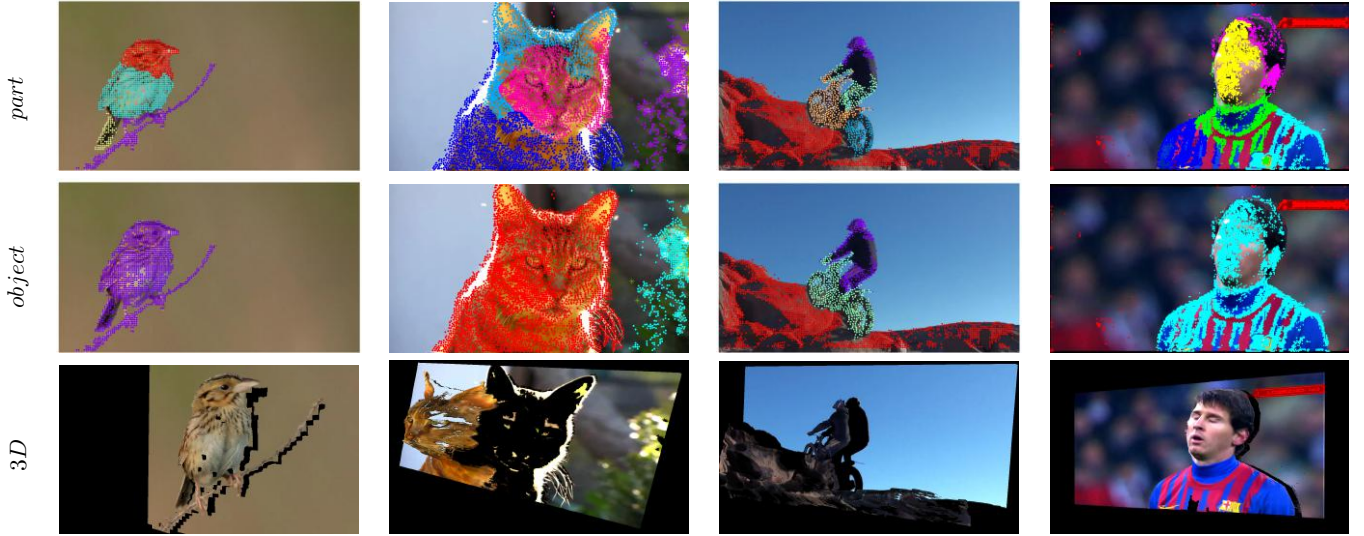


Fig. 1: Segmentation and 3D reconstruction results of three dynamic sequences of the *Youtube-Objects* Dataset [27] and a football sequence downloaded from YouTube. **Top:** segmentation into *parts* (rigid models). **Middle:** segmentation into *objects*. **Bottom:** densified 3D video pop-up from a novel viewpoint. The *bird* and *cat* sequence are non-rigid sequences occluding a static background. The *motorbike* sequence, acquired with a moving camera, shows articulated motion. Bottom row shows a reconstruction of football footage.

optical flow is estimated, followed by a motion segmentation step to cluster trajectories into different independent motions before 3D reconstruction is applied independently to each of the objects. The first approach to **multibody reconstruction** [8] extended the classic affine factorisation algorithm for static scenes [36] to the case of multiple independently moving rigid objects. While the original approach [8] was unable to deal with dependencies in the motions it was later extended to deal with degenerate [44] and articulated motions [38], [42]. More recent approaches to multibody SfM such as Ozden *et al.* [25] are able to perform simultaneous tracking, segmentation and reconstruction of a few feature points on realistic sequences. Roussos *et al.* [30] proposed a dense approach to multibody SfM in which depth values are estimated for every pixel in the image. However, none of these approaches can deal with non-rigidity or articulation in each of the objects which are assumed to be rigid.

Providing robust solutions to **video and motion segmentation** is a fundamental problem in computer vision and often a preliminary step towards 3D reconstruction. A wealth of motion segmentation algorithms for multi-rigid scenes have been proposed including algebraic frameworks such as GPCA [40] and methods that can deal with noise and outliers [29]. Motion segmentation has also been cast as a motion subspace clustering problem, first applied to the affine camera case [9], [19] and later extended to the case of perspective scenes [22]. Approaches such as Brox and Malik’s [7] exploit the consistency of point trajectories over time and can deal with non-rigid motion. On the other hand, superpixel [13] and supervoxel [41] methods for video segmentation can produce high quality video over-segmentations that respect object boundaries, are temporally consistent and are aligned with objects. However, since their aim is to segment non-rigid and articulated objects as a single segment, they are not appropriate for piecewise 3D reconstruction.

Non-rigid structure from motion (NRSfM) approaches reconstruct 3D models of non-rigid objects from monocular video,

typically by fitting a global low-rank shape model [37], [26] to 2D tracks. Piecewise reconstruction has also been successfully applied to NRSfM [31], [39] and articulated reconstruction [11] by fitting local models. However all existing methods can only reconstruct a single presegmented object and can not resolve entire scenes.

Our approach is most closely related to the paradigm of *multiple model fitting* where tracks, that might contain outliers, belong to an unknown number of models. The assignment of tracks to models and the estimation of model parameters is then optimised jointly [18], [31] to minimise a geometric cost subject to the constraint that neighbouring tracks must belong to the same model. The cost also incorporates a minimum description length (MDL) cost that prefers sparse solutions. The cost function is optimised by alternating between a discrete graph-cuts algorithm to solve the labeling problem and continuous optimisation to update the model parameters. This approach has previously been applied to computer vision problems such as stereo [3]; motion segmentation [18]; 3D reconstruction of non-rigid [31] and articulated objects [11]; and multi-body reconstruction [30].

Our approach departs from previous work in geometric multiple model fitting in multiple ways: (i) Our model offers segmentation at two granularities: object-level and part-level. At the object-level, we segment the scene into a small number of disjoint *objects*. At the part-level, objects are further divided into a set of overlapping parts; (ii) Our model uses a combination of appearance and geometry cues for segmentation which encourages salient foreground objects to be separated accurately from the background even when the motion is not distinctive enough; (iii) Our geometric cost uses a perspective camera model and is able to deal with perspective effects and incomplete tracks.

3 PROBLEM FORMULATION

We consider a monocular video sequence, possibly downloaded from the web, captured by a single camera observing a complex

dynamic scene that contains an unknown mixture of multiple moving and possibly deforming objects. First, we extract a set $\mathcal{T} = [1, \dots, T]$ of feature point tracks using Sundaram *et al.*'s publicly available code [35]. Although the tracker aims to provide long-term video correspondences, the length of tracks is variable and not all points tracked are visible in all the frames. We make no assumptions about the number of objects in the scene or their motions which could be rigid, articulated or non-rigid. Our goal is to estimate the 3D coordinates for all feature points in every frame.

3.1 Piecewise Overlapping Models

The works [31], [11] proposed a novel piecewise approach to the problem of 3D reconstruction of non-rigid objects. Rather than attempting to reconstruct objects by fitting a global low-rank shape model [37], [26] that is sufficiently expressive to capture deformations, but sufficiently low-rank to discourage overfitting, they automatically segmented the object to be reconstructed into a set of parts, each of which could be expressed by a simple model – either local rigid reconstructions [11] or local quadratic deformations [31]. By forcing these parts to overlap, and to agree about the reconstruction of the region of overlap, per part depth/scale and sign-flip ambiguities can be resolved. Figure 2 shows an illustration of the segmentation of an articulated object into overlapping rigid parts.

The problem was formulated as a labelling one where the assignment of tracks to models and the fitting of models to tracks were jointly optimised to minimise a geometric fitting cost subject to the spatial constraint that neighbouring tracks should also belong to the same model.

3.1.1 Assignment of point tracks to models

Let $\mathcal{T}$ refer to a set of point tracks and $\mathcal{M}$ a set of models. We use the notation $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ to refer to a labelling, where $\mathbf{x}_i$ is the *set* of models assigned to track i . Assuming a known topology, or graph, which connects tracks together in a neighbourhood structure $\mathcal{N}$, the following objective was proposed by [11], [31]

$$C(\mathbf{x}) = \sum_{i \in \mathcal{T}} \sum_{m \in \mathbf{x}_i} U_i(m) + \text{MDL}(\mathbf{x}) \quad (1)$$

where tracks are allowed to belong to multiple models in $\mathcal{M}$. The unary term $U_i(m)$ is the cost of assigning track i to model m and the term $\text{MDL}(\mathbf{x})$ is a label cost that encourages sparse solutions. In [11] local rigid models were used where each model was parameterised with the rotation and translation associated with a rigid motion and the unary cost $U_i(m)$ was defined as the image reprojection error under orthographic projection for that point given the model parameters. The optimisation of (1) was subject to the constraint that each track must be an interior point of some model i.e. that for every track there is a model such that that track and all its neighbours belong to that model (Figure 2 illustrates the concept of *interior point*), or more formally $\forall i \exists \alpha : \alpha = I_i$ and

$$I_i = \alpha \rightarrow \forall j \in N_i, \alpha \in \mathbf{x}_j \quad (2)$$

where $\mathbf{I} = \{I_1, I_2 \dots I_T\}$ refers to the assignment of each track i to the interior of one model I_i and N_i is the neighbourhood of track i . Russell *et al.* [31] showed how this problem could be formulated as a labeling problem over the assignment of tracks to the interior of models and efficiently solved using a novel

variant of α -expansion. Starting from an excess of models the optimisation followed a hill climbing approach that alternates between assigning tracks to models, and refitting the models to minimise the geometric error (image reprojection error).

3.2 Obstacles to reconstruction in the wild

Although these multiple model fitting approaches based on overlapping models do provide a robust approach to non-rigid [31] and articulated [11] reconstruction, they have shortcomings. First, they cannot deal with whole scenes in which the neighbourhood graph maintains connections between tracks of different objects (see Figure 2) – the constraints (2) combined with a bad neighbourhood structure can force parts to straddle multiple objects, leading to an error that can not be recovered from. Secondly, the unary terms of [11], [31] minimise a geometric cost based on multiview affine factorisation. Therefore, they have difficulty dealing with incomplete tracks. In real-world videos, tracks are likely not to persist for a large number of frames. Finally, a further limitation of the above approaches comes from the fact that only motion cues are used for the segmentation. Combining motion and appearance cues is useful to encourage object boundaries to be respected. Besides, these cues complement each other particularly if there are frames in the sequence with small motion.

The main contribution of our work is to offer solutions to these three limitations: (i) Our approach adapts the topology of the neighbourhood graph by breaking edges where necessary to preserve boundaries between objects. This allows our approach to deal with complete video footage where objects might occlude one another and not just singular pre-segmented objects. (ii) Our geometric unary cost is based on frame-to-frame fundamental matrices, an approach that leads itself naturally to handling incomplete tracks. (iii) Our data term combines geometric and appearance costs. We use the saliency score provided by [33] to encourage parts of similar saliency to belong to the same object.

4 SCENE RECONSTRUCTION

We propose a novel cost that allows us to modify the topology of the original neighbourhood by deleting edges between point tracks that belong to different physical objects, and should not overlap. Our new cost has four terms

$$C(\mathbf{x}) = E_{\text{data}} + E_{\text{edge_break}} + E_{\text{sparse}} + E_{\text{mdl}} \quad (3)$$

$$= \sum_{i \in \mathcal{T}} \sum_{m \in \mathbf{x}_i} U_i(m) + \sum_{i \in \mathcal{T}} \sum_{j \in N_i} d_{i,j} \Delta(j \notin N'_i) \quad (4)$$

$$+ \sum_{m \neq n \in \mathcal{M}} \Delta(\exists i : I_i = m, n \in \mathbf{x}_i) + \text{MDL}(\mathbf{x}) \quad (5)$$

where as before $\mathbf{x}_i$ is the *set* of models that point i belongs to; $\Delta(\cdot)$ is the indicator function, taking value 1 if the statement is true and 0 otherwise; and N'_i the modified neighbourhood of track i . This optimisation is subject to the constraints that every track i belongs to the interior of one model I_i , or more formally

$$\forall i, I_i = \alpha \rightarrow \forall j \in N'_i, \alpha \in \mathbf{x}_j \quad (6)$$

We now describe in detail each term of our cost function.

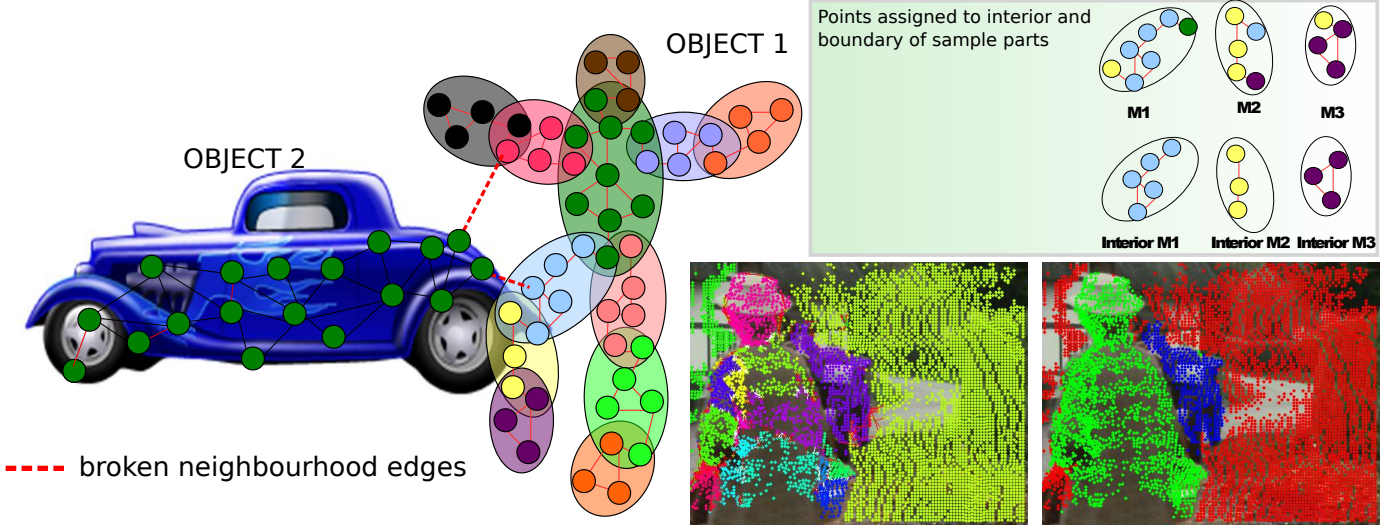


Fig. 2: **Left:** Conceptual illustration of our approach to 3D reconstruction of complex dynamic scenes. The image shows a person occluding a car. In the original neighbourhood graph, some point-tracks on the car are path connected with tracks on the person. Our approach reasons about object boundaries by adapting the neighbourhood, and breaking edges where necessary to detach parts from other occluding objects. **Top Right:** Illustration of the concept of overlapping models and *interior points* [31]. A tracked point belongs to the interior of a model (points with the same colour) if all its neighbours also belong to that model (though not necessarily as interior points). **Bottom right:** Real world example of segmentation into parts (left) and two objects and background (right).

4.1 Unary Costs (E_{data})

Our unary term is the sum of two costs i.e. $U_i(m) = G_i(m) + P_i(m)$, that encourage tracks that both move consistently as a rigid object and have similar saliency scores, to belong to the same model. The geometric term $G_i(m)$ evaluates the cost of assigning track i to a rigid model m as the deviation from the epipolar geometry across all pairs of consecutive frames. The second term $P_i(m)$ computes a saliency score for each pixel in every frame and encourages tracks with similar saliency scores, to belong to the same model.

4.1.1 Rigidity term G_i

Given a set of point tracks assigned to the same rigid part, we parameterise the rigid model m associated with them as a set of $F-1$ fundamental matrices $\mathbf{F}_m = \{\mathbf{F}_m^{1,2}, \dots, \mathbf{F}_m^{f,f+1}, \dots, \mathbf{F}_m^{F-1,F}\}$ for every pair of consecutive frames in the sequence $f = \{1, \dots, F-1\}$. The cost of associating track i to a specific rigid model m is the Sampson error [17] added over all pairs of fundamental matrices

$$G_i(m) = \sum_{f < F} \gamma^{-1} (u_i^{f+1T} \mathbf{F}_m^{f,f+1} u_i^f)^2 \quad (7)$$

where u_i^f encodes the homogeneous image coordinates of track i in frame f and u_i^{f+1} its corresponding position in frame $f+1$ and γ is the Sampson weight [17]. This cost is summed over all frames in which the track is visible. To estimate the fundamental matrices, we use the eight-point algorithm embedded in a RANSAC scheme followed by non-linear refinement of (7). This fitting cost has several clear advantages over the affine factorisation cost used by [11]. First, it allows to model perspective effects which are often present in unconstrained videos and to perform perspective reconstruction given an estimate of the camera calibration matrix. Second, it behaves better in the presence of missing data or short tracks, as it computes frame-to-frame geometric costs only for

the frames where the track is visible rather than the multiframe factorisation cost of [11].

4.1.2 Saliency Term

The work [33] provides a fully unsupervised method for object detection in an image I , using a novel saliency map S_I . While [33] made use of both the statistics taken from a large corpus of unlabelled images, and from the image itself, we only make use of the statistics of the single image (this measure is termed *within image saliency* in [33]). We compute saliency maps S_{I_f} for each frame f in the video sequence and define the saliency cost $P_i(m)$ of point i belonging to model m as the distance from the mean saliency of model m

$$P_i(m) = \lambda_s \sum_{f \leq F} (S_{I_f}(i) - \bar{S}_m)^2 \quad (8)$$

where $\bar{S}_m$ is the mean saliency of all tracks that currently belong to model m , $S_{I_f}(i)$ is the saliency score of point i in frame f and λ_s a weight on the importance of this term.

The second row of figure 3 shows the saliency detection results on our synthetic dancer sequence, where brighter means more salient. It can be seen that the saliency maps are very discriminative between the lower part of the person and the background. By incorporating saliency term 8 into our unary cost, we are able to successfully separate the moving person and the static background, which is challenging for just motion cues as the left foot of the person is on the ground plane and does not leave the ground during the whole sequence. As shown in figure 3, without saliency cues the algorithm is not able to segment the dancer from the background.

4.2 Adaptive Neighbourhood (E_{edge_break})

The cost (1) proposed in [31] was internally represented as a local MDL prior defined over the set of interior labels present in a local

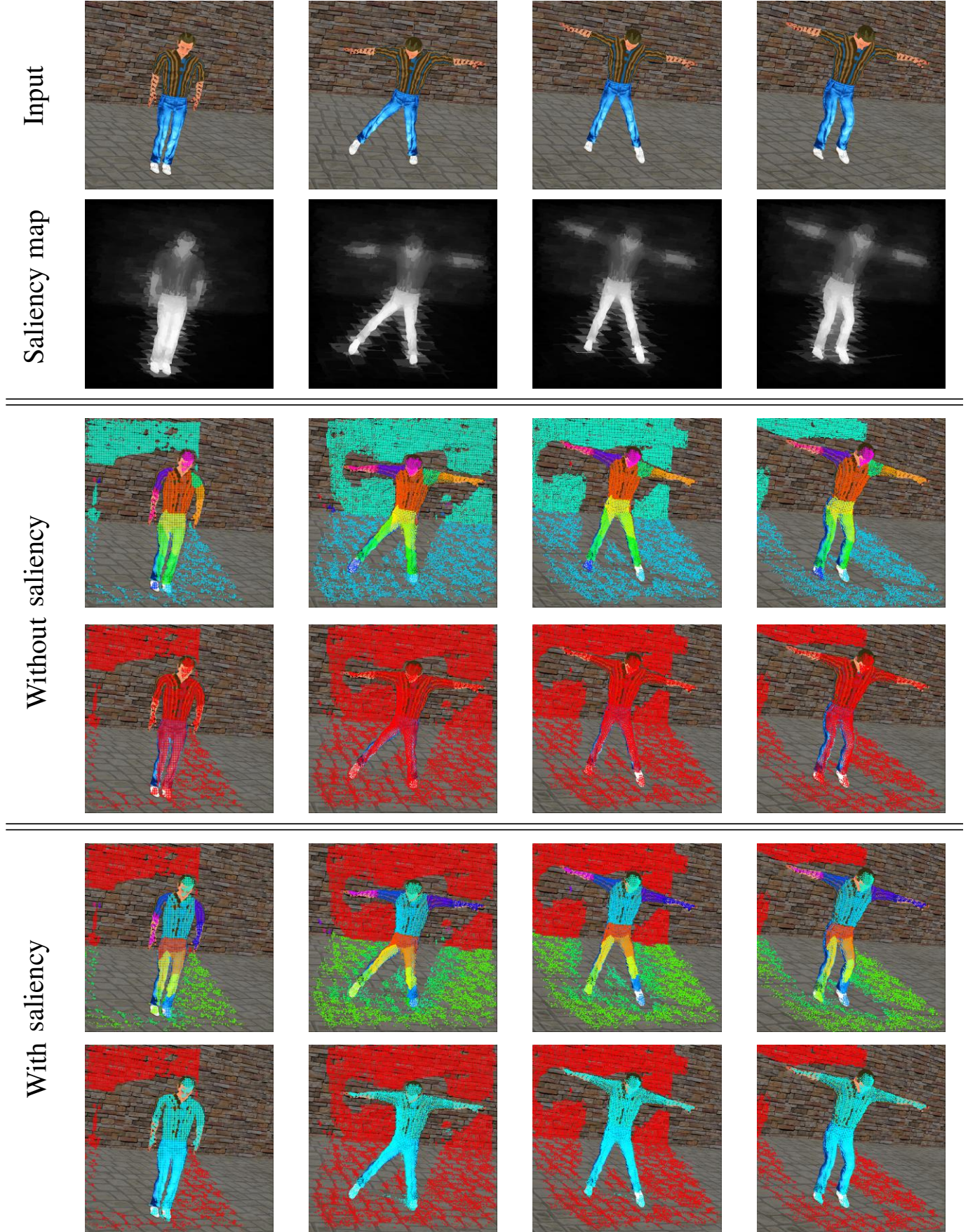


Fig. 3: Comparison of part and object segmentation results on dancer sequence with and without using saliency. The first two rows are the input sequence and saliency results respectively. The third and fourth rows show the part and object segmentation results without using saliency, while the last two rows show corresponding results with saliency. It can be seen that as saliency cues are discriminative between the person and the background, adding a saliency term into unary cost can help break the connections between the left foot and the ground floor, resulting in a correct object segmentation.

neighbourhood, and took the cost

$$\sum_{i \in \mathcal{T}} \sum_{m: \exists j \in N_i \cap m = I_j} U_i(m) \quad (9)$$

As discussed, in order to separate connected objects from one another, we wish to discard edges from the neighbourhood N_i with a per edge cost $d_{i,j}$. As such, the new cost will be of the form

$$\sum_{i \in \mathcal{T}} \sum_{m: \exists j \in N_i \cap m = I_j} \min \left(\sum_{j: I_j = m} d_{i,j}, U_i(m) \right) \quad (10)$$

where $d_{i,j}$ is a varying cost modulated by a sigmoid function of the form

$$d_{i,j} = \frac{M_c}{1 + e^{\frac{M_{ij} - M_a}{M_b}}} \quad (11)$$

Here M_{ij} is computed as the pairwise neighbourhood distance between points i and j in the image and velocity spaces. M_a , M_b and M_c are parameters of the sigmoid function.

4.3 Overlap Sparsity Term (E_{sparse})

By itself, discarding edges from the neighbourhood graph improves the quality of the parts found, and allows more objects to be found. However, it does not correctly separate objects from the background. In almost all sequences, we find that one or two ambiguous tracks exist that could be easily explained as either object or background parts. These ambiguous tracks act as junctions, or regions of overlap between foreground and background objects, connecting the two and making it impossible to distinguish between foreground and background.

To eliminate this leaking, we introduce a novel sparsity term that penalises the total number of models that overlap and encourages regions with limited overlap to disconnect. We formulate this penalty as a count of the number of pairs of models (m, n) such that there exists a track belonging to the interior of model m and also to model n , i.e.

$$\sum_{m \neq n \in \mathcal{M}} \Delta(\exists i : I_i = m, n \in \mathbf{x}_i) \quad (12)$$

As this cost does not depend on the number of tracks in the region of overlap, it dominates in small regions of overlap or where the cost of discarding edges is small, and is ignored elsewhere.

5 EFFICIENT OPTIMISATION

As with other multiple model fitting approaches [11], [18], [31], we initialise with an excess of models which are generated by sampling randomly neighbouring groups of ten feature tracks and computing the frame-to-frame fundamental matrices using the eight-point algorithm [17]. We then optimise the cost (3) using a hill-climbing approach alternating between: (i) fixing the parameters $\mathbf{F}_m$ and optimising the labelling that assigns tracks to a set of parts (models) $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ and (ii) fixing the labelling and optimising $\mathbf{F}_m$ for all models.

Alpha expansion [6] finds a local optimum of a difficult to optimise cost function by iteratively moving from a current labelling to the lowest-cost solution obtained by relabelling some of the variables as α . Finding an optimal move is formulated as a pseudo Boolean optimisation [4] and solved using graph-cuts [5]. We follow [31] in considering expansion moves over the interior of labels. We use $A \in 2^T$ to refer the found expansion move, with A_i taking value 1 if variable I_i transitions to label α in the move,

and 0 otherwise. Unlike [31] we will need to explicitly keep track of whether or not tracks belong to models at all (either as interior or boundary tracks) and for a particular expansion move on label α this will be done by means of binary variables $M_i^\alpha = 1$ if $\alpha \in \mathbf{x}_i$ and a complementary set of variables M_i^β , such that $\beta \neq \alpha$ and $M_i^\beta = 0$ if $\beta \in \mathbf{x}_i$.

Optimisation of the costs E_{data} and E_{mdl} can be done using the techniques of [31]. We now deal with the modifications to the optimisation required by the terms E_{break} and E_{sparse} . Although exact optimisation of either of these costs is straightforward, optimising both together is challenging, and we make use of the convex-concave procedure (CCP) [43], [24], and find an optimisable cost that is tight at the current location, but an overestimate elsewhere.

5.1 Exactly Optimising E_{break}

We can rewrite cost (10) in terms of the auxiliary variables

$$\begin{aligned} & \sum_{i \in \mathcal{T}} \sum_{\substack{\beta \in \mathcal{M} \\ \beta \neq \alpha}} \min \left(\sum_{\substack{j \in N_i \\ j: I_j = \beta}} ((1 - A_j) d_{i,j} M_i^\beta) + U_i(\beta)(1 - M_i^\beta) \right) \\ & + \sum_{i \in \mathcal{T}} \min_{M_i^\alpha} \left(\sum_{\substack{j \in N_i \\ I_j \neq \alpha}} (A_j d_{i,j} (1 - M_i^\alpha)) + U_i(\alpha) M_i^\alpha \right) \\ & + \sum_{\substack{j \in N_i \\ I_j = \alpha}} (d_{i,j} (1 - M_i^\alpha)) \end{aligned} \quad (13)$$

This change can be seen as a robustification of the local co-occurrence potentials of [31] analogous to the robust P^n model [20]. As with the P^n potentials, it can be formulated as a graph-cut problem simply by adjusting the used edge weights. Figure 4 shows an example when no track in the neighborhood takes label α . Please see the caption of the figure for a detailed explanation of the graph construction and differences between overlapping cost and edge breaking cost. For neighborhood containing α , the graph construction is similar. Note that we always assume that $d_{i,i} = \infty$, indicating that the interior model of a track is always a model for itself.

5.2 Approximately Minimising E_{sparse}

For the following section it is more convenient to use sets to describe which points belong to which models. We use $\mathbf{M}^\beta$ to refer to the set of points belonging to model β , $\mathbf{I}^\beta$ for the interior of model β , $\mathbf{M}_{last}^\beta$ for the region (fixed throughout the move) that was assigned to model β by the previous move, and $\mathbf{I}_{last}^\beta$ for points previously belonging to the interior of model β . Performing an expansion move on label α , we have three cases to consider:

- 1) The cost is a direct function of the interior labels $\mathbf{I}^\alpha$
- 2) The cost depends on tracks belonging to the boundary of α : $\mathbf{M}^\alpha \setminus \mathbf{I}^\alpha$
- 3) The cost is not a function of α and depends on: $\mathbf{I}^\beta \cap \mathbf{M}^\gamma$, where $\beta, \gamma \neq \alpha$.

For an expansion move on label α , $\mathbf{I}^\alpha$ is monotone increasing while $\mathbf{I}^\beta$ is monotone decreasing. If one of either the sparsity costs, or the edge breaking of the previous subsection was not used, the labelling of $\mathbf{M}^\alpha$ and $\mathbf{M}^\beta$ would also be guaranteed to be monotone increasing/decreasing, but together the situation is more

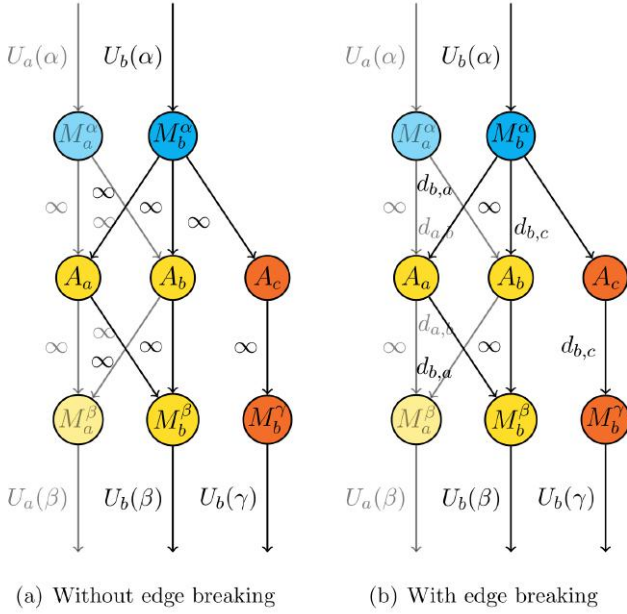


Fig. 4: Comparison of graph constructs between overlapping cost [31] and our new edge breaking cost 10. Left figure shows the graph construct for overlapping cost over the neighborhood of point b (a , b and c , taking label β , β and γ respectively). Transparent nodes indicate the contribution of overlapping neighborhoods. Right figure shows the modified graph construct for edge breaking cost, allowing point a to be dropped from N_b with cost $d_{b,a}$. The top row contains auxiliary variables indicating if tracks, a , b or c belong to model α . The middle row contains the standard expansion variables which govern whether or not a variable transitions to the interior of model α , while the bottom row shows auxiliary variables indicating if a variable belongs to model β or γ . Here we show the case when label α is not present in the neighborhood. To understand the differences between these two figures, we could have a look at the point a from neighbourhood N_b . In particular, there are two edge connections, $M_b^\alpha \rightarrow A_a$ and $A_a \rightarrow M_b^\beta$. In the left figure, these two edges cannot be broken (with connection weights ∞), indicating that α is not a model of point b ($M_b^\alpha = 0$) while α is an interior model of point a ($A_a = 1$) cannot happen, and similarly β is not a model of point b ($M_b^\beta = 1$) while β is an interior model of point a ($A_a = 0$) cannot happen as well, both due to the overlapping constraint. In the right, we allow both these two cases to happen, each with a penalty $d_{b,a}$.

complex. In the following discussion, we artificially constrain the set of possible moves of $\mathbf{M}^\beta$ to be monotone decreasing, and allow $\mathbf{M}^\alpha$ to change arbitrarily. Let us deal with these costs in turn:

Interior of α cost: We consider the localised MDL costs

$$\Delta(\mathbf{M}_{\text{last}}^\beta \cap \mathbf{I}^\alpha \neq \emptyset) + \Delta(\mathbf{I}^\beta \neq \emptyset) - 1. \quad (14)$$

This cost is 1 if $\mathbf{I}^\alpha$ expands into $\mathbf{M}_{\text{last}}^\beta$ without completely removing model β (which can only be done by making sure no tracks belong to the interior of model β) and 0 otherwise. Clearly this is an over-estimate as the true $\mathbf{M}^\beta$ in the set of all moves considered is always smaller than $\mathbf{M}_{\text{last}}^\beta$, and tight at the current location. As this cost is simply two MDL costs defined over subregions of the graph, it can be optimised using the techniques

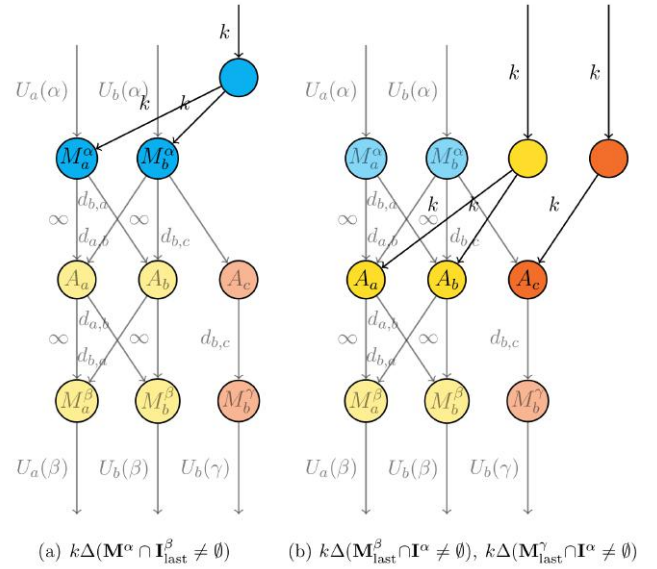


Fig. 5: Graph construct for overlap sparsity term. Transparent graph shows the graph construction result of edge breaking cost, as shown in Figure 4. Here we show the cost over the neighbourhood of point b (a , b and c , taking label β , β and γ respectively). We assume the edge between point b and c is currently broken, so $\mathbf{M}_{\text{last}}^\beta = \{a, b\}$ and $\mathbf{M}_{\text{last}}^\gamma = \{c\}$.

of [21]. As these move costs satisfy the CCP criteria, they reduces the original cost function.

Boundary of α cost: A similar argument can be made for the above cost. Instead of directly optimising it, we solve the over-approximation

$$\Delta(\mathbf{M}^\alpha \cap \mathbf{I}_{\text{last}}^\beta \neq \emptyset) + \Delta(\mathbf{I}^\beta \neq \emptyset) - 1. \quad (15)$$

This can be formulated as a local MDL prior over the auxiliary variable of the previous section and an MDL cost over label β .

Costs not dependent on α : The local co-occurrence potentials considered here, fall into the class of potentials that can not be exactly optimised by an expansion move over label α . Instead we follow the strategy of [21] and optimise the cost

$$0.5\Delta(\mathbf{I}^\gamma \cap \mathbf{M}_{\text{last}}^\beta \neq \emptyset) + 0.5\Delta(\mathbf{I}_{\text{last}}^\gamma \cap \mathbf{M}^\beta \neq \emptyset) \quad (16)$$

5.3 Merging Parts into Objects

The final result of our scene segmentation algorithm is the labeling $\mathbf{x}$ which assigns each feature track to a set of rigid parts. Figure 6 shows some results of the part segmentation (second row) for five videos of the Berkeley Motion Segmentation Dataset [7]. To segment the scene into objects we label connected components of overlapping parts as object detections.

6 3D RECONSTRUCTION

The optimisation of our cost function results in the labeling of rigid models or parts. Using the information about the regions of overlap, we also have a decomposition of the scene into different objects.

The 3D reconstruction of each object is then carried out using a piecewise rigid reconstruction approach. For each object we have a list of its constituent parts and a rigid model (set of fundamental matrices) for each part.

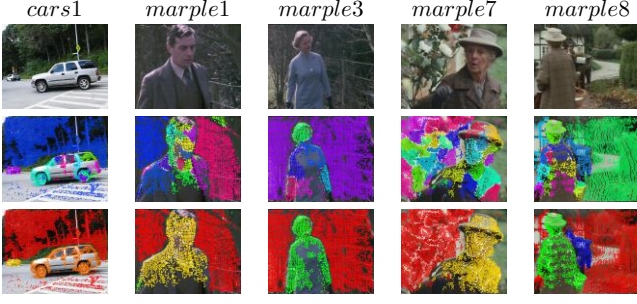


Fig. 6: Motion segmentation results on five sample sequences of the Berkeley Motion Segmentation Dataset [7]. **Second row:** Part segmentation. **Third row:** Object segmentation.



Fig. 7: Reconstruction results for a cat sequence of the Youtube-Objects Dataset [27]. **Second row** and **third row** are reconstruction results.

6.1 Piecewise Rigid Orthographic Reconstruction

When the average depth of an object is much larger than its depth relief, the 2D projections of the object can be well approximated with an orthographic camera model. In general, scenes captured with large focal length and small field of view tend to satisfy these conditions. In our case, the *bird*, *cat*, *bike* and *messi* sequences all belong to this category.

For these sequences, given the part and object segmentation results, we reconstruct each part independently with an orthographic factorization approach [23]. Then parts belonging to the same object are stitched together to obtain a consistent object reconstruction result.

In general, there are two ambiguities that can not be resolved when performing orthographic reconstruction for each patch, the depth and flip ambiguities. Here we assume temporal flip ambiguity has been resolved for each patch independently and we focus on solving the depth and flip ambiguity.

Figure 8 illustrates the two ambiguities for orthographic reconstruction. As patches are reconstructed separately, the ambiguities of overlapping patches do not necessarily agree with each other. To register these patches together to form a consistent object reconstruction, we need to solve the ambiguities between all overlapping patches.

Note that there is always a global depth and flipping ambiguity that could not be resolved, which is equivalent to pick one of the patches as reference and fix its depth and flip. Although depth ambiguity can be resolved trivially given the solution to the flip, the problem of solving the flip ambiguity is NP-hard. Fortunately, in most cases, as the number of patches within an object is relative

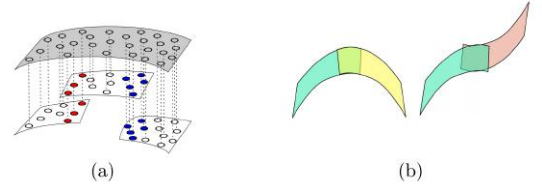


Fig. 8: Two ambiguities of orthographic reconstruction for each patch. (a) Reconstruction of shared points in different patches differ by a translation in the depth direction. (b) Representation of the ambiguity on the sign of the Z coordinate of the reconstructions. Figure from [10].

small we can afford a brute-force search over all possibilities.

In particular, we create a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ for overlapping patches where each node denotes a patch, and each edge denotes an overlapping between a pair of patches. We fix a patch as reference and precompute the cost for each edge. Assuming an edge e_{ij} between patch i and patch j , we compute two costs, c_{ij}^1 and c_{ij}^2 , associated with this edge:

$$\begin{aligned} c_{ij}^1 &= \|\mathbf{d}_i - \mathbf{d}_j - \overline{\mathbf{d}_i - \mathbf{d}_j}\|_2^2 \\ c_{ij}^2 &= \|\mathbf{d}_i + \mathbf{d}_j - \overline{\mathbf{d}_i + \mathbf{d}_j}\|_2^2 \end{aligned} \quad (17)$$

where $\mathbf{d}_i$ and $\mathbf{d}_j$ are the depth value matrices for all the common points over all frames of patch i and patch j respectively, $\overline{\mathbf{d}_i - \mathbf{d}_j}$ is the per frame average of the difference between these two depth matrices. Therefore c_{ij}^1 represents the alignment cost when two overlapping patches have the same flip. Similarly c_{ij}^2 denotes the cost when the two patches take different flips.

We then search over all possible flips and find the configuration $\hat{\mathbf{f}}$ with the lowest overall cost:

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\arg\min} \sum_{(i,j) \in \mathcal{E}} (c_{ij}^1 \cdot \Delta(f_i = f_j) + c_{ij}^2 \cdot \Delta(f_i \neq f_j)) \quad (18)$$

where the flip value f_i takes 1 if patch i is flipped, otherwise 0.

When there is enough overlap between neighboring patches and the points of the overlapping area have enough curvature to differentiate between flip and non-flip shapes, this depth alignment based stitching approach works robustly. However, when the overlapping region is not discriminative enough, this method fails. In practice, we also use another stitching strategy based on spatial rotation similarity, assuming that the rotations of neighboring patches on the same object are similar. The only difference between these two methods is in the way we compute edge costs. Instead of computing edge cost using depth alignment error, we compute the rotation difference cost, between two neighbouring rotations for both values of f_i .

After solving the flips, we then use a greedy strategy to compute the depth of each patch and align all the overlapping pieces together. Specifically, we greedily select the pair with most common points and merge these two patches with the larger patch fixed until we have only one patch left.

6.2 Piecewise Rigid Perspective Reconstruction

When the depth range of the scene is not negligible compared to the average depth (for instance, sequences from KITTI dataset [16]), we use a perspective camera model and perform piecewise perspective reconstruction. Similar to orthographic case,

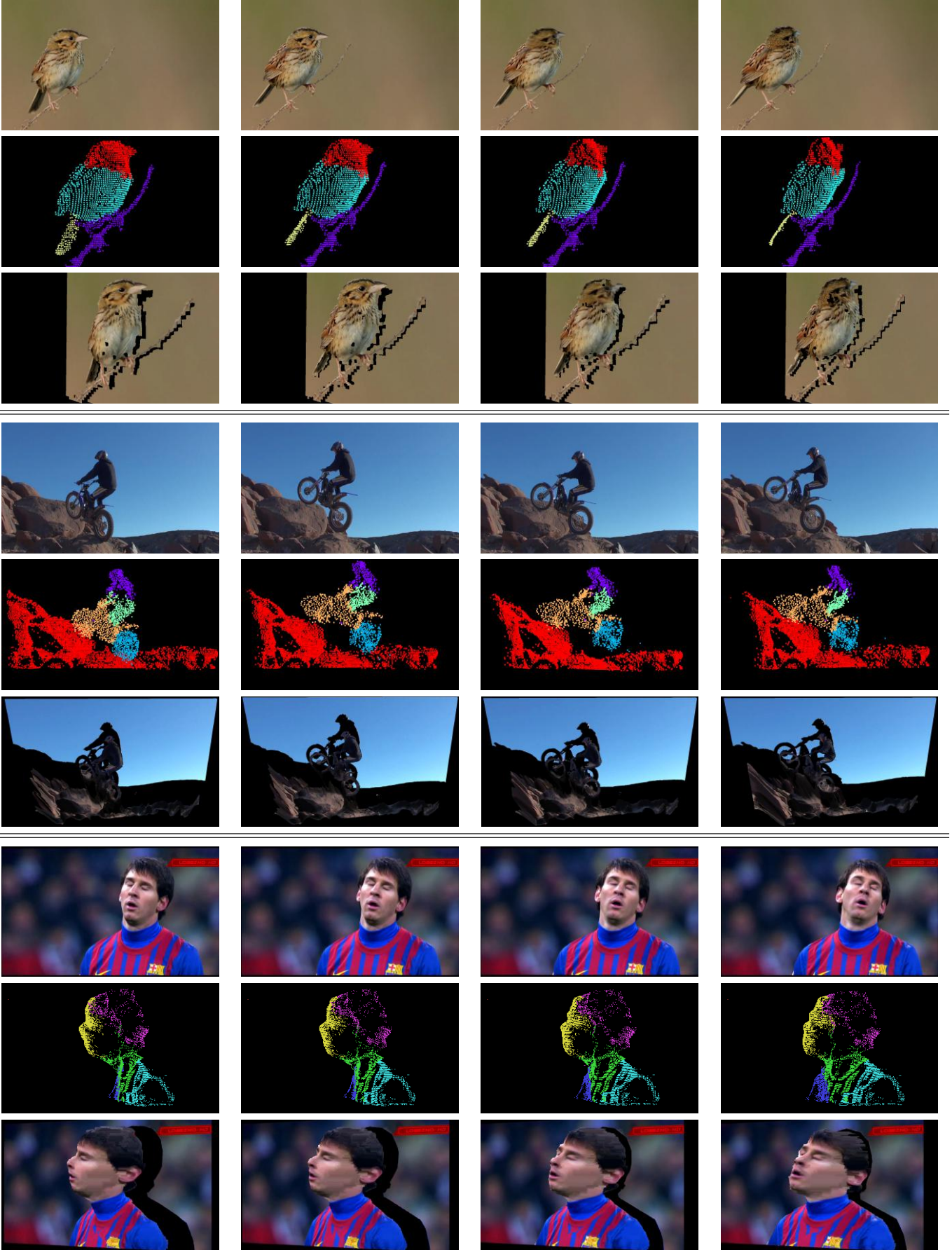


Fig. 9: **Top:** and **Middle:** Reconstruction results for a bird sequence and motorbike sequence *Youtube-Objects* Dataset [27]. **Bottom:** Reconstruction results of a football footage. For all three sequences, we show registered sparse reconstruction of rigid parts and the dense results after densification.

we do rigid perspective reconstruction on each individual part and then stitch connected parts into different objects. For our piecewise rigid reconstruction, we assume the calibration is known, and the only existing ambiguity between parts is a depth/scale ambiguity, which can then be resolved by enforcing that tracks belonging to two or more parts should be reconstructed at the same depth by each part model.

We use an incremental method to reconstruct the 3D shape of each rigid part. First we initialise from the two frames using five-point algorithm [34], and then keep adding new frame into the process, triangulating new points and doing incremental bundle adjustment until all the frames are reconstructed.

6.3 Depth-map Densification

Our reconstruction algorithm is based on sparse feature tracks. To densify the 3D reconstruction, we apply Gaussian filtering on the sparse 3D tracks in xy-RGB image space using the fast implementation of [2] that performs filtering using the permutohedral lattice. Regions of the video far from any tracks in the xy-RGB space are assigned to a flat background billboard.

7 EXPERIMENTAL RESULTS

Since we recover both a segmentation of the scene into multiple moving objects and a 3D model for each object, we evaluate both of these steps independently.

7.1 Evaluation of the Motion Segmentation Step

We evaluate the results of our *object-level* segmentation on the Berkeley Motion Segmentation Dataset using the tracks and evaluation tool proposed in [7]. Table 1 shows a comparison between the scores of our approach and the results from Brox and Malik’s motion segmentation algorithm [7]. The results show that our method exhibits comparable performance to [7]. While our *over-segmentation* error is higher than [7], the *overall error* and *average error* are very close, and in some cases lower. Although our algorithm can be used for motion segmentation exclusively, it is geared towards 3D reconstruction of complex dynamic scenes. Providing object boundaries are respected, our 3D reconstruction method is unharmed by a slight over-segmentation given that we perform piecewise reconstruction. The same set of parameters was used for all the experiments. The results of Fayad *et al.* [11] show how our algorithm would perform without the novel edge breaking and sparsity terms. Objects are never discovered in sequences longer than 10 frames, and in the majority of the 10 frame long sequences no objects are discovered.

7.2 Evaluation of Orthographic Reconstruction

We demonstrate our approach on videos from the *Youtube-Objects* Dataset [27]. These are unconstrained real-world videos downloaded from YouTube, with the purpose of object detection in video [27]. Figure 1 shows reconstructions of a *bird*, a *cat*, a *motorbike* and a *footballer*. We show the decomposition into *parts*, *objects* and a 3D model of the objects from a novel viewpoint for one frame. Figures 7 and 9 show 3D reconstructions for further frames of the four sequences. Our algorithm shows a good segmentation of the scenes and a convincing 3D reconstruction of these challenging videos.

TABLE 1: Evaluation results on the Berkeley Motion Segmentation Dataset using the metrics of [7]. Fayad *et al.* shows performance without discarding edges, using the same optimisation as in [11], [31].

	Density	overall error	average error	over-segmentation	extracted objects
First 10 frames(26 sequences)					
Brox Malik	3.34%	7.75%	25.01%	0.54	24
Fayad <i>et al.</i>	3.28%	15.23%	51.89%	0.23	7
Our method	3.28%	8.00%	25.46%	1.00	22
First 50 frames(15 sequences)					
Brox Malik	3.27%	7.13%	34.76%	0.53	9
Fayad <i>et al.</i>	3.25%	24.95%	63.67%	0.20	0
Our method	3.25%	5.93%	27.84%	3.70	13
First 200 frames(7 sequences)					
Brox Malik	3.43%	7.64%	31.14%	3.14	7
Fayad <i>et al.</i>	3.42%	28.81%	66.78%	0.29	0
Our method	3.42%	13.28%	39.86%	8.60	4

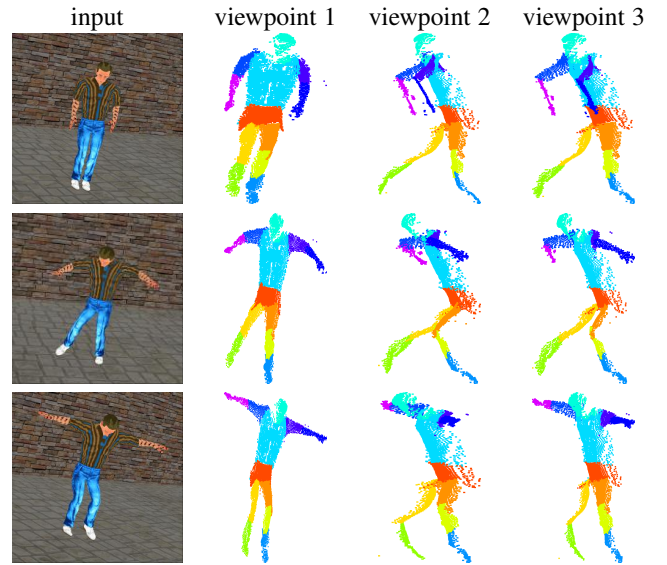


Fig. 10: Failure orthographic reconstruction results on dancer sequence. From left to right, we show input image and reconstruction results from three different viewpoints. Results are generated by performing orthographic reconstruction on the segmentation results with saliency cues, as shown in Figure 3. This sequence is very challenging as the rigid parts are small, and our method fails to resolve the flip ambiguities of small and nearly planar parts, such as the arms and legs.

However, our approach also has some limitations. As our method is purely geometric, to achieve an accurate 3D reconstruction, there needs to be enough 3D information to be exploited in the sequence. In other words, for each object, the parts should have non-planar 3D shapes, not be degenerate and have enough out of plane rotations within the sequence. Figure 10 shows a failure example of our algorithm. In this dancer sequence, we perform reconstruction on the segmentation results with saliency cues, as shown in Figure 3. This sequence is very challenging as the rigid parts are small and some of them are nearly planar and degenerate. Our method has difficulties resolving both the temporal flip ambiguity (the flip of a single part across different frames) and spatial flip ambiguity (the flip between overlapping parts). For instance, as shown in the figure, the flip of both arms and legs are wrong, despite that the 2D projections on the input image (viewpoint 1 shows camera viewpoint) look good.

7.3 Evaluation of Perspective Reconstruction

We evaluate our perspective reconstruction method on sequences from the KITTI dataset [16] both quantitatively and qualitatively. As sequences from KITTI mainly consist of static background and moving rigid objects, mostly cars, we turn off the overlapping between different parts since each rigid part readily corresponds to an individual object.

We show two different use cases of our approach, dense monocular depth estimation using two consecutive frames and monocular 3D sparse point cloud reconstruction using multiple frames.

7.3.1 Monocular Depth Map Estimation using Two Frames

In this experiment, we reconstruct a dense depth map from two consecutive frames taken from a monocular sequence of KITTI. First we obtain a sparse reconstruction of the background (we assume that the object with most points is the background) and of each moving object separately (by triangulating segmented sparse point tracks). Scale ambiguities between individual reconstructions are then resolved based on the assumption that all dynamic objects are always located on the background and occlude the static environment. We fix the scale of the background, and reconstruct the entire scene up to this fixed scale. For evaluation purposes, the unknown global scale will be estimated by registering the estimated depth values to the ground truth depth map. The final dense depth map is computed by linearly interpolating the sparse depth values of point tracks. In Figure 12, we show the obtained dense depth map after linear interpolation.

In order to compute the relative scales between foreground moving objects and the static environment, we first obtain a dense motion segmentation result based on superpixels and then enforce the constraint that the depth values of the boundary superpixels between dynamic objects and the static scene should be close. For dense segmentation, we simply divide the input image into superpixels [1], and assign all the pixels inside each superpixel segment the label with most support from the point tracks within the superpixel. Figure 11 shows an example of our sparse and dense segmentation results.

To compare with other state-of-the-art approaches, we implement the reconstruction algorithm recently proposed by Ranftl *et al.* [28]. In their paper, they focus on estimating dense depth maps of dynamic scenes from two consecutive monocular frames. Similarly to our approach, theirs also consists of two steps, motion segmentation and 3D reconstruction. For a fair comparison between both reconstruction methods, we use our dense segmentation result as input for both 3D reconstruction approaches. In their 3D reconstruction step, they first estimate an initial reconstruction result by triangulation and then they perform a global optimization to obtain the final result. In fact, our method can be thought of as an initialization for their global optimization step. Figure 12 shows a qualitative comparison between our method and the approach from Ranftl *et al.*. For quantitative evaluation, we compute the mean relative error (MRE) with regard to the sparse laser scan ground truth, as used in Ranftl *et al.* [28]. As shown in table 2, our approach has an error of 0.196, compared with 0.148 from Ranftl *et al.*.

7.3.2 3D Point Cloud Reconstruction using Multiple Frames

To further evaluate our approach, we show multi-frame piecewise rigid perspective reconstruction results on sequences from the

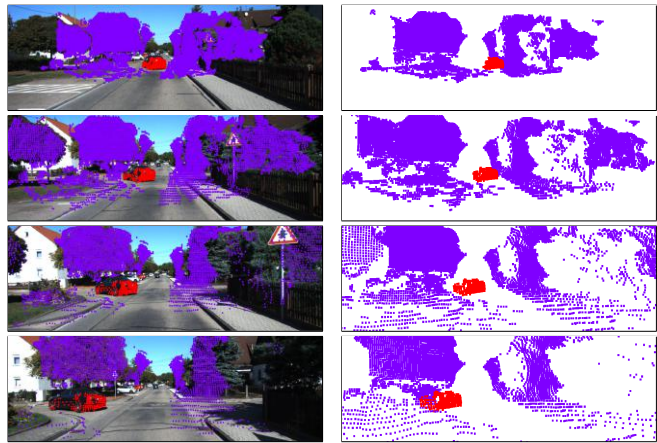


Fig. 13: Multi-frame dynamic scene reconstruction results on a 15 frame short sequence from KITTI dataset. From left to right, four input images overlaid with segmented sparse point tracks, and corresponding reconstruction results.

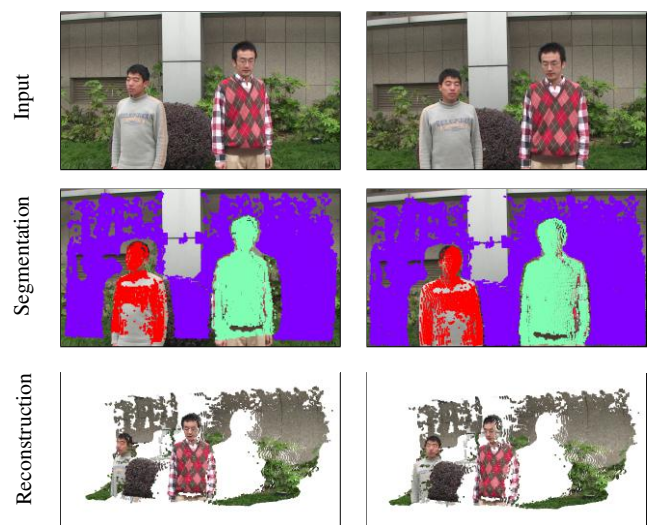


Fig. 14: Segmentation and reconstruction results on two-men sequence from Zhang *et al.* [45]. We successfully segment the scene into three rigid components, the static background and two moving person. Last row shows the corresponding reconstruction results.

	Ours	Ranftl <i>et al.</i>
Mean relative error (MRE)	0.196	0.148

TABLE 2: Quantitative comparison between our method and Ranftl *et al.* [28] on the KITTI odometry set [16]. Our result is the mean error over 427 randomly sampled frames out of 43552.

KITTI dataset [16] and from Zhang *et al.*'s [45] dataset. In this experiment, we first use an incremental method to reconstruct each rigid part separately and then assemble them together to achieve dynamic scene reconstruction results. We initialize from two frames with wide baseline and then perform incremental reconstruction. Specifically, in our implementation, we use the open source structure from motion library OpenSfM [15] for rigid perspective reconstruction. As each object is reconstructed

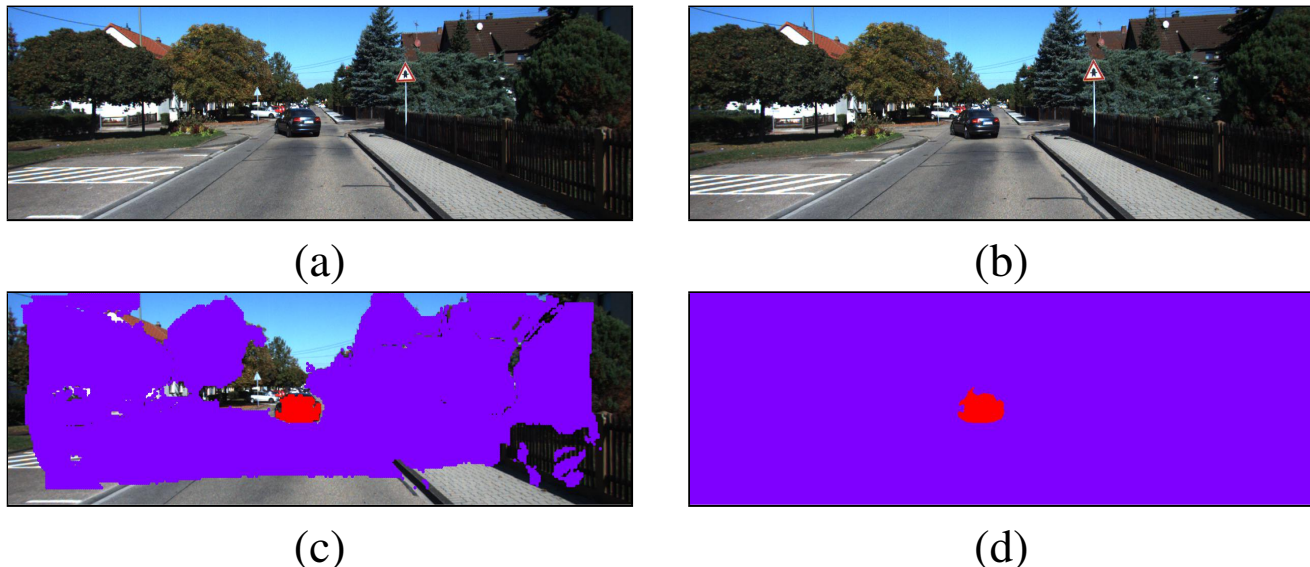


Fig. 11: Motion segmentation results on two consecutive images from the KITTI dataset. (a)-(b) two consecutive images. (c) sparse motion segmentation result on point tracks. (d) dense segmentation result.

independently, we need to solve the scale ambiguity to put everything together to create a dynamic scene reconstruction. To make it easier to register everything together, we create a graphical user interface where one could adjust the scale of each object manually. Examples shown in figure 13 and figure 14 are our scene reconstruction results after manual scaling.

Figure 13 shows our 3D reconstruction results for a short sequence with 15 consecutive frames, taken from KITTI. As shown in the figure, our method successfully segments the scene into the moving car and the static environment and reconstructs them accordingly. Figure 14 shows our reconstruction on the two-men sequence from Zhang *et al.* [45]. In this 30 frame long sequence, the two men move rigidly towards each other. From our result, the two people and the static background are correctly segmented and reconstructed.

8 CONCLUSION

In this work we propose a fully unsupervised approach to the challenging problem of simultaneously segmenting a dynamic scene into its constituent objects and reconstructing a 3D model of the scene. We focus on the reconstruction of real-world videos downloaded from the web or acquired with a single camera observing a complex dynamic scene containing an unknown mixture of multiple moving and possibly deforming objects. Our method consists of two steps, motion segmentation and 3D reconstruction. The motion segmentation step segments the dynamic scene into rigid parts and combines overlapping parts into objects. In the subsequent 3D reconstruction step, we reconstruct each rigid part separately and stitch these part reconstruction results into a consistent whole scene reconstruction. Our results show examples of successful orthographic reconstruction on videos from the *Youtube Objects* dataset and perspective reconstruction on KITTI sequences, *etc.*

REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE*

transactions on pattern analysis and machine intelligence, 34(11):2274–2282, 2012. 11

[2] A. Adams, J. Baek, and A. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Eurographics*, 2010. 10

[3] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR*, 2010. 2

[4] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, page 155–225, 2002. 6

[5] 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. 6

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. 23, 2001. 6

[7] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 2, 7, 8, 10

[8] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *ICCV*, pages 1071–1076, 1995. 1, 2

[9] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009. 2

[10] J. Fayad, L. Agapito, and A. Del Bue. Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *European Conference on Computer Vision ECCV 2010, Crete, Greece*, volume 6314 of *Lecture Notes in Computer Science*, pages 297–310. Springer Berlin / Heidelberg, September 2010. 8

[11] J. Fayad, C. Russell, and L. Agapito. Automated articulated structure and 3d shape recovery from point correspondences. In *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011. 1, 2, 3, 4, 6, 10

[12] A. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3-d reconstruction of independently moving objects. In *European Conference on Computer Vision (ECCV)*, Dublin, Ireland, 2000. 1

[13] F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *Asian Conference on Computer Vision (ACCV)*, November 2012. 2

[14] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *CVPR*, 2013. 1

[15] P. Gargallo, Y. Kuang, et al. Opensfm. <https://github.com/mapillary/OpenSfM>. 11

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013. 8, 11

[17] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 4, 6

[18] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision (IJCV)*, 97(2), 2012. 2, 6

[19] K. Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, volume 2, pages 301–306, Vancouver, Canada, July 2001. 2

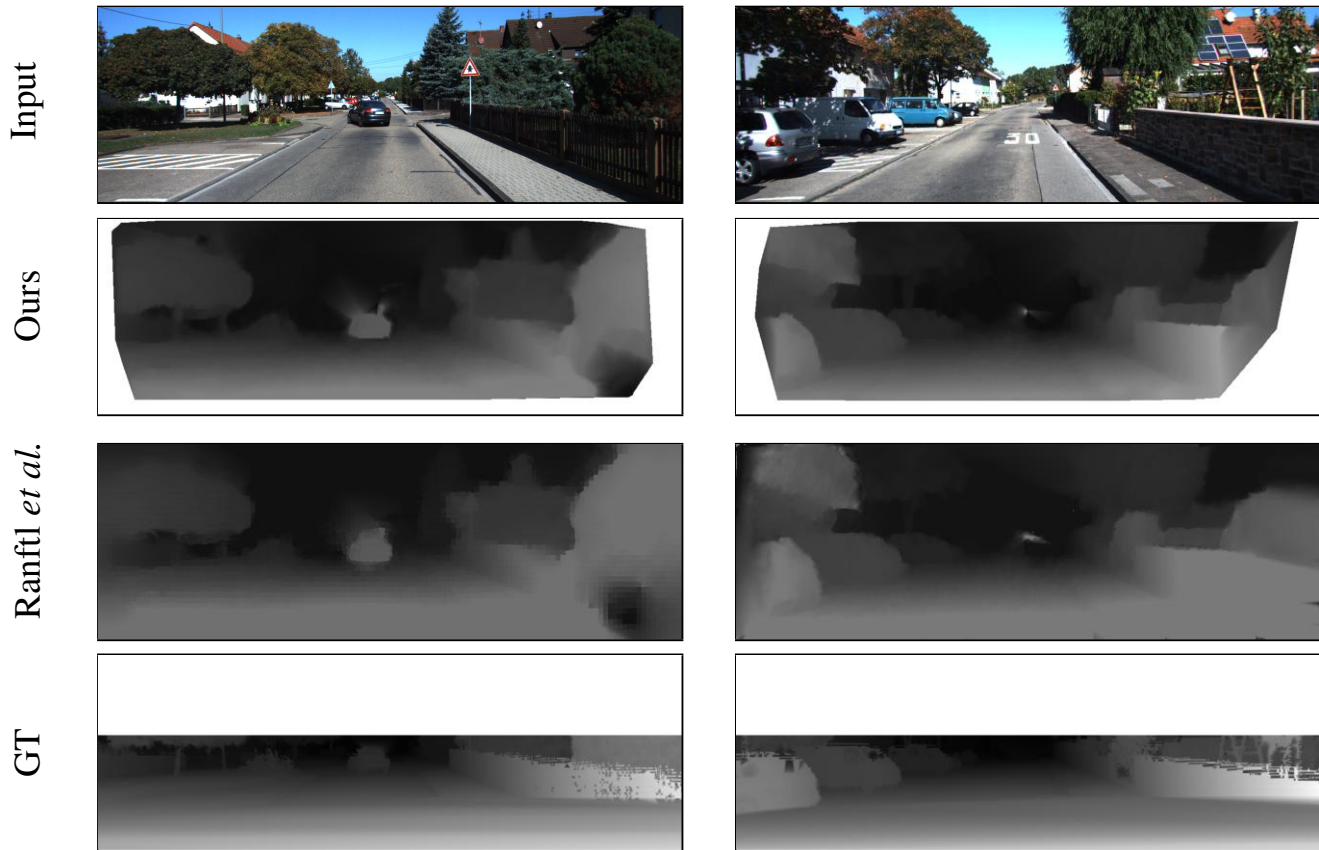


Fig. 12: Depth map reconstruction results on two frames from KITTI dataset. From top to bottom, we show input images, our dense reconstruction results (linear interpolation results from sparse reconstruction), our implementation of the reconstruction method from Ranftl *et al.* [28], and the ground truth (linear interpolation from sparse values).

- [20] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008. 6
- [21] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph-cut based inference with co-occurrence statistics. In *ECCV*, 2010. 7
- [22] Z. Li, J. Guo, L.-F. Cheong, and Z. Zhou. Perspective motion segmentation via collaborative clustering. In *ICCV*, 2013. 2
- [23] M. Marques and J. P. Costeira. Estimating 3D shape from degenerate sequences with missing data. *CVIU*, 2008. 8
- [24] M. Narasimhan and J. A. Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. *arXiv preprint arXiv:1207.1404*, 2012. 6
- [25] K. Ozden, K. Schindler, and L. van Gool. Multibody structure-from-motion in practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2010. 1, 2
- [26] M. Paladini, A. Del Bue, J. Xavier, L. Agapito, M. Stosic, and M. Dodig. Optimal metric projections for deformable and articulated structure-from-motion. *IJCV*, 2012. 2, 3
- [27] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 2, 8, 9, 10
- [28] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 11, 13
- [29] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(10):1832–1845, 2010. 2
- [30] A. Roussos, C. Russell, R. Garg, and L. Agapito. Dense multibody motion estimation and reconstruction from a handheld camera. In *ISMAR*, 2012. 1, 2
- [31] C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *CVPR*, 2011. 1, 2, 3, 4, 6, 7, 10
- [32] K. Schindler, D. Suter, and H. Wang. A model selection framework for multibody structure-and-motion of image sequences. *International Journal of Computer Vision (IJCV)*, 79(2):159–177, 2008. 1
- [33] P. Siva, C. Russell, T. Xiang, and L. Agapito. Looking beyond the image: Unsupervised learning for object saliency and detection. In *CVPR*, 2013. 3, 4
- [34] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294, June 2006. 10
- [35] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 3
- [36] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method - part 3 detection and tracking of point features. Technical Report CMU-CS-91-132, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, April 1991. 2
- [37] L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 2008. 2, 3
- [38] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *CVPR*, volume 2, pages 1110–1115, June 2005. 2
- [39] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *ICCV*, 2009. 1, 2
- [40] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). In *CVPR*, pages 621–628, 2003. 2
- [41] C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, 2012. 2
- [42] J. Yan and M. Pollefeys. A factorization-based approach for articulated non-rigid shape, motion and kinematic chain recovery from video. *PAMI*, 30(5), May 2008. 2
- [43] A. L. Yuille and A. Rangarajan. The concave-convex procedure (cccp). *NIPS*, 2002. 6
- [44] L. Zelnik-Manor and M. Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. In *CVPR*, volume 2, pages 287–293, June 2003. 2
- [45] G. Zhang, J. Jia, and H. Bao. Simultaneous multi-body stereo and segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011. 11, 12