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

Chris Russell†, Rui Yu†, and Lourdes Agapito

University College London
http://www0.cs.ucl.ac.uk/staff/lagapito/research/youtube3d

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* dataset.

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.

Multibody SfM and non-rigid structure from motion (NRSfM) have addressed some of the limitations of SfM and have seen sustained progress in dealing with dynamic scenes [21,25] or creating vivid life-like reconstructions of deformable

† The first two authors assert equal contribution and joint first authorship.

This research was funded by the European Research Council under the ERC Starting Grant agreement 204871-HUMANIS.

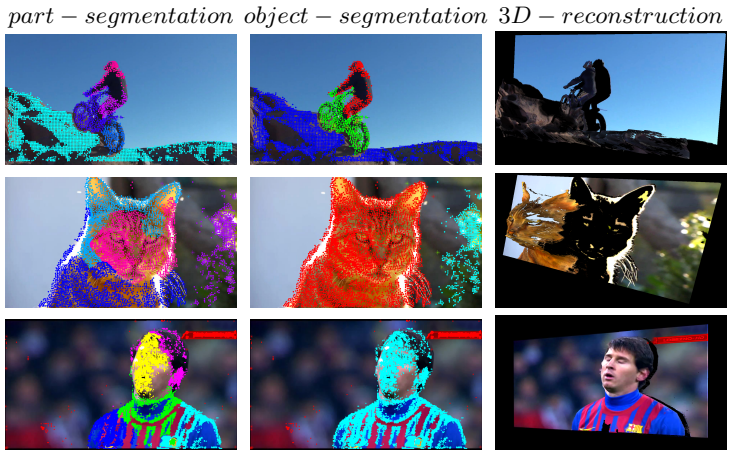


Fig. 1. Segmentation and 3D reconstruction results of two dynamic sequences of the *Youtube-Objects* Dataset [23] and a football sequence downloaded from YouTube. **Left:** segmentation into *parts* (rigid models). **Centre:** segmentation into *objects*. **Right:** densified 3D video pop-up from a novel viewpoint. The *motorbike* sequence, acquired with a moving camera, shows articulated motion. The *cat* sequence is a non-rigid sequence occluding a static background. Bottom row shows a reconstruction of football footage. For videos see the project website http://www0.cs.ucl.ac.uk/staff/lagapito/research/youtube3d.

objects [12]. However, they remain far behind their rigid counterparts. Multi-body SfM approaches can segment the scene into multiple rigidly moving objects, however they cannot deal simultaneously with the presence of deformable or articulated objects in the scene. Although NRSfM algorithms can reconstruct a single pre-segmented deformable surface moving in front of a camera [12,33], they require manual segmentation of the scene into background and foreground.

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 [9,26,33]. However, if 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.

Contributions: The main contribution of this paper is to offer a solution to the problem of *scene reconstruction* for real-world dynamic monocular videos that deals seamlessly with the presence of 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 [7,10,27,21] follow a pipeline approach where first feature point tracks or dense 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** [7] extended the classic affine factorisation algorithm for static scenes [30] to the case of multiple independently moving rigid objects. While the original approach [7] was unable to deal with dependencies in the motions it was later extended to deal with degenerate [38] and articulated motions [32,36]. More recent approaches to multibody SfM such as Ozden *et al.* [21] are able to perform simultaneous tracking, segmentation and reconstruction of a few feature points on realistic sequences. Roussos *et al.* [25] 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 [34] and methods that can deal with noise and outliers [24]. Motion segmentation has also been cast as a motion subspace clustering problem, first applied to the affine camera case [8,15] and later extended to the case of perspective scenes [18]. Approaches such as Brox and Malik's [6] exploit the consistency of point trajectories over time and can deal with non-rigid motion. On the other hand, superpixel [11] and supervoxel [35] 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 [31,22] to 2D tracks. Piecewise reconstruction has also been successfully applied to NRSfM [26,33] and articulated reconstruction [9] 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 [14,26] 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 labelling problem and continuous optimisation to update the model parameters. This approach has previously been applied to computer vision problems such as stereo [2]; motion segmentation [14]; 3D reconstruction of non-rigid [26] and articulated objects [9]; and multi-body reconstruction [25].

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 Simultaneous Segmentation and Reconstruction

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 $\mathcal{T} = [1, \ldots, T]$ of feature point tracks using Sundaram *et al.*'s publicly available code [29]. 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 Reconstruction with Overlapping Models

The works [26,9] 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 [31,22] 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 [9] or local quadratic deformations [26]. 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.

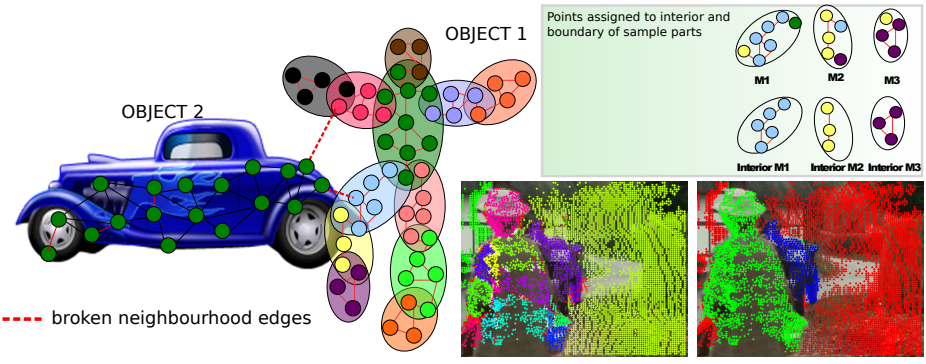


Fig. 2. **Left:** Conceptual illustration of our approach to 3D reconstruction of complex dynamic scenes. The image shows a person occluding a car. In the original neighbourhood graph, some point-tracks on the car are path connected with tracks on the person. Our approach reasons about object boundaries 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* [26]. 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).

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 $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_T\}$ 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 [9,26]

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

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 [9] 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

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

where $\mathbf{I} = \{I_1, I_2 \ldots I_T\}$ 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.* [26] showed how this problem could be formulated as a labelling 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 [26] and articulated [9] 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 [9,26] 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 [28] to encourage parts of similar saliency to belong to the same object.

4 Scene Reconstruction with an Adaptive Neighbourhood

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

$$C(\mathbf{x}) = E_{data} + E_{edge\_break} + E_{sparse} + E_{mdl} \tag{3}$$

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

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

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

$$\forall i, I_i = \alpha \rightarrow \forall j \in N_i', \alpha \in \mathbf{x}_j \tag{6}$$

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

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.

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 $\mathbf{F}_m = \{\mathbf{F}_m^{1,2}, \ldots, \mathbf{F}_m^{f,f+1}, \ldots, \mathbf{F}_m^{F-1,F}\}$ for every pair of consecutive frames in the sequence $f = \{1, \ldots, F-1\}$. The cost of associating track i to a specific rigid model m is the Sampson error [13] added over all pairs of fundamental matrices

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

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 [13]. 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 [9]. 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 [9].

Saliency Term The work [28] provides a fully unsupervised method for object detection in an image I, using a novel saliency map S_I. While [28] 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 [28]). 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

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

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.

4.2 Topologically Adaptive Neighbourhood (E_{edge_break})

The cost (1) proposed in [26] was internally represented as a local MDL prior defined over the set of interior labels present in a local neighbourhood, and took the cost

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

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 of $d_{i,j}$. As such, the new cost will be of the form

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

Here, the weights $d_{i,j}$ are found by passing the distance between points i and j in the image and velocity spaces through a 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.

$$\sum_{m \neq n \in \mathcal{M}} \Delta(\exists i : I_i = m, n \in \mathbf{x}_i) \tag{11}$$

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 [9,14,26], we initialise with an excess of models which are generated by sampling randomly groups of ten feature tracks and computing the frame-to-frame fundamental matrices using the eight-point algorithm [13]. 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) $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_T\}$ and *(ii)* fixing the labelling and optimising $\mathbf{F}_m$ for all models.

Alpha expansion [5] 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 [3] and solved using graph-cuts [4]. We follow [26] 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 [26] 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 [26]. 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 relatively straightforward, optimising both together is challenging, and we make use of the convex-concave procedure (CCP) [37,20], and find an optimisable cost that is tight at the current location, but an over-estimate elsewhere.

5.1 Exactly Optimising E_{break}

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

$$\sum_{i\in\mathcal{T}} \sum_{\substack{\beta\in\mathcal{M}\\ \beta\neq\alpha}} \sum_{j:I_j=\beta} \min_{M_i^\beta} \left(A_j d_{i,j}(1-M_i^\beta) + U_i(\beta)(1-M_i^\beta)\right) \tag{12}$$

$$+ \sum_{i\in\mathcal{T}} \sum_{j:I_j=\alpha} \min_{M_i^\alpha} \left((1-A_j)d_{i,j}M_i^\alpha + U_i(\alpha)M_i^\alpha\right) \tag{13}$$

This change can been seen as a robustification of the local co-occurrence potentials of [26] analogous to the robust Pn model [16]. As with the Pn potentials, it can be formulated as a graph-cut problem simply by adjusting the edge weights used as shown in Figure 3, left and centre left.

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, $\mathbf{M}^\beta_{\text{last}}$ for the region (fixed throughout the move) that was assigned to model β by the previous move, and $\mathbf{I}^\beta_{\text{last}}$ 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 in the boundary of points belonging 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 $\beta, \gamma \neq \alpha$.

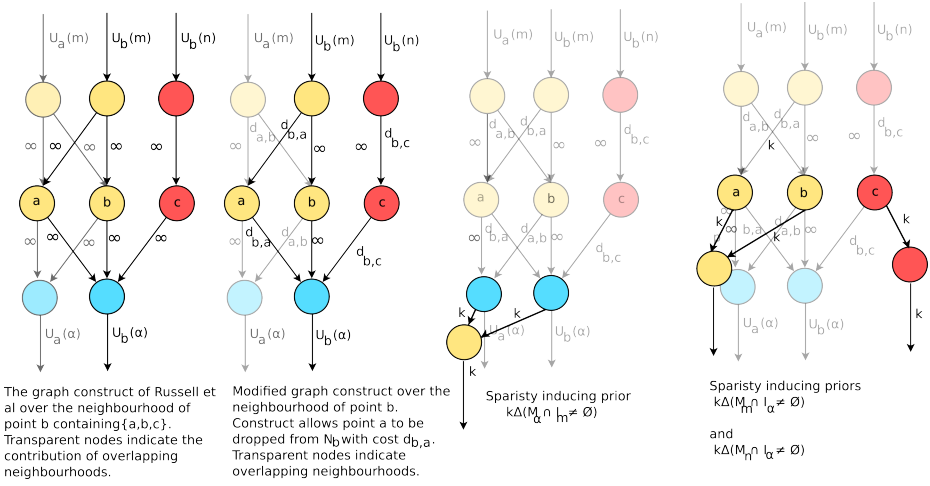


Fig. 3. **Graph constructs**. Of the main graph construct repeated in all subfigures, the top row contains all auxiliary variables indicating if tracks, a, b or c belong to model m or n. The middle row contains the standard expansion variables which govern whether or not a variable transitions to the interior of model α, while the bottom row indicates if a variable belongs to model α.

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 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 by turn:

Interior of α cost: We consider the localised MDL costs

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

This cost is 1 if $\mathbf{I}^\alpha$ expands into $\mathbf{M}^\beta_{\text{last}}$ 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 $\mathbf{M}^\beta_{\text{last}}$, 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 of [17]. 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

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

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

quality of the approximation of these terms. If we assume that all tracks belonging to the same model at every iteration of graph-cuts are path-connected[1], and if no edges are discarded[2] then the over-estimate is tight. There are 3 straightforward cases to consider:

1. No variables in $\mathbf{M}^{\beta}_{\text{last}}$ are in $\mathbf{I}^{\alpha}$. Here the cost is trivially correct.
2. No variable belongs to $\mathbf{M}^{\beta}$. Again, the cost is trivially correct.
3. At least one variable i belongs to $\mathbf{I}^{\alpha}$ and $\mathbf{M}^{\beta}_{\text{last}}$, and one variable j belongs to $\mathbf{I}^{\beta}$. As i and j are path-connected in $\mathbf{M}^{\beta}_{\text{last}}$, and for any possible move variables in $\mathbf{M}^{\beta}_{\text{last}}$ must either stay as variable β or move to label α, we have a chain of variables $\{i, k, l, \ldots j\}$ belonging to either $\mathbf{I}^{\alpha}$ or $\mathbf{I}^{\beta}$ such that $i \in N_{\text{k}}, k \in N_{\text{l}}, l \in \ldots N_{\text{j}}$. As $i \in \mathbf{I}^{\alpha}, j \in \mathbf{I}^{\beta}$, there must be at least one pair where $k \in N_{\text{l}}, k \in \mathbf{I}^{\alpha}$ and $l \in \mathbf{I}^{\beta}$ – and the cost is tight. □

Although the proof does not hold where edges are discarded it does provide intuition as to how alpha-expansion minimises the cost. In the first iterations, regions are swept out without breaking all edges, and finding solutions with excessive overlap between models. In subsequent iterations, the region belonging to the boundaries of models contracts cleanly separating parts.

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 [17] and optimise the cost

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

5.3 Merging Parts into Objects: Object-level Segmentation

The final result of our scene segmentation algorithm is the labelling $\mathbf{x}$ 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 [6]. 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 labelling of rigid models or parts. Using the information about the regions of overlap and the saliency scores, we also have a decomposition of the scene into different objects. In addition, our optimisation estimates model parameters for each rigid model m. Each rigid

[1] In practice this is almost always true, due to the regularisation caused by overlapping models or pairwise terms

[2] Again, most edges are not discarded. The majority of models that would overlap without edge discardation, continue overlapping in the solution found.



Fig. 4. Reconstruction results for a cat sequence of the *Youtube-Objects* Dataset [23].

Table 1. Evaluation results on the Berkeley Motion Segmentation Dataset using the metrics of [6]. *Fayad et al.* shows results without discarding edges, using the same optimisation as in [9,26].

	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 *et al.*	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 *et al.*	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 *et al.*	3.42%	28.81%	66.78%	0.29	0
Our method	3.42%	13.28%	39.86%	8.60	4

model is parameterised as the set of fundamental matrices $\mathbf{F}_m$ that describe the epipolar geometry between every pair of consecutive frames.

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 of part. First each part is reconstructed independently using the estimated fundamental matrices $\mathbf{F}_m$. If the calibration of the camera is known, each fundamental matrix can be decomposed into the relative rotation and translation between frames and an initial estimate of the shape is obtained using the DLT algorithm [13]. The shape and motion parameters are then refined using the sparse implementation of bundle adjustment [19]. If the camera calibration is unknown the shape is initialized using a factorization algorithm followed by per-frame motion estimation using the PnP algorithm [13]. A final refinement of the shape, motion and focal length parameters is then carried out via bundle adjustment.

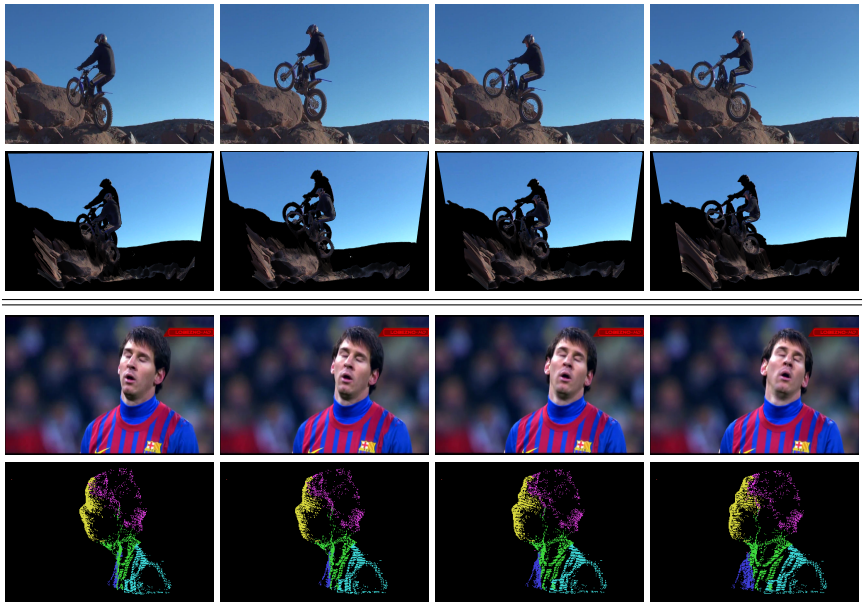


Fig. 5. **Top:** Reconstruction results for a motorbike sequence *Youtube-Objects* Dataset [23]. **Bottom:** Sparse reconstruction of football footage, showing both the assignment of tracks to parts and the quality of reconstruction before densification.

Aligning overlapping segments: Objects are segmented as a set of overlapping parts that require a final *stitching* step using the areas of overlap to enforce global consistency on the 3D surface. As we use a perspective camera model, the only existing ambiguity between parts is a depth/scale ambiguity which can be resolved by enforcing that tracks belonging to two or more parts should be reconstructed at the same depth by each part model.

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 [1] 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.

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 [6]. Table 1 shows a comparison between the scores of our approach and the results from Brox and Malik's motion

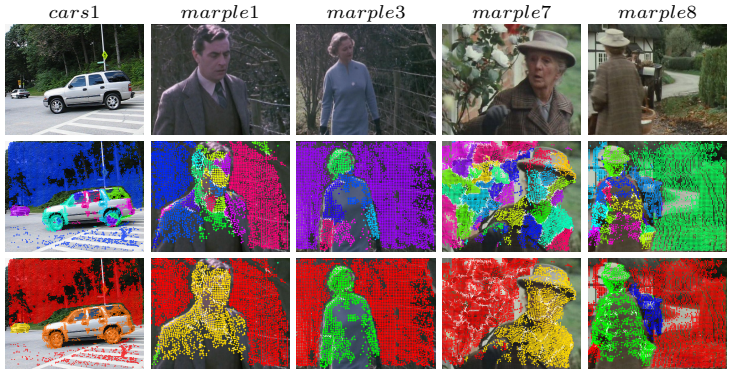


Fig. 6. Motion segmentation results on five sample sequences of the Berkeley Motion Segmentation Dataset [6]. **Second row:** Part segmentation. **Third row:** Object segmentation.

segmentation algorithm [6]. The results show that our method exhibits comparable performance to [6]. While our *over-segmentation* error is higher than [6], 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.* [9] 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.

Evaluation of the 3D Reconstruction: We demonstrate our approach on videos from the *Youtube-Objects* Dataset [23]. These are unconstrained real-world videos downloaded from YouTube, with the purpose of object detection in video [23]. Figure 1 shows reconstructions of 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 4 and 5 show 3D reconstructions for further frames of the three sequences. Our algorithm shows a good segmentation of the scenes and a convincing 3D reconstruction of these challenging videos.

8 Conclusion

In this paper 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 results show examples of segmentation and 3D reconstruction on videos from the *Youtube Objects* dataset.

References

1. A. Adams, J. Baek, and A. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Eurographics*, 2010. 13
2. M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR*, 2010. 4
3. E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, page 155 – 225, 2002. 9
4. 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. 9
5. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23, 2001. 9
6. T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 3, 11, 12, 13, 14
7. J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *ICCV*, pages 1071–1076, 1995. 3
8. E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009. 3
9. 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. 2, 3, 4, 5, 6, 7, 8, 12, 14
10. 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. 3
11. F. Galasso, R. Cipolla, and B. Schiele. Video segmentation with superpixels. In *Asian Conference on Computer Vision (ACCV)*, November 2012. 3
12. R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *CVPR*, 2013. 2
13. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 7, 8, 12
14. H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision (IJCV)*, 97(2), 2012. 4, 8
15. K. Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, volume 2, pages 301–306, Vancouver, Canada, July 2001. 3
16. P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008. 9
17. L. Ladickỳ, C. Russell, P. Kohli, and P. H. Torr. Inference methods for crfs with co-occurrence statistics. *International journal of computer vision*, 103(2):213–225, 2013. 10, 11
18. Z. Li, J. Guo, L.-F. Cheong, and Z. Zhou. Perspective motion segmentation via collaborative clustering. In *ICCV*, 2013. 3
19. M. A. Lourakis and A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 2009. 12
20. M. Narasimhan and J. A. Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. *arXiv preprint arXiv:1207.1404*, 2012. 9
21. 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, 3
22. M. Paladini, A. Del Bue, J. Xavier, L. Agapito, M. Stosic, and M. Dodig. Factorization for Non-Rigid and Articulated Structure using Metric Projections. *IJCV*, 2012. 3, 4

23. A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors fromweakly annotated video. In *CVPR*, 2012. 2, 12, 13, 14
24. 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. 3
25. A. Roussos, C. Russell, R. Garg, and L. Agapito. Dense multibody motion estimation and reconstruction from a handheld camera. In *ISMAR*, 2012. 1, 3, 4
26. C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *CVPR*, 2011. 2, 3, 4, 5, 6, 8, 9, 12
27. 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. 3
28. P. Siva, C. Russell, T. Xiang, and L. Agapito. Looking beyond the image: Unsupervised learning for object saliency and detection. In *CVPR*, 2013. 6, 7
29. N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 4
30. 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. 3
31. L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, pages 878–892, 2008. 3, 4
32. P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *CVPR*, volume 2, pages 1110–1115, June 2005. 3
33. A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *ICCV*, 2009. 2, 3
34. R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). In *CVPR*, pages 621–628, 2003. 3
35. C. Xu and J. J. Corso. Evaluation of super-voxel methods for early video processing. In *CVPR*, 2012. 3
36. 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. 3
37. A. L. Yuille and A. Rangarajan. The concave-convex procedure (cccp). *NIPS*, 2002. 9
38. 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. 3