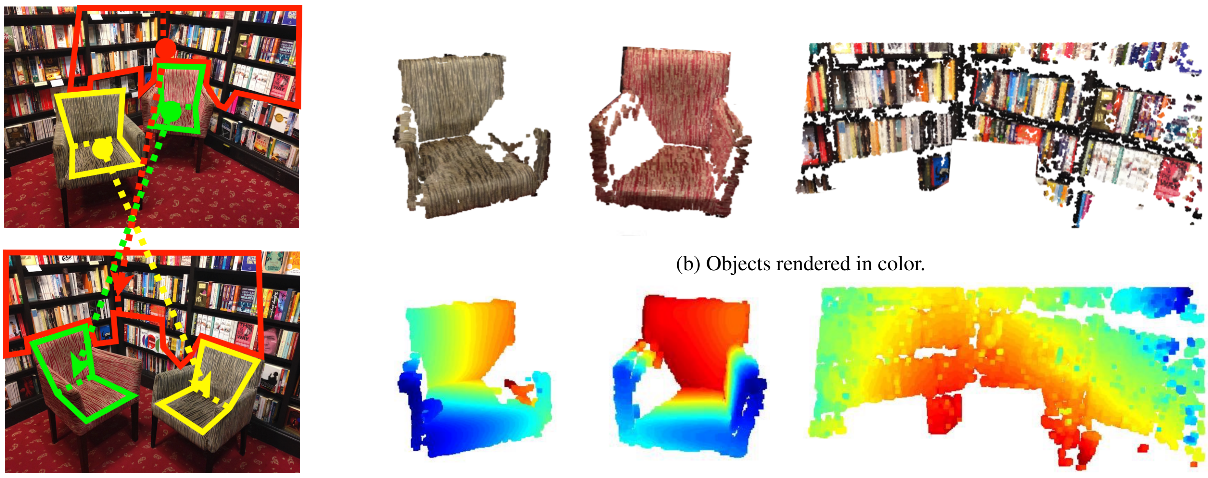
Eurographics Symposium on Geometry Processing 2015
Mirela Ben-Chen and Ligang Liu
(Guest Editors)

Volume 34 (2015), Number 5

Dynamic SfM: Detecting Scene Changes from Image Pairs

Tuanfeng Y. Wang[1] Pushmeet Kohli[2] Niloy J. Mitra[1]

[1]University College London [2]Microsoft Research Cambridge



(b) Objects rendered in color.

(a) From an input image pair, we detect the scene changes.

(c) Objects rendered by depth. For visualization, we color code the estimated depth with values increasing from blue to red.

Figure 1: Using the input image pair (a), our algorithm automatically performs change detection between the two images and calculates the motion of each rigidly moving part (b) while simultaneously estimating their 3D structure to enhance performance (c). Note that due to two-view ambiguity, we have no information about the absolute depth of each object. Instead, we show the depth maps for each object separately.

Abstract

Detecting changes in scenes is important in many scene understanding tasks. In this paper, we pursue this goal simply from a pair of image recordings. Specifically, our goal is to infer what the objects are, how they are structured, and how they moved between the images. The problem is challenging as large changes make point-level correspondence establishment difficult, which in turn breaks the assumptions of standard Structure-from-Motion (SfM). We propose a novel algorithm for dynamic SfM wherein we first generate a pool of potential corresponding points by hypothesizing over possible movements, and then use a continuous optimization formulation to obtain a low complexity solution that best explains the scene recordings, i.e., the input image pairs. We test the algorithm on a variety of examples to recover the multiple object structures and their changes.

1. Introduction

Everything changes and nothing stands still.
Heraclitus

We live in a dynamic world where objects regularly move or are moved around. Understanding such a world naturally amounts to detecting what changes and what does not. This constitutes a fundamental goal in scene analysis and understanding. In the context of time-coherent acquisition, e.g., using a video feed, advanced methods exist to reliably *track* objects to detect changes. However, limited options exist for uncontrolled settings with sparse measurements.

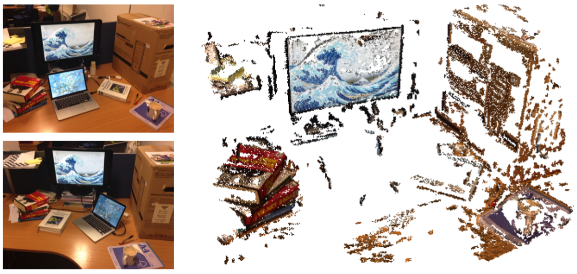


Figure 2: Result of directly running Structure-from-motion on images of a dynamic scene (using [Wu11]). Note, how the algorithm misses moving objects due to its static scene assumption.

In this paper, we investigate the problem of detecting scene changes from only a pair of input images. By *change*, we focus on what objects moved (i.e., segmentation), how the objects are structured (i.e., their 3D shape), and how the objects moved (i.e., motion parameters). We assume throughout the paper, that all objects as well as the background are moving rigidly during the *change*. A seemingly natural option is to use structure from motion (SfM) to first reconstruct the 3D scene from the input images, and then analyze the reconstructed scene. However, such an approach simply ignores moving objects as the the point-to-point correspondence search fails. This is particularly so in situations as in our setting, where the input images are assumed to capture large scene changes. For example, in Figure 2, only (part of) the background is recovered and the changes are completely missed.

We propose a solution based on two main steps: First, starting from a superset of candidate correspondences (i.e., including false positive matches) between the input image pairs, solving the above problems amounts to correctly *grouping* the correspondences based on the (unknown) motion models. To this end, we propose a continuous grouping formulation to simultaneously solve for segmentation, object structure, and object motion. Second, it is possible to generate a superset of candidate correspondences by pre-warping one of the input images to simulate the effect of possible homographies relating (near) planar surfaces in the two images. We realize this pre-boosting step to capture the correspondence pairs that are easily missed by direct analysis of the input image pairs. Herein we specifically make use of the structure of the scene to solve the dynamic SfM problem. Finally, in the dense reconstruction step, we improve the coarse correspondence obtained at the end of the grouping optimization to create the final output. Figure 1 shows a typical output of our method.

We evaluate the proposed algorithm on a range of test inputs of varying complexity. We also perform quantitative analysis on simulated test scenes with access to groundtruth and evaluate the effects of different parameter settings.

2. Related Work

Analyzing acquired scenes remains a central topic in computer graphics and vision. The goal involves establishing correspondences, performing motion segmentation, and detecting changes in the scenes. The complexity of the problem varies greatly based on how much the objects move, how often recordings are made (i.e., isolated images versus video sequence), and how reliably correspondence can be extracted. On one hand, a static scene can be reliably reconstructed from a set of unorganized images using SfM; while, on the other hand, advanced methods exist to perform motion segmentation from video sequences by tracking correspondence. We study the problem in the context of dynamic environments recorded with only single pair of images.

Motion segmentation. Majority of the methods assume spatio-temporal coherence and access to video sequences as input. Factorization, originally proposed by Tomasi et al. [TK92], remains the method of choice for multi-body motion segmentation [CK95, Gea98]. However, in real world settings with noisy inputs the method can produce only partial (tracked) trajectories. Gruber et al. [GW04, GW06] use *Maximum Likelihood Estimation* to extend factorization to handle uncertainty and missing data. More recently, advanced methods investigate the problem as selection from a family of models while balancing between model complexity and modeling accuracy. This results in a unified formulation [SSW08, OSVG10] that works robustly on a variety of real-world video sequences. We consider [JPS14] representing the state-of-the-art in motion segmentation. Their system uses a cluster of 480 carefully synchronized cameras to continuously capture changing scenes allowing tracking thousands of points and handling non-rigid objects. The above methods, however, rely on access to densely sampling video sequences, and hence are not applicable in our setting.

Subspace analysis. The grouping problem has also been investigated as an instance of subspace clustering, and algebraic methods such as GPCA are extended to deal with missing data [VH04] and outliers [YRM06, RTVM10]. However, without the hypothesis of spatio-temporal coherence (i.e., access to video sequences), the methods quickly become impractical due to the exponential complexity with respect to both the dimension of the ambient space and the number of moving objects in the scene. More recently, analysis has been restricted to sparse, low-dimensional subspace representations to encode trajectory data. [EV09] achieve impressive performance in terms of accuracy on the Hopkins155 motion segmentation database [TV07]. PEARL [IB12, DOIB12] is largely considered as the state-of-the-art method for few-view motion segmentation. By injecting the idea of model refitting, PEARL achieves higher accuracy on problems with a small number of input frames. However, in real world scenes, PEARL's α-expansion step is adversely affected by outliers especially under large view changes. Please refer to Section 6.3 for comparison with our method.

Prior knowledge. Relying on additional information, for example to recover the 3D structure of a scene while performing motion segmentation has been shown to improve performance. For example, in the case of articulated bodies, Fayad et al. [FRA11] optimize a single cost function to jointly solve the problem of segmentation and 3D reconstruction using an input set of point tracks. The approach has been extended to handle non-rigid objects [RRGA12]. These methods, however, require multiple frames from a video to obtain a good initialization and are not applicable in our setting.

Correspondence. The central challenge to the problem is to establish point correspondences between image pairs with large changes of viewpoint introducing severe distortions to object texture. The most common approach is to use the SIFT feature extractor [Low04]. However, the detected correspondences are very unreliable under large scene changes. [MY09] extend the concept of SIFT to create an affine invariant descriptor (see Figure 12 for comparison).

However, none of the above methods are designed for *only* using a single pair of input images. In this paper, we directly work on a pair of images with large camera motions and object displacements, which makes correspondence detection on the image level difficult. We show that by *simultaneously* optimizing both the object structure (i.e., 3D point locations) and object motion, we can detect scene changes and establish good point-level correspondences.

3. Overview

Our goal is to detect changes in indoor scenes from a pair of image recordings. This requires answering the following: (i) what are the moving parts, i.e., obtain point clouds for the moving parts of the scene; (ii) how did they move, i.e., estimate the movement for the respective objects between frames; and (iii) what are the camera parameters for the two (uncalibrated) input images.

There are two main problems: (i) direct structure-from-motion (SfM) computation on the input image pairs fails as the scene is not static (see Figure 2); and (ii) obtaining good quality point correspondences is challenging in presence of large scene changes as in our setting.

To address the first problem, we observe that if sufficient number of good correspondence pairs are available, then the problem reduces to grouping the correspondences according to the (unknown) moving parts. Specifically, if we have the correspondence pairs correctly grouped, we can simply perform SfM for each individual group under the additional constraint that each image has a common calibration. Hence, we formulate continuous energy minimization to group the created dense feature point matches into different rigid motion trajectories, estimate the 3D object positions and identify outlier samples (see Section 4.1). Note that the continuous formulation allows to take advantage of the additional information contributed by the scene structure.

To address the second problem, we integrate correspondence boosting and camera model hypothesis generation with the multi-hypothesis grouping. Essentially, we increase the set of potential correspondence pairs and only later recover the subset of correct correspondences. First, we initialize our algorithm with a sparse set of high-quality feature correspondences using any feature descriptor (SIFT in our case). Then, in a critical step, we boost the set of available correspondences by hypothesizing part motions (or equivalently camera motions) as described in Section 4.2.

Finally, in Section 4.3 we describe how we employ a patch-based correspondence post-boosting strategy using the optimized motion grouping to generate an even denser point cloud, that can be used as input to other applications.

4. Algorithm

In this section, we first formulate the dynamic SfM as a grouping problem, wherein we categorize candidate correspondence pairs into different motion groups while identifying outlier correspondences (false positive feature matches). We then describe how to initialize the grouping optimization, that is, how to create a sufficiently rich set of correspondence pairs from two images containing significant scene changes. Finally, we describe how the grouped correspondences can be used to obtain dense structures (i.e., point clouds) for the different moving objects in the scene. Figure 3 shows an overview of our method.

4.1. Multi-body structure and motion

At the core of our method is an energy-based continuous optimization that recovers both the 3D structure and motion

Table 1: List of symbols

M	Number of motion candidates
$L_i,\ i \in 1:M$	Motion model candidate, holds one set of camera parameters for each image
N	Number of feature correspondences
$\mathbf{d}_k,\ k \in 1:N$	3D position implied by a correspondence k
α_i^k	Element in label vector representing assignment likelihood of $\mathbf{d}_k$ to motion model L_i
$\|\cdot\|_{L_i}$	Operation representing the sum of reprojection errors for motion L_i in both images
δ_k	Likelihood of $\mathbf{d}_k$ being an inlier match
SN_k^j	Neighborhood of corresp. k in image j
$\lvert\cdot\rvert$	Set cardinality
β	Sparsity coefficient
$\theta_{p,q}$	Consistency weight for corresp. $p \leftrightarrow q$
ω_1	Complexity penalty weight
ω_2	Outlier penalty weight
ω_3	Consistency penalty weight

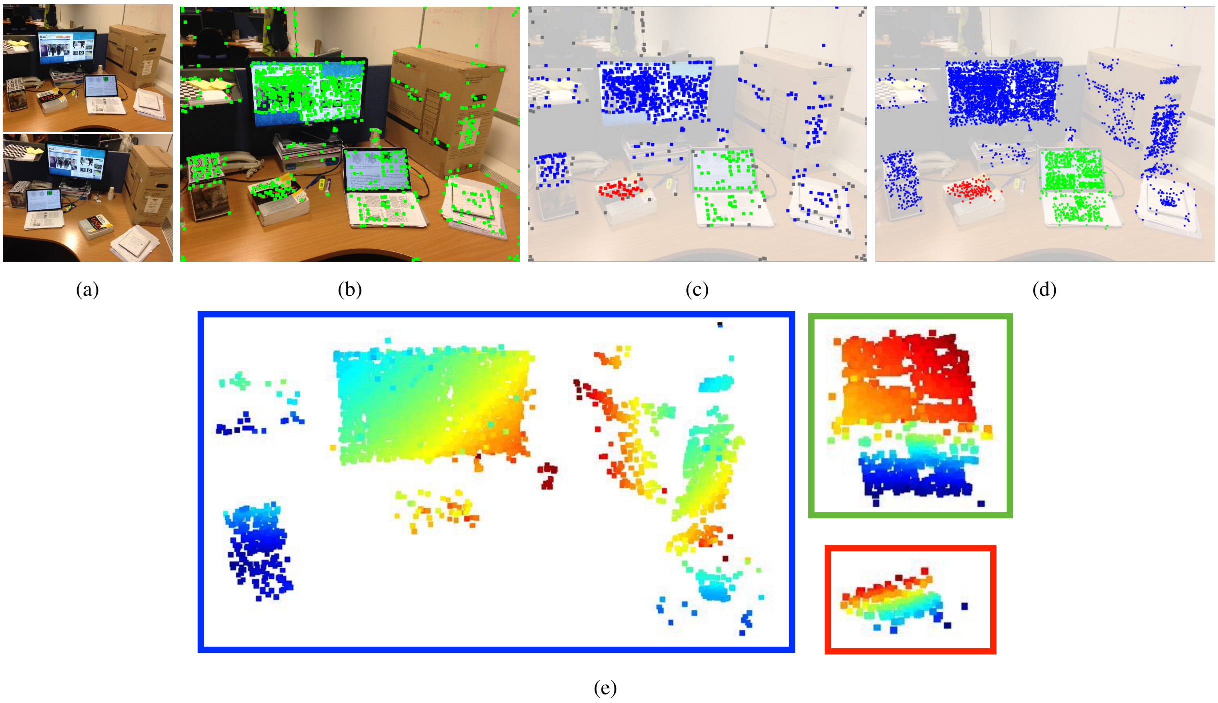


Figure 3: Algorithm pipeline. From two images (a) of a dynamic scene with multiple objects undergoing rigid motions. We first generate a set of candidate correspondence pairs using our pre-boosting strategy indicated by green dots in (b). In this example, 1046 correspondence pairs were generated, instead of only 554 using SIFT directly. Next, we use continuous optimization to simultaneously recover the motion of each rigid part (c) along with their coarse 3D structure (colors show assignment to the different motion groups). Finally, we use the grouping result to obtain a denser set of correspondence pairs (d). Here, 5281 correspondence pairs are generated from 832 inlier correspondence pairs obtained from pre-boosting. We show the structure of each object (e) color coded by estimated depth, distances increasing from blue to red.

of each rigid part in a dynamic scene. We seek to extract a low complexity explanation of the scene in terms of objects and their motion, that best explains the observations, i.e., the input pair of images. We observe that the problem amounts to robustly grouping a set of candidate correspondences into motion groups. Later, in Section 4.2, we describe how to initially extract such a set of candidate correspondences, possibly containing a significant amount of outliers. We pose the grouping problem as minimizing the reprojection error, outlier penalty, group complexity penalty, and non-smoothness by labeling the correspondences into different groups, while simultaneously estimating their 3D positions.

Let there be M^* possible moving objects, and $M > M^*$ motion model candidates captured by the corresponding camera motions L_i with $i \in 1 : M$ for each image. The goal is to assign each correspondence pair to one of these (unknown) motions, or mark it as an outlier. We encode this grouping as an $N \times M$ label matrix, with row vectors α^k for each correspondence. For each inlier correspondence, we also maintain the respective (unknown) 3D position as $\mathbf{d}_k$.

For each correspondence, we use selection variables α_i^k, the i-th $(i \in 1 : M)$ component of α^k, to capture the likelihood of a correspondence belonging to motion described by the motion model L_i. Note that $\alpha_i^k \in [0, 1]$ and each correspondences can at most be assigned to one model, as captured by

$$\sum_{i=1}^{M} \alpha_i^k \leq 1 \quad \forall k. \tag{1}$$

We parametrize the target energy via motion models $\{L_i\}$, 3D points $\{\mathbf{d}_k\}$, and label vectors $\{\alpha^k\}$. Specifically, the energy estimate consists of four terms:

$$E(\{L_i\}, \{\mathbf{d}_k\}, \{\alpha^k\}) := \\ E_{data} + E_{complexity} + E_{outlier} + E_{consistency}. \tag{2}$$

Note that the formulation takes full advantage of (unknown) structure in the motion segmentation problem by estimating a 3D position for each correspondence. This is in contrast to representing correspondence only as a pair of 2D feature point positions in two images and subsequently measuring geometric error using for example squared Sampson's distance [HZ03].

The data term E_{data} captures the sum of reprojection errors in the two images, weighted by the assignment likelihood α. Specifically,

$$E_{data}(\{L_i\},\{\mathbf{d}_k\},\{\alpha^k\}) = \sum_{k=1}^{N}\sum_{i=1}^{M} \alpha_i^k \parallel \mathbf{d}_k \parallel_{L_i}, \quad (3)$$

where $\parallel \cdot \parallel_{L_i}$ in Equation 3 is the sum of reprojection errors of 3D point $\mathbf{d}_k$ to the two input images under camera motion L_i. We use a perspective camera model in our implementation and the reprojection error is calculated as sum of squared distances similarly to [HZ03].

The complexity term $E_{complexity}$ penalizes having too many separate groups to describe the motions. In other words, this term exists in pursuit of the sparsity of label vectors α^k with respect to model k. Specifically,

$$E_{complexity}(\alpha) = \omega_1 \cdot \sum_{i=1}^{M}\left(\sum_{k=1}^{N} \alpha_i^k\right)^{\beta} \quad (4)$$

where, β is an exponent close to zero. (Alternately, one can use a reweighted L1 formulation here.) The term weight ω_1 can be considered as a threshold of minimum number of correspondences in a group, since points in any group having less than $\lceil(\omega_1/\omega_2)^{1/\beta}\rceil$ correspondences will be identified as outliers.

For any correspondence $\{\mathbf{d}_k\}$, if it is too costly to fit using every model candidate, we consider it as an outlier by allowing α^k to tend to zero for all i. However, to avoid the trivial assignment of marking all the correspondences as outliers, we introduce the outlier term:

$$E_{outlier}(\alpha) = \omega_2 \cdot \left(\sum_{k=1}^{N} \delta_k (1 - \sum_{i=1}^{M} \alpha_i^k)\right). \quad (5)$$

Here, δ_k is a pre-calculated coefficient for each correspondence to indicate how much we want to penalize the k-th correspondence if it is an outlier. A simple case is to set $\delta_k = 1$ for all k. However, as illustrated in Figure 4, for the two-view situation, the reprojection error is not fully reliable. Unfortunately, sometimes an outlier might have very low reprojection error w.r.t a particular motion model solely by chance.

In order to address this issue, we use δ_k to account for the possibility of the k-th correspondence being an outlier. A simple criteria is how much the two groups of neighbors of the feature points in two images overlap. Specifically, we set $\delta_k = \max(1 - 0.2 \cdot |SN_k^1 \cap SN_k^2|, 0)$, where $SN_k^{1,2}$ is the set of 10 nearest neighbors of the correspondence k in the two images, measured in image space and $|\cdot|$ here denotes the cardinality of the intersection set. The weight ω_1 can also be considered as an outlier threshold as correspondences with reprojection error larger than ω_1 will finally be labeled as outliers when the optimization converges.

The consistency term $E_{consistency}$ regularizes false positive matches caused by two-view ambiguity (the tower case in Figure 4). Specifically,

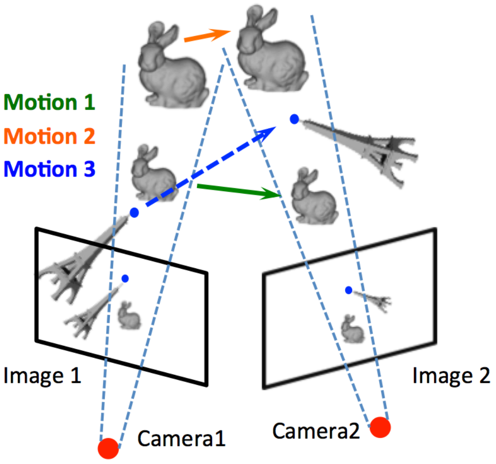


Figure 4: There are two types of 2-view ambiguities preventing us from estimating the relative depth of each moving object from only two viewpoints. For example, the bunny can be quite small and have quite large displacement (Motion 1) or can be very large with a relatively small displacement (Motion 2). For some particular cases, a correspondence could fit well into several different motion models, as for example the top of the tower. The correspondences in the images marked by blue dots fit Motion 1 well, although they actually belong to Motion 3.

$$E_{consistency}(\alpha) = \omega_3 \cdot \sum_{(p,q)\in DN} \theta_{p,q} \parallel \alpha^p - \alpha^q \parallel. \quad (6)$$

We use Delaunay triangulation in our implementation to create the neighborhood DN of each data point. The prior weight $\theta_{p,q}$ captures that correspondence p and correspondence q should belong to the same group. We use the inverse of the average squared distance between two feature points p, q on the two images to estimate $\theta_{p,q} = \left((dist_{img1}(p,q)^2 + dist_{img2}(p,q)^2)\right)^{-1}$.

We minimize $E(\{L_i\},\{\mathbf{d}_k\},\{\alpha^k\})$ with the constraints that the label vector is valid $\alpha_i^k \in [0,1]$ and Equation 1 to solve the dynamic SfM problem.

Generating motion model candidates. In order to initialize the above optimization, we need a good initial set of motion model candidates. Note that having a good set of correspondences allows direct generation of motion models by using the 8-points algorithm for evaluating the fundamental camera matrix (cf. [IB12]). Since the original set of input correspondences has both inliers and outliers, we use RANSAC [FB81] to create a superset of motion candidates, with some of them being correct. This is sufficient for the grouping optimization described above to extract the suitable motion models.

Domain problem. Generally, candidate models can be generated by running RANSAC until each rigid part is covered. Unfortunately, as shown in [IB12], the theoretical estimate of necessary iterations is always too large to be practical (in their example, 3 objects with 20/24/56 inliers can only achieve 0.92 confidence, even when sampling 10^6 times). In practice, for some lucky situations, a small number of samples may also be sufficient.

However, in real world scenarios RANSAC still falls short due to problems caused by large differences (i) in the size and (ii) feature richness of image regions that move rigidly together. This will lead to differences in the magnitudes of number of correspondences generated from different image regions. A static background or a colorful information poster will generate proportionally more feature points than the rest of the image parts. They will act as a *domain* to smaller regions, hence we refer to this as the *domain problem*. Given such proportional differences, RANSAC faces difficulties finding the smaller, rigidly moving regions in the images.

In this paper, we use a *reweighted RANSAC* strategy to get reliable motion models even from a very small fraction of good candidates. Specifically, we lower the weight of data points that have been considered as inliers by multiplying by a weight decrement factor, in order to boost the selection probability of points from other parts of the scene (see Algorithm 1 for detail). This is particularly effective in our case, when dealing with multiple moving objects. For example, in Figure 3, we obtained three rigid parts with 40, 82 and 710 (domain part) inlier correspondences and 114 outliers. Our optimization converges to the correct solution after that only 5-10 candidates have been generated by our reweighted RANSAC algorithm. The behavior was similar in the other examples presented in this paper.

Generating initial point locations. After generating a finite set of motion candidates, we estimate the initial 3D positions $\{\mathbf{d}_k^{initial}\}$ by triangulating each correspondence using the camera model that gives the smallest reprojection error. We then initialize the assignments α_i^k proportional to the inverse of the re-projection error of $\{\mathbf{d}_k^{initial}\}$ to all camera models L_i as

$$\alpha_i^k = \| \mathbf{d}_k^{initial} \|_{L_i} \; / \; \sum_{p=1}^{M} \| \mathbf{d}_k^{initial} \|_{L_p}.$$

Optimization. We use MATLAB's interior-point solver for the optimization using the following parameter settings in our experiments: $\omega_1 = 1000$, $\omega_2 = 500$ (means outlier threshold adds up to $\sqrt{500}$ pixels), $\omega_3 = 500$, and $\beta = 0.4$. As discussed in Equation 2, we optimize for the variables $\{L_i\}, \{\mathbf{d}_k\}, \{\alpha^k\}$. Once the optimization converged, we round α_i^k to 1 if it is greater than 0.9 and to 0 otherwise. This assigns each correspondence to a single camera model as we experimentally found $\alpha_i^k \simeq 0.99$ at convergence, indicating that $\mathbf{d}_k$ was linked to L_i.

Algorithm 1 *reweighted* RANSAC

```
// Initialization
Weight decrement factor $\mu = 0.2$
Weight $w_k = 1, k \in 1 : N$
Average sample times $T = 1.5$  ▷ each point is expected to contribute as inlier on average $T$ times
// Reweighted RANSAC
while $\sum_{k=1}^{N} w_k > \mu^T \cdot N$ do
    $nBest = 0$;
    for $i = 1 : nIterations$ do
        Randomly sample 8 points $p_{1...8}$
        $tmpF$ = Compute fundamental matrix from $p_{1...8}$
        $tmpIniliers$ = index of inliers under fundmental matrix $tmpF$
        $S = \sum_{tmpIniliers} w_k$
        if $S > nBest$ then
            $fBest = tmpF$
            $nBest = S$
            $inliers = tmpIniliers$
        end if
    end for
    $w_{inliers} = \mu \cdot w_{inliers}$
    Output fundmental matrix $fBest$.
end while
```

4.2. Correspondence pre-boosting

The main difficulty in matching feature points between images, where the camera view points are far apart is that the orientation of the surfaces we are interested in vary a lot relative to the cameras. The change of viewpoint results in distortion of texture, and feature descriptors generated from the same image locations will change significantly as a con-

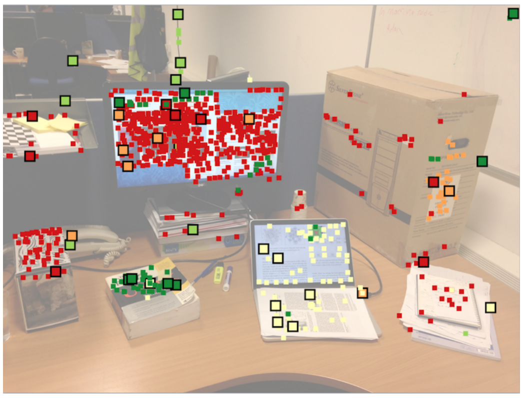


Figure 5: This figure shows the first five candidate models generated by *reweighted RANSAC*. Sampled sets of 8 points are shown as larger squares with black borders. Smaller squares with the same colour show inliers under the corresponding generated camera model.

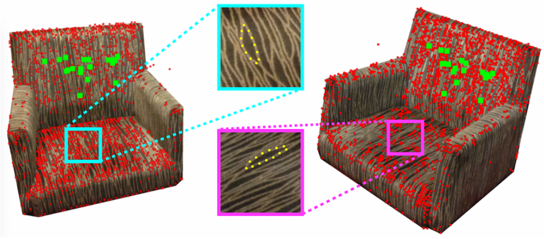


Figure 6: We directly extract SIFT feature points and run feature matching [Low04, VF08] on a pair of images of a chair. Red dots are generated feature points and green dots are matched correspondences. The close-up views show the difference in the observed texture of the same patch on a surface perpendicular to the image plane. We conclude, that the failure of feature matching is mainly caused by texture distortion. A set of yellow dots are marked manually to visualise the same pattern.

sequence (as shown in Figure 6). Simply extracting feature points from two images and matching them based on the distance metric will only reveal correspondences between image parts, where the perspective did not change a lot, in most of the cases surfaces parallel to the image plane.

The first step of our pipeline is to as much as possible boost the number correspondences obtained from areas, where the surface texture underwent significant change in distortion due to the change of camera viewpoint. These areas are initially very sparsely covered by high confidence matches. We keep one image fixed, and 'rotate' the other image in 3D, as shown in Figure 7. Note that this step implicitly guesses a potential motion or alternately a homography between corresponding (near) planar parts in the scene. The intuition being that if the guess is correct, then the corresponding moving parts are likely to pick up correct correspondence pairs. We extract feature points in each rotated image and perform correspondence matching w.r.t. the fixed image. Rotating the image in 3D simply allows us to change the image plane normal. We approximately create S rotated image copies by sampling a half-sphere uniformly with the parametrization described as:

$$u_i = \arcsin\left(1 - \frac{2i-1}{2S}\right) \qquad v_i = u_i\sqrt{2\pi S} \tag{7}$$
$$\mathbf{n}_i = [\cos(u_i)\cos(v_i);\ \cos(u_i)\sin(v_i);\ \sin(u_i)]$$

where, $i = 1, 2, \ldots, S$. Warping an image by spatial rotations allows us to compensate for the difference in texture caused by the change of viewpoint. It allows us to match more feature points and results in a more complete coverage of the scene as shown in Figure 8. Simultaneously, more mismatches are also generated when performing pre-boosting. However, this can be handled well by our grouping optimization using the mismatch-aware outlier penalty. The idea of our pre-boosting method is similar to [MY09], but ours performs better due to a more uniform sampling of warping rotations. Note that this warping can also provide us with information about the structure of the scene. However, we carefully avoid to explicitly rely on this heuristic when sampling homographies, because our continuous optimization recovers the same knowledge with much higher reliability.

Implementation details. We use VLFeat's implementation of SIFT extraction and matching [VF08] in our experiments. We set the parameters *"peak threshold"* $= 6$ and *"edge threshold"* $= 10$. As uniqueness threshold during matching, we use a quite strong value of 0.45 to allow us to

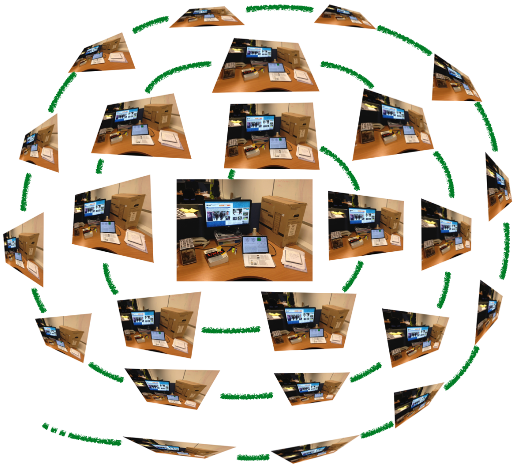


Figure 7: Image 1 is warped by rotating the normal of the image plane according to the parametric Equation 8.

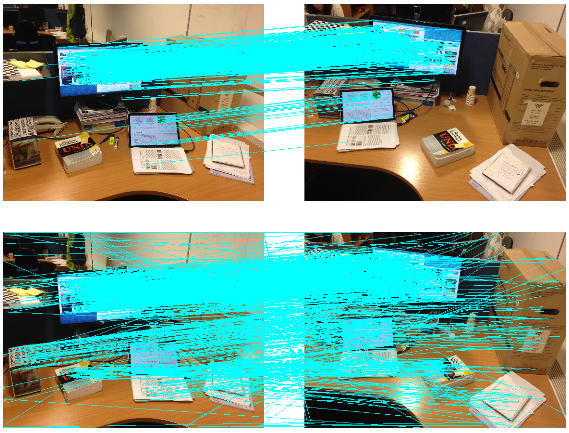


Figure 8: Top figure shows the feature point correspondences (cyan lines) matched between the two original images. Bottom figure shows, that after pre-boosting, the number correspondences is significantly increased, especially in areas with upward pointing surface normals. Note that a substantial amount mismatches (outliers) are generated as well, which we robustly handle in our optimization.

select correspondences with significant confidence and reliability. Setting $S = 30$ was enough during our experiments. We used a density threshold $d = 30$ *pixel* in order to prevent the generation of too many repetitive correspondences during matching over the warped images. This guarantees, that the minimum distance between any two matched features in the original image exceeds d pixels.

4.3. Correspondence post-boosting

Finally, we generate a denser 3D point cloud from the structure and motion recovered by the continuous optimization to utilize the recovered information in a more efficient manner. We designed our patch-based method to take advantage of the warping strategy described along with pre-boosting (Section 4.2), which gives us a competitive edge compared to general dense reconstruction.

Patch-based boosting. Our goal is to use our optimized, high-quality correspondences to generate a denser output point cloud using a local smoothness prior. Many methods employ the spatial regularity assumption (e.g., [FP10]) to locally propagate information around recovered feature point matches. We propose a hierarchical feature matching algorithm. In patch-based boosting, we perform a second round of feature point matching on each pair of patches, each connected by an optimized output correspondence. These matches were missed in the initial correspondence search, because they had several possible matches in the global scene, and were labeled as insignificant. We take advantage of our pre-boosting step by looking up between features in all warped versions of the corresponding patch.

Correspondences are stored in a *FIFO*-queue and we propagate them using a *breadth-first* strategy. Specifically, each time a correspondence is picked from the front of the queue, we extract a square patch with 80 × 80 pixels from one input image. We then find the corresponding patch in each of the warped images, and perform SIFT feature point extraction similarly to the pre-boosting step, using the same parameters. These feature points are then matched locally between the warped patches and the resulting correspondences are appended to the end of the queue. Each newly generated correspondence is assigned to the same motion model and group of correspondences as its parent. Similar to [FP10] we also applied a match density threshold set to $d = 6$ in our case, to control the blooming of correspondences, see Figure 3d.

5. Application

5.1. Dense reconstruction

The output of our pipeline is a motion model and 3D point cloud for each rigidly moving part of the dynamic scene. This can naturally be achieved using any dense stereo reconstruction method, i.e., CMVS/PMVS [FP10], CMPMVS [JP11], MVE [SFG14] and SURE [RWFH12]. We apply Furukawa's PMVS algorithm (implemented in [Wu11]), which takes a 3D point cloud of a rigid object and two 2D views of it as input. For each motion group segmented by our algorithm, we use the relative camera positions and densely matched features associated with this motion to initialize PMVS. The advantage of our result (Figure 9) over running structure from motion directly on each, manually segmented rigid part is that our pre- and post-boosting steps efficiently increases the covered area in the images, from which 3D structure can be recognized in our two view setting.

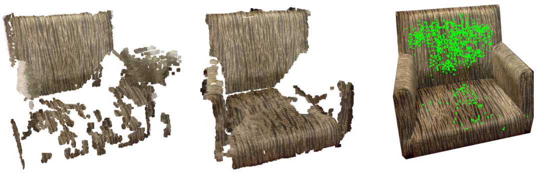


Figure 9: Classic dense reconstruction (left) and dense reconstruction based on our post-boosting step (middle) using the chair scene from Figure 6. Our pre- and post-boosting methods (right, rendered with green dots) enables the reconstruction of larger areas of the input images.

5.2. Motion interpolation

We calculate a motion in our method for each rigidly moving part of the dynamic scene, interpretable as the motion of the part from the first image to the second. As an application of this motion information, we can interpolate the dynamic scene between the input two images to generate a possible set of trajectories for the rigidly moving parts. We perform the interpolation as described in [MGPG04], see Figure 10.

5.3. Working with multi-view

Our two-view based pipeline can be naturally extended to multi-view cases. Assume that we have Π images as input and D_k is 3D points for the k-th trajectory. The energy is formulated in the same way as in Equation 2, with the natural modification of some of the terms. Specifically, $\|\cdot\|_{L_i}$ in the data term is the sum of reprojection error w.r.t. all Π images, the outlier prior δ_k in the outlier penalty term will be based on all image pairs as well as neighborhood DN and inlier likelihood $\theta_{p,q}$ in the consistency term.

6. Results

6.1. Performance

We tested our algorithm on several input cases and evaluated our performance with respect to the correctness of our motion grouping output.

As shown in Figure 11, our algorithm generates accurate grouping results and overperforms comparable state-of-art

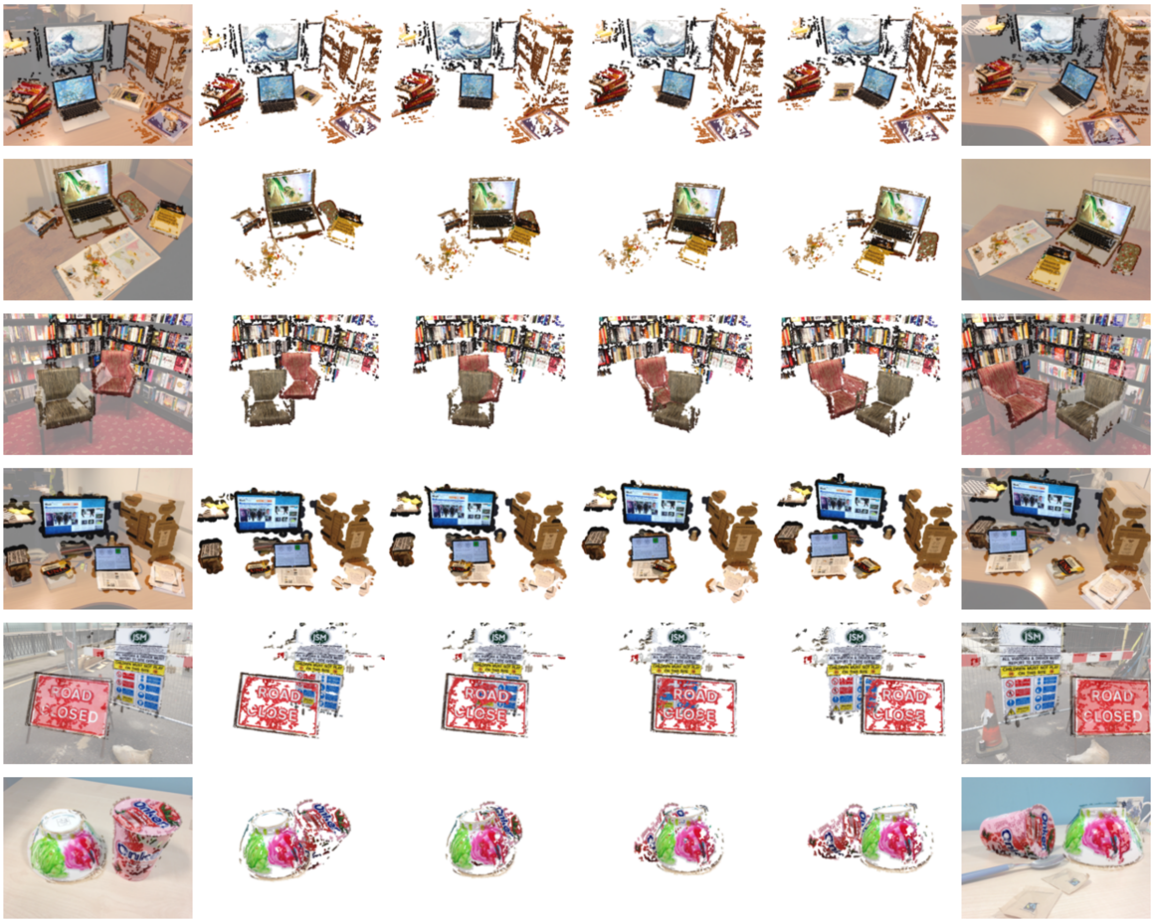


Figure 10: Motion interpolation with our dense reconstruction from only a pair of input images (overlaid on first/last columns).

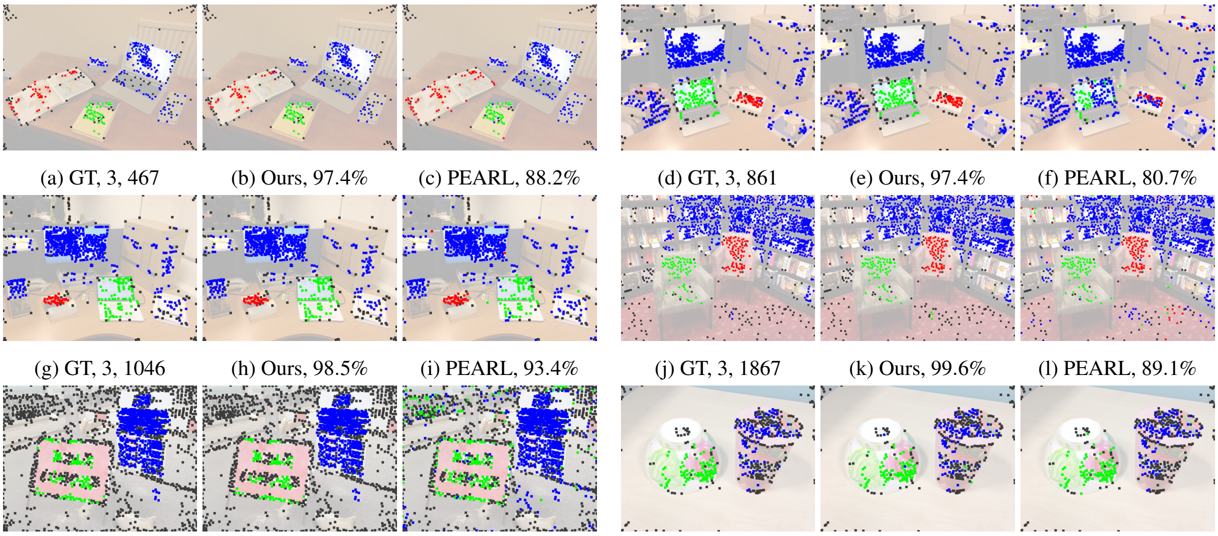
methods, i.e., PEARL. Our motion interpolation (Figure 10) allows us to qualitatively inspect the high quality structure generated by our method. The runtime of our optimization depends on the number of correspondences and motions. For the presented scenes, our algorithm takes 30 minutes to converge on average (1 hour in worst case).

6.2. Evaluation

Ground truth. We performed experiments to evaluate our labeling performance on real data, and correctness of structure on a synthetic scene. To create the ground truth labeling, we first manually select correct correspondences within the same motion and label them into the same motion groups. We then estimate the groundtruth transformation for each motion with the selected correct correspondences. With groundtruth transformation, we can perform a raw grouping of correspondences and an annotation of outliers. We evaluate our output point cloud structure later in this section by comparing our reconstruction results to a synthetic scene, where the ground truth structure is known.

Pre-boosting. As shown in Figure 8, our pre-boosting method increases the number of correspondences between the image pair from 554 to 1046, with an outlier (mismatch) ratio 10.9% (114/1046). We define the outlier ratio as the ratio of the number of mismatch correspondences with respect to the number of all correspondences. Our measurements showed, that after pre-boosting we on average over our test scenes arrive to an outlier ratio of 10%, which, as later shown, can be handled well by the initialization of our optimization. In comparison, Figure 12 shows the result of ASIFT [MY09] and we compare favorably in terms of consistency of the match orientations.

Initialization. Our pre-boosting effectively generates a lot more correspondences, however, with a moderate ratio of outliers. This is directly related to the reweighted RANSAC strategy we apply to calculate camera poses for initialization. In Figure 13, we show that our initialization is insensitive to the outlier ratio of the input correspondences given a single, universal parameter (weight decrement factor = 0.2 during our experiments). For evaluation, we manually delete



(a) GT, 3, 467 (b) Ours, 97.4% (c) PEARL, 88.2% (d) GT, 3, 861 (e) Ours, 97.4% (f) PEARL, 80.7%

(g) GT, 3, 1046 (h) Ours, 98.5% (i) PEARL, 93.4% (j) GT, 3, 1867 (k) Ours, 99.6% (l) PEARL, 89.1%

(m) GT, 2, 2283 (n) Ours, 99.0% (o) PEARL, 86.7% (p) GT, 2, 497 (q) Ours, 98.0% (r) PEARL, 95.8%

Figure 11: Algorithm performance. Groundtruth with number of motions and number of correspondences (inlier and outlier), our result and PEARL's result with labeling correctness. Note that the PEARL was provided with our robust initialization.

outliers or add random correspondences to the output of the pre-boosting step to simulate the change in outlier ratio. We successfully show, that our initialization is able to generate high-quality camera model candidates with very limited redundancy at different levels of outlier ratios, both with and without the domain problem.

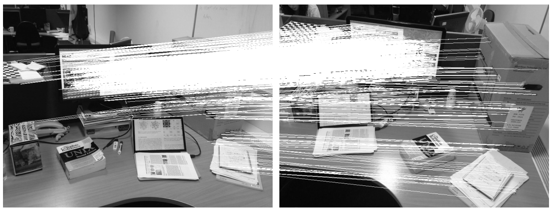
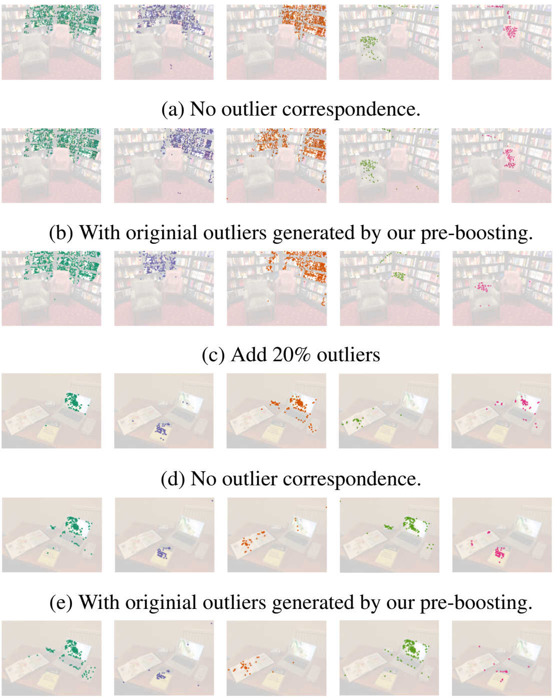


Figure 12: ASIFT result on input presented in Figure 8.

Correspondence grouping. The validation of our optimization consists of two aspects: correctness of group labeling and reconstruction quality. Group labeling is more essential as we can apply any structure from motion method consecutively to our output segmented correspondences. We first show the results with and without the adaptive outlier weight. Since we only use two input views, there might be some outliers (mismatches), that reproject well in the two images and therefore cannot be penalized by our data term. In these cases, our algorithm relies on the mismatch penalty whilst producing the reliable outputs. We then investigate the effect of our complexity and consistency terms. Our consistency term is most effective in resolving problems caused by two-view ambiguity as described in Section 4.1. We demonstrate the utility of the different energy terms in Figure 14. Omitting the consistency term results in neighboring points getting mislabeled, and prevents us from benefiting from redundancy residing in local context. Similarly, if the quality of correspondences in the local neighborhoods is not taken



(a) No outlier correspondence.

(b) With originial outliers generated by our pre-boosting.

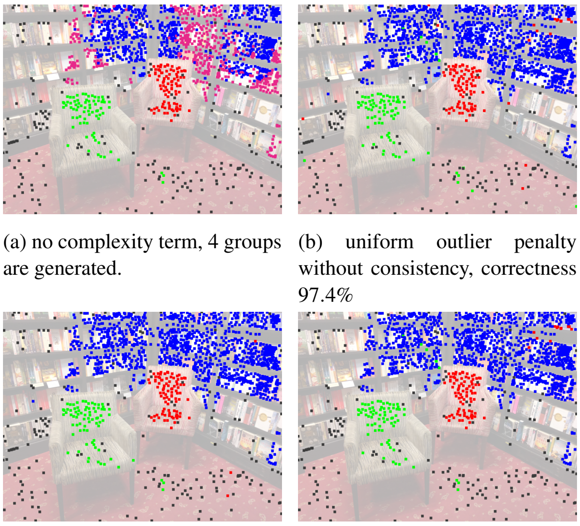
(c) Add 20% outliers

(d) No outlier correspondence.

(e) With originial outliers generated by our pre-boosting.

(f) Add 20% outliers

Figure 13: Initialization with different outlier ratios in different types of scenes (with and without domain problem).



(a) no complexity term, 4 groups are generated.

(b) uniform outlier penalty without consistency, correctness 97.4%

(c) uniform outlier penalty, correctness ratio 98.8%

(d) no consistency term, correctness ratio 98.3%

Figure 14: Evaluation of our different energy terms. Note that the correctness ratio achieved using our full formulation is 99.6%, as shown in Figure 11.

into account when identifying outliers (Figure 14c), camera estimates become less accurate due to a higher rate of false-positive matches being used or more reliable contributions being culled.

Point cloud structure. One of the main reasons, why we generate structure simultaneously with the motion segmentation is, that point clouds with correct structure will further enhance the segmentation of correspondences to consistent motions. To evaluate our reconstruction quality, we ran our algorithm on a synthetic scene (Figure 15). We measured reconstruction quality with respect to average depth error and groundtruth depth for each point. Our output point cloud (after post-boosting) has 5% depth difference ($\frac{\text{average depth error}}{\text{average groundtruth depth}}$) on average over the four objects. This allows us to enhance the motion segmentation and enables the further application of our outputs.

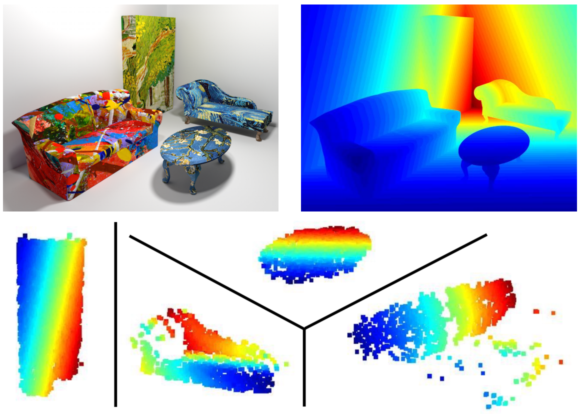


Figure 15: Depth map to show our output structure. Upper-left: synthetic scene; upper right: depth ground truth, from blue (close) to red (far); lower row: depth map for each object we detect after post-boosting. Due to two-view ambiguity, we actually have no information about the relative depth between objects. Therefore, we only show the depth map individually for each object.

6.3. Comparison

We compared our method to the state of the art method PEARL [IB12]. In Figure 16, we ran PEARL based on our pre-boosting result but with PEARL's initialization and based on direct SIFT matching results. In Figure 11, we show the performance of an improved PEARL version, that is based on our initialization. This example shows well, that although PEARL performs well on a manually annotated benchmark dataset, it actually falls short in real world cases, where the ratio of outliers is not strictly zero and/or the domain problem complicates the inference. Our method, on the other hand can be applied in real life scenarios, since it can cope with the above problems arising.

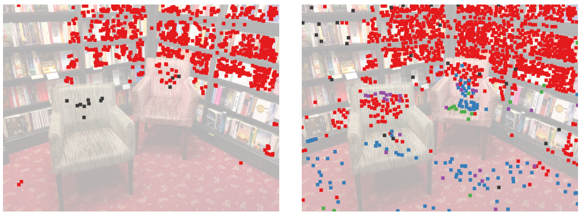


Figure 16: Comparison to PEARL. Left: running PEARL after extracting and match SIFT features directly. Right: running PEARL based on our pre-boosting strategy. Dark dots indicate outliers. There are three parts in the scene with 1482/83/101 inlier correspondences and 201 (10.7%) outliers. In this case, we show that PEARL fails in a real world scenario when outliers and the domain problem exists.

6.4. Evaluation on *Hopkins155*

To better evaluate the labeling efficiency, we ran our algorithm on a scene from *1R2RC* in the *Hopkins155* dataset [TV07]. We test our method using the first and the last frames of the sequence as the input image pair. For fair comparison with other methods, we designed the experiment to evaluate labeling correctness only, i.e., we used the feature point locations and correspondences provided by the dataset as input. Note that the omitted pre-boosting and post-boosting steps represent a significant part of our contribution, rendering our method more general then the algorithms attempting to solve the Hopkins155 dataset. Figure 17 shows how our method performs with a correctness ratio of 98.7%.

Note, that the input of *Hopkins155* dataset is manually annotated. Hence, it does not suffer from outliers or sensor noise, which contradicts our data assumptions. Further, in this specific example, the relative viewpoint of the rotating box (marked by green dot) changes very little between the first and the last frames, which does not provide our method with enough information to perform the structure estimation.

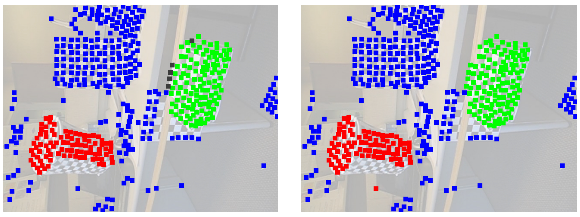


Figure 17: Results based on scene *1R2RC* from the Hopkins155 dataset. Input was the first and last frames of the video sequence. Left: our result with a correctness ration of 98.7%. Right: PEARL's result with a correctness ration of 99.5%. Note, that the input correspondences here are perfectly free of noise, i.e. no outlier correspondences are present, which is not a realistic real-life scenario.

6.5. Limitation

Our method has two main limitations. First, as our algorithm is based on feature points extracted directly from the images, objects with less texture cannot be recognized well. Therefore those objects will be ignored. Second, as we estimate the structure from a pair of images, it is necessary, that the relative pose of an object in the two images is sufficiently different to ensure the robustness of the 3D reconstruction, otherwise there is simply not enough information in the raw input. Empirically, we have established, that the relative view angle w.r.t. the objects should change more than 10 degrees, as confirmed by [FP10].

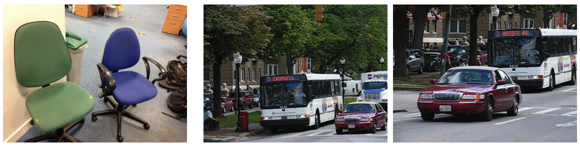


Figure 18: Typical failure scenarios. Left: low number of initial correspondences due to lack of texture usually causes our algorithm to miss some of the important structures in the scene. Middle and right (*car10* in *Hopkins155*): although the bus is moved, the change of perspective of the bus is barely noticeable leading to less robust structure estimation.

7. Conclusion

We presented an algorithm for dynamic SfM to recover both part structure and their motion starting from a pairs of input images. The main gain is to detect scene changes from sparse uncontrolled measurements. We proposed a solution that simultaneously recovers the 3D structures along with part motions, and in the process achieves increased accuracy.

Several future avenues remain unexplored: First, we believe having additional prior will further regularize the problem (e.g., using a partially known set of known objects like chairs, tables, etc. in office environments). Second, it will be worthy to make the algorithm faster to support near realtime performance. For example, the RANSAC sampling and initialization can be performed in parallel, possibly using GPU speedup. Finally, we would like to test the multi-view setting with images coming from multiple mobile phone inputs.

Acknowledgements

We thank the reviewers for their comments and suggestions for improving the paper. Special thanks to Aron Monszpart, Duygu Ceylan and Christopher Russell for invaluable comments, support and discussions. This work was supported in part by ERC Starting Grant SmartGeometry (StG-2013-335373) and a Microsoft PhD scholarship.

References

[CK95] Costeira J., Kanade T.: A multi-body factorization method for motion analysis. In *Computer Vision, 1995. Proceedings., Fifth International Conference on* (1995), IEEE, pp. 1071–1076. 2

[DOIB12] Delong A., Osokin A., Isack H. N., Boykov Y.: Fast approximate energy minimization with label costs. *International journal of computer vision 96*, 1 (2012), 1–27. 2

[EV09] Elhamifar E., Vidal R.: Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 2790–2797. 2

[FB81] Fischler M. A., Bolles R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (1981), 381–395. 6

[FP10] Furukawa Y., Ponce J.: Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32*, 8 (2010), 1362–1376. 8, 12

[FRA11] Fayad J., Russell C., Agapito L.: Automated articulated structure and 3d shape recovery from point correspondences. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 431–438. 3

[Gea98] Gear C. W.: Multibody grouping from motion images. *International Journal of Computer Vision 29*, 2 (1998), 133–150. 2

[GW04] Gruber A., Weiss Y.: Multibody factorization with uncertainty and missing data using the em algorithm. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (2004), vol. 1, IEEE, pp. I–707. 2

[GW06] Gruber A., Weiss Y.: Incorporating non-motion cues into 3d motion segmentation. In *Computer Vision–ECCV 2006*. Springer, 2006, pp. 84–97. 2

[HZ03] Hartley R., Zisserman A.: *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5

[IB12] Isack H., Boykov Y.: Energy-based geometric multi-model fitting. *International journal of computer vision 97*, 2 (2012), 123–147. 2, 6, 11

[JP11] Jancosek M., Pajdla T.: Multi-view reconstruction preserving weakly-supported surfaces. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 3121–3128. 8

[JPS14] Joo H., Park H. S., Sheikh Y.: Map visibility estimation for large-scale dynamic 3d reconstruction. 2

[Low04] Lowe D. G.: Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*, 2 (2004), 91–110. 3, 7

[MGPG04] Mitra N. J., Gelfand N., Pottmann H., Guibas L.: Registration of point cloud data from a geometric optimization perspective. In *Symposium on Geometry Processing* (2004), pp. 23–31. 8

[MY09] Morel J.-M., Yu G.: Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences 2*, 2 (2009), 438–469. 3, 7, 10

[OSVG10] Ozden K. E., Schindler K., Van Gool L.: Multibody structure-from-motion in practice. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32*, 6 (2010), 1134–1141. 2

[RRGA12] Roussos A., Russell C., Garg R., Agapito L.: Dense multibody motion estimation and reconstruction from a handheld camera. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on* (2012), IEEE, pp. 31–40. 3

[RTVM10] Rao S., Tron R., Vidal R., Ma Y.: Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32*, 10 (2010), 1832–1845. 2

[RWFH12] Rothermel M., Wenzel K., Fritsch D., Haala N.: Sure: Photogrammetric surface reconstruction from imagery. In *Proceedings LC3D Workshop, Berlin* (2012). 8

[SFG14] Simon Fuhrmann F. L., Goesele M.: MVE - A Multi-View Reconstruction Environment. In *Proceedings of the Eurographics Workshop on Graphics and Cultural Heritage (GCH)* (2014). 8

[SSW08] Schindler K., Suter D., Wang H.: A model-selection framework for multibody structure-and-motion of image sequences. *International Journal of Computer Vision 79*, 2 (2008), 159–177. 2

[TK92] Tomasi C., Kanade T.: Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision 9*, 2 (1992), 137–154. 2

[TV07] Tron R., Vidal R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8. 2, 11

[VF08] Vedaldi A., Fulkerson B.: VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 7

[VH04] Vidal R., Hartley R.: Motion segmentation with missing data using powerfactorization and gpca. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (2004), vol. 2, IEEE, pp. II–310. 2

[Wu11] Wu C.: Visualsfm: A visual structure from motion system. *URL: http://homes. cs. washington. edu/~ ccwu/vsfm 9* (2011). 2, 8

[YRM06] Yang A. Y., Rao S. R., Ma Y.: Robust statistical estimation and segmentation of multiple subspaces. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on* (2006), IEEE, pp. 99–99. 2