Computer Vision and Image Understanding 117 (2013) 1575–1588

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Multi-object reconstruction from dynamic scenes: An object-centered approach

Young Min Shin, Minsu Cho, Kyoung Mu Lee *

Department of Electrical and Computer Engineering, ASRI, Seoul National University, Seoul 151-742, Republic of Korea

ARTICLE INFO

Article history:
Received 3 May 2012
Accepted 17 June 2013
Available online 6 July 2013

Keywords:
3D reconstruction
Dynamic scenes
Co-segmentation
Multiple objects

ABSTRACT

In this paper, we present a new framework for three-dimensional (3D) reconstruction of multiple rigid objects from dynamic scenes. Conventional 3D reconstruction from multiple views is applicable to static scenes, in which the configuration of objects is fixed while the images are taken. In our framework, we aim to reconstruct the 3D models of multiple objects in a more general setting where the configuration of the objects varies among views. We solve this problem by object-centered decomposition of the dynamic scenes using unsupervised co-recognition approach. Unlike conventional motion segmentation algorithms that require small motion assumption between consecutive views, co-recognition method provides reliable accurate correspondences of a same object among unordered and wide-baseline views. In order to segment each object region, we benefit from the 3D sparse points obtained from the structure-from-motion. These points are reliable and serve as automatic seed points for a seeded-segmentation algorithm. Experiments on various real challenging image sequences demonstrate the effectiveness of our approach, especially in the presence of abrupt independent motions of objects.

1. Introduction

Multiple view geometry is an important research area in computer vision. Traditionally, multiple view-based three-dimensional (3D) reconstruction systems are restricted to static scenes, in which the scene configuration is unchanged as the images are taken in different views. In this paper, we aim to develop a reconstruction system that is capable of building 3D models of multiple rigid objects in dynamic scenes where each object and camera are moving freely.

Suppose there are multiple target objects appearing simultaneously in the input images, and they are settled in different configurations in each scene. For example, several independent objects are captured in the multiple images, and the position and pose of these objects vary from image to image. In such settings, the reconstruction problem from the dynamic scene raises various challenges. Since there are arbitrary motions of multiple target objects in the scene, the multi-view constraint of the *whole* scene is not satisfied. Given that the 3D reconstruction applied to the whole images is not established due to the lack of global geometric consistency, the geometric conflict among the target objects has to be resolved prior to reconstruction. Thus, a natural way is to separate each object from the others and reconstruct each one independently. Once identifying and segmenting corresponding regions of same object among images and establishing feature correspondences among the regions, the multi-body problem can be easily resolved by finding relative camera poses for each object region individually.

To solve the multi-body problem, we approach from the object recognition viewpoint. If we recognize each independently moving object in the image sequences, the solution gives us answers to the following concerns: the number of the objects, the membership of the objects, and their correspondences. These data serve as clues in reconstructing the structure of individual objects in the dynamic scene environment [1]. In the present work, we apply object recognition technique to decompose the dynamic scene into coherent regions that correspond to separated objects distinguished by color or texture characteristics.

The key contributions of our approach are summarized as follows:

1. We designed a 3D reconstruction system for multiple objects in dynamic scenes. It is the first solution to the 3D reconstruction problem of dynamic abrupt scenes with arbitrary view points. The solution to such problem has never been proposed in previous works. Our method is general in that it does not have any restriction on the number of objects or the ordering of sequences, yet yielding compelling results.
2. Extended co-recognition technique has been utilized for identifying objects in multiple images. Co-recognition avoids feature

* Corresponding author. Fax: +82 2 878 1452.
E-mail addresses: shinyoungmin@gmail.com (Y.M. Shin), chominsu@gmail.com (M. Cho), kyoungmu@snu.ac.kr (K.M. Lee).

1077-3142/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2013.06.008

tracking issues, one of the most troublesome problems in this area, when motion segmentation is applied to real images. Without feature tracking, our method can handle the general settings of objects between images.

3. We have refined the patch-level object boundary obtained from object recognition algorithm up to the pixel-level precision. We applied an interactive image segmentation method in a non-interactive way by providing seed points automatically. The 3D structure of the target object is utilized to get reliable object segment.

The structure of this paper is as follows: in Section 2, we start with a brief review of related works; Section 3 describes our approach and the detail of the proposed system; we present the experimental results on various data in Section 4; and the paper concludes in Section 5.

2. Related work

Only few studies have focused directly on this subject, and the works of Rothganger et al. [2] and Ozden et al. [3] are the closest ones to our system. Rothganger et al. presented a 3D structure representation using a collection of small planar patches combined with their normalized local appearance description. They segmented a scene into rigid components and constructed 3D models of the components up to the local affine patch level. The major objects are distinguished one after another using the RANSAC (RANdom SAmple Consensus) procedure on the feature tracks [4]. They applied their representation in detecting moving objects in video sequences. The work of Ozden et al. [3], meanwhile, is an online method to cope with motion segmentation and reconstruction simultaneously. They attempted to estimate the number of moving objects by splitting and merging feature tracks. They focused on practical considerations to build a reconstruction system, and their method can handle more realistic scenes.

Both [2,3] rely heavily on the feature tracker. Since Lucas and Kanade have introduced the KLT tracker based on optical flow [5], it has been used as a popular element for approaches to motion segmentation including methods for non-rigid object motions [6,7]. In general, however, the feature tracker is prone to failure when the inter-frame motion is large. Given that the tracker assumes strong continuity between frames, one missing track can lead feature trajectories to drift away and make their return very difficult. Moreover, if the input data is given as an unordered set of images we cannot apply tracking-based methods to match the images.

Another issue of multiple object reconstruction is the relative scale between objects. If each object is treated separately, each 3D model may have a different scale factor because reconstruction is determined up to the unknown scale. This subtle problem is known as relative scale ambiguity problem, and has no exact solution; however a previous work [8] has introduced an estimation method based on generic motion constraints. In this method, the objects are re-arranged by their determined scale to create a whole 3D scene configuration.

Apart from the above, the topic of motion segmentation has to be addressed when dealing with the multi-body problem. Model selection and subspace separation are common concepts in this area. Wang and Adelson [9] presented the idea of assigning pixels to overlapping layers, where each layer's motion is described by a smooth flow field. This has been referred to as a layered approach, which explains pixel motions as a parametric motion model featuring several layers. Such representation has been adopted by a number of algorithms. They have often used expectation–maximization (EM) [10,11] or graph cuts [12,13] to minimize energy functions.

Some works [14–16] approached the multiple motion segmentation problem as mathematical multi-body factorization. They are based on the subspace constraints that the trajectories of points on the independent rigid objects are from independent subspaces. The factorization method is primarily focused on the algebraic explanation to the problem; the authors assumed that the feature-tracking issue has been solved, in fact, this is an unrealistic assumption. Most studies on motion segmentation tend to concentrate on theoretic aspects and have derived solutions from simple data with restricted environments.

In response to the large disparity discrete motion problem, Wills et al. [12] presented a method that combined the layer-based approach concept and feature-based motion estimation. First, the initial correspondences matched by comparing the descriptor vectors of interest points are computed. Established initial matches are perturbed to check correctness and boost the inlier matches. They then used a RANSAC-based procedure to detect and partition the motion fields of the frames. Finally, an approximate graph cut method is applied to assign pixels densely to each motion field. Their proposed approach demonstrated the ability to handle large inter-frame motion, which is the limitation of the optical flow-based feature tracking methods. The work of Wills et al. is similar to our method in that it can handle large abrupt motions and boost true inlier interest point matches. However, their work was based on a strong assumption that objects are matched by a single planar homography between images. This assumption does not hold for most general 3D objects, particularly when images are taken under a wide baseline setting.

Despite previous attempts to attain feature-level motion segmentation based on tracking method or theoretic factorization, the dynamic multi-object reconstruction problem has never been directly investigated. As a consequence, there is no practically available solution to this challenging problem. Thus, we emphasize that our work has made a substantial contribution in the form of a novel integrated system based on the object-centered approach (see Fig. 1).

3. Proposed method

In this section, we introduce our reconstruction system, which uses an object recognition-based matching and segmentation. The main goal of a dynamic scene reconstruction system is to search for all major independent objects in multiple images and build 3D models of each object by gathering all visual information extracted from the images. Our approach stands on a number of existing techniques to achieve this goal. The proposed method consists of four major stages. The starting point of our algorithm is the automatic recognition of common objects among multiple images using a co-recognition technique, in which the recognized objects with the same identity across images are clustered together. Each cluster is a collection of the same object regions, roughly segmented in all images with matched feature correspondences. Following the recognition and the clustering steps, we applied structure from motion (SfM) to each object to calibrate the virtual camera. The SfM yields sparse 3D points on the object surface as well as the corresponding camera matrices through the bundle adjustment optimization. Then, the roughly segmented result of co-recognition and sparse 3D points projected on the images are used as seeds for the RWR (Random Walk with Restart) segmentation algorithm. Finally, we used patch based multi-view reconstruction algorithm to build the 3D models of each object. An overview of our system is illustrated in Fig. 2.

3.1. Recognition

The first step of our algorithm is object recognition across images. Given that we have no object-level prior knowledge of

(a) Input images contain two house objects with arbitrary independent poses.

(b) 3D reconstruction results of both *house* objects.

Fig. 1. An example of multiple object reconstruction from a dynamic scene. Despite the arbitrary positions and poses of the objects in the images, each object is separated and reconstructed to the individual 3D model.

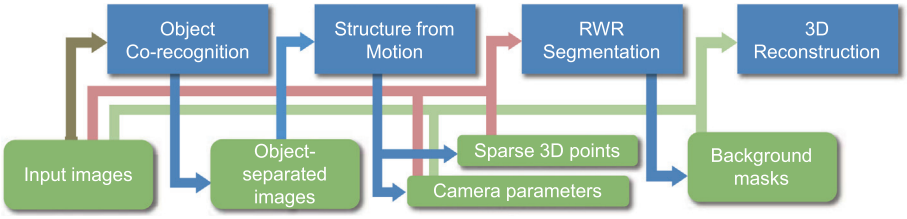


Fig. 2. Overview of the proposed system.

the scene, we start by determining the number and locations of the objects in images. Given a set of images, as shown in Fig. 4, multiple objects appear simultaneously in a view, while their poses and arrangements vary in every view. To reconstruct multiple objects in the dynamic scene, each object should be segregated individually. By segmenting each object regions and finding the association among those in multiple images with feature correspondences, we can apply SfM algorithm on each separated object to construct its 3D model.

3.1.1. Co-recognition

The object recognition problem for multiple object reconstruction is different from the exemplar-test object recognition framework. In our problem, there is no explicit distinction between exemplar and test images. Instead, we only have input images containing multiple dynamic objects. Therefore, the problem is to find and localize common objects in the images. The essence of our object recognition routine is based on co-recognition [17]. Co-recognition is an image matching method, which establishes correspondence among multiple common objects in image pairs without prior knowledge of the objects.

The building block of co-recognition is pair-wise image matching. Thus, we divided the matching problem of multiple images into sub-problems. Suppose we are given a set of N images, then we have a total of $N(N-1)/2$ image pair combinations. We solve each pair-wise sub-problem and get the final solution by integrating the results of those sub-problems.

Generative model:

We proceed by describing the generative model formulation. Given the image pair (I_i, I_j), the co-recognition problem can be modeled as a maximum a posteriori (MAP) estimation of the parameter θ on the basis of the image pair observation. According to the Bayesian formulation, the solution is found by maximizing posterior probability that is decomposed into likelihood and prior terms as follows.

$$\theta^* = \arg\max_\theta p(\theta|I_i, I_j) = \arg\max_\theta p(I_i, I_j|\theta)p(\theta). \quad (1)$$

We define θ as a set of K matching clusters μ between an image pair expressed as:

$$\theta = \{\mu_1, \mu_2, \ldots, \mu_K\}. \quad (2)$$

Each matching cluster is a set of local patch matches between images given by:

$$\mu_k = \{\lambda_{k:1}, \lambda_{k:2}, \ldots, \lambda_{k:L_k}\}, \quad (3)$$

where L_k denotes the number of local region matches in μ_k. Thus, θ is a set of local correspondences grouped into the object-level matching.

The prior $p(\theta)$ represents the geometric properties that true common object matches should obey. It constrains the position

of the matched local patches in the image pair. The relative position of the local patches in each cluster must have similar arrangement in both images. We penalized the position discrepancy between corresponding patches in the image pair. The geometric discrepancy error of cluster μ_k is the sum of local deformation cost of each local match, formulated as

$$\mathrm{E_g}(\mu_k) = \sum_{i=1}^{L_k} \mathrm{d_g}(\lambda_{k;i}). \tag{4}$$

The local deformation cost $\mathrm{d_g}(\lambda_{k;i})$ is the average geometric distances between the center position of neighboring patches of $\lambda_{k;i}$ and their matches in the other image. The neighbor relation is determined by the Delaunay triangulation of the patch centers, and the distance is measured in the normalized domain, in which the elliptical patch is transformed to the unit circle shape.

It also encodes the preference for larger clusters because reliable common objects are expected to have strong support from many local patches,

$$\mathrm{E_m}(\theta) = \sum_{k=1}^{K} (-L_k - |\Delta_k|), \tag{5}$$

where $|\Delta_k|$ denotes the number of Delaunay triangles of the cluster μ_k.

The likelihood $p(I_i, I_j|\theta)$ reflects the photometric similarity between matched patches in images. The normalized cross-correlation (NCC) values of the matched image patches are accumulated to measure the photometric error,

$$\mathrm{E_p}(\mu_k) = \sum_{i=1}^{L_k} (1 - \mathrm{NCC}(\lambda_{k;i}))^2. \tag{6}$$

To sum up, we finally get the posterior probability from the prior and the likelihood with balancing parameter $\beta_\mathrm{p} = 3$ as

$$p(\theta|I_i, I_j) \propto \exp\left(-\sum_{k=1}^{K} \mathrm{E_g}(\mu_k) - \mathrm{E_m}(\theta) - \beta_\mathrm{p} \sum_{k=1}^{K} \mathrm{E_p}(\mu_k)\right). \tag{7}$$

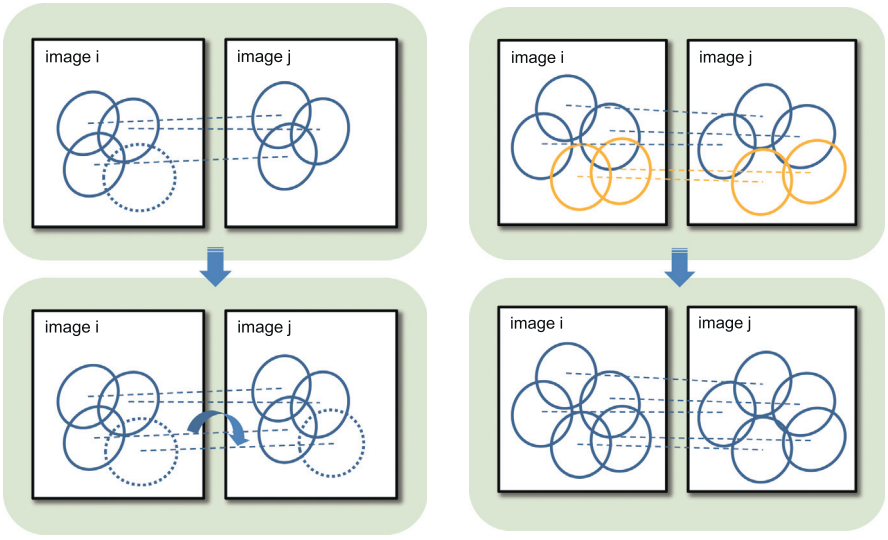
Inference:

The first step of co-recognition is establishing initial feature-level matches between extracted local feature points. The feature detectors [18,19] extract affine invariant regions from the images. Features with a distance of less than 0.45 in the SIFT descriptor space are considered to be matched. Usually, there are many false matches among initial matches.

After the initialization step, each feature match forms an initial cluster and grows to a larger cluster. Each initial cluster has its own expansion layer consisting of a set of overlapping circular grid, which covers the image. The overlapping circular grid on the image is the basic unit of growth. Then, we begin to run two iterative moves (expansion/merge) to grow the initial clusters.

In an expansion move, propagation and refinement operations are performed. The algorithm makes a proposal to propagate one of the current established matches to one of the unoccupied regions of the expansion layer. Then the new match is refined by local search around the proposed region to find the best matching region. In a merge move, two large clusters are selected and merged into one. Also their expansion layers are combined. Fig. 3 shows conceptual illustrations explaining the notion of expansion and merge moves. Expansion and merge proposals are accepted when the proposed state yields improved posterior $p(\theta|I_i, I_j)$. Expansion moves encourage merge moves to find congruous clusters by enlarging them. Likewise, merge moves help plausible expansion moves to have more expansion opportunities through gradual integration of compatible clusters. Utilizing cooperative expansion/merge moves, our greedy algorithm explores the solution space iteratively. Iterative growing is then performed until the convergence of posterior probability $p(\theta|I_i, I_j)$. After convergence, we eliminate unreliable clusters from θ^*. We measure the reliability of the cluster as the expanded area of the region, because larger clusters are more likely to originate from reliable seed matches.

Clearly, co-recognition has advantages over the feature tracking-based methods in terms of object identification. First, unlike the feature tracker, co-recognition-based approach can handle sudden object viewpoint changes between images efficiently. Given that inference starts from initial local feature matching, objects appearing at arbitrary position of images are recognized regardless of motion continuity. Although [2,3] can also reconstruct 3D model of multiple objects, they require smoothly captured video data, since they are based on the feature tracker. Besides, there are some prior works [20–23] that perform reconstruction from unordered



(a) Expansion move (b) Merge move

Fig. 3. Expansion and merge moves. The concept of local patches and their correspondences between image pair (I_i, I_j) are expressed in elliptical regions and dashed lines connecting them. (a) In expansion move, current established local matches propagate an unoccupied region (dotted region) by transferring the transformation information from the nearby match. After local search to refine the propagated region, the new match is established and added to the cluster. (b) In merge move, two different clusters (depicted in dark blue and orange colors, respectively) merge into one cluster. This figure is best viewed in color.

images; however, they require the scenes to be static. Thus, we argue that we solved more challenging and generalized problem to deal with unordered set of images containing multiple dynamic objects. Second, due to expansion procedures of co-recognition, new matches are augmented from initial matches to cover object region. Therefore, the detected object regions are not restricted to the output of local feature detectors. Third, refinement presents flexibility to the expansion procedure and non-planar 3D objects are successfully recognized. It can be explained by the ability to overcome deformation coming from viewpoint variation.

3.1.2. Integration of the sub-results

As mentioned earlier, we divided the co-recognition problem on multiple images into sub-problems. The result of each sub-problem is a set of commonly appearing object regions matched in image pair. As depicted in Fig. 4, each object in one image can have several matching regions produced from pair-wise matches with different images in the dataset. Since each pair-wise matching is performed independently, there is no inter-connection of object identity between the results. Therefore, we combined the results of sub-problems into one integrated result. The integrated result has object-level correspondence network information.

The hierarchical agglomerative clustering [24] is used to unite pair-wise results. In the present work, we define the similarity measure of two object-level correspondences as the ratio of overlapping areas to the smaller region. Assuming that two matching clusters μ_p and μ_q, have a common sharing image, we let R_p and R_q be the region occupied by μ_p and μ_q on the common image, respectively. The distance between μ_p and μ_q is expressed as follows:

$$\mathrm{dist}(\mu_p, \mu_q) = \left(\frac{\min(\mathrm{Area}(R_p), \mathrm{Area}(R_q))}{\mathrm{Area}(R_p \cap R_q)} \right). \quad (8)$$

The distance is set to infinity when there is no common image between the matching clusters. The object correspondences are joined by single-linkage hierarchical clustering until the distance is larger than 1.25. Object correspondences with distance closer than 1.25 are gradually agglomerated in the bottom-up manner. Therefore, the integrated result consists of detected object regions categorized into set of identical objects in the images.

As explained in Section 3.1.1, the basic unit of multi-layer growing is based on [17]. Although [17] detects identical objects within and across the images, we made a variation on it from a practical application aspect. Our algorithm applies different matching scheme according to each different type of data. The matches are allowed to be established in one of three ways: across the images, within the images, or across the temporal order. This modification increases efficiency and makes the algorithm applicable to various input data. The modification detail according to the input data type is described in Section 4.

3.2. Camera calibration

The next stage in our system is camera calibration. After the recognition stage, all objects in the images were detected and clustered to the object correspondence network. Each set of object region segments across images satisfies the scene geometric consistency. They are equivalent to images of underlying object only taken by cameras in various positions and viewing angles, in which other distracting objects do not appear. Fig. 4(b) shows a typical example of object-centered segmented images. We perform camera calibration on each separated set of object regions depicted as blue and red outlines.

In this step, we compute the camera projection matrices of each individual object using the SfM technique. The point correspondences are optimized by the bundle adjustment optimization and

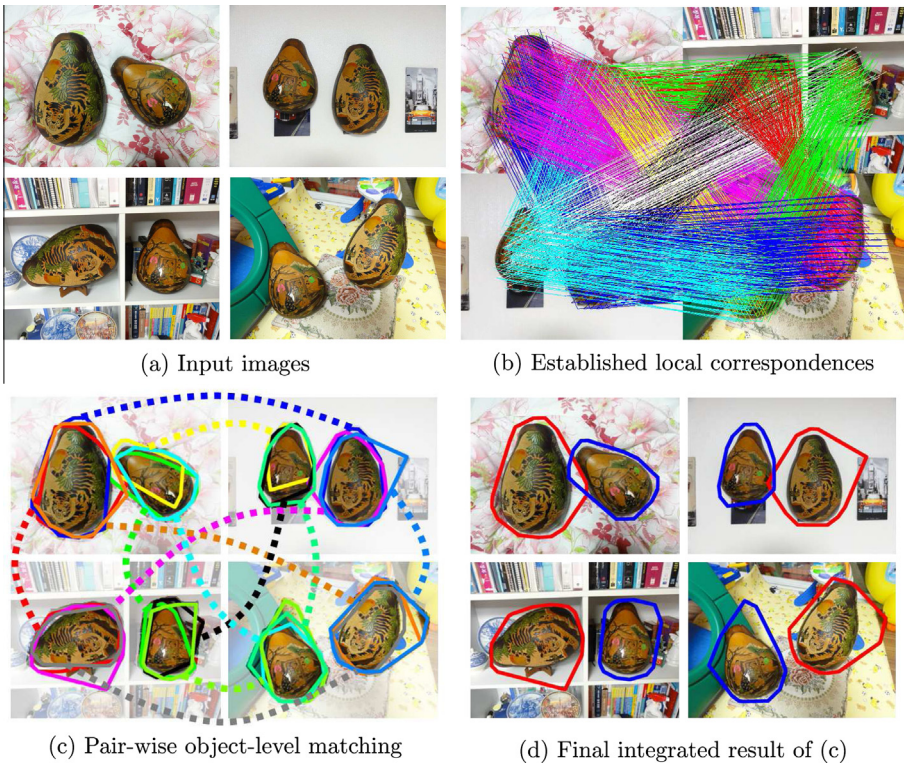


Fig. 4. Pair-wise co-recognition and integrated result are illustrated on the *Gourd* dataset. (a) Input images have two objects in four different backgrounds. (b) The local correspondences are established between the instances of same object. (c) The pair-wise object-level matching and their resulting regions are superimposed on the images in different colors. (d) The identity of each object is distinguished by integrating the results of pair-wise object-level matching. Each color (red/blue) identities each object.

the SfM yields both sparsely reconstructed set of 3D point coordinates **X** with camera matrices **P**. Assuming we are given K objects in N images, we can have following set of camera matrices and points:

$$\{\mathbf{P}_{nk}; n = 1, \ldots, N, k = 1, \ldots, K\} \quad (9)$$

$$\{\mathbf{X}_k; k = 1, \ldots, K\}. \quad (10)$$

Note that not all objects have to be visible in all images; thus, if an object is missing in some images, the corresponding camera matrices and 3D points are not available.

Co-recognition produces object boundary segmentations up to the overlapping circular grid. Some extra parts outside the objects are included in the object regions due to the expanding nature of co-recognition. However, these small noises hardly affect the performance of SfM. The SfM multi-view constraint easily prunes these noises.

3.3. Object boundary refinement

By virtue of object co-recognition, we perform object-centered camera calibration with segmented object images. Although the object boundary provided by co-recognition is useful in calibrating camera parameters, it is still rough and inappropriate for accurate 3D reconstruction of an object shape. It is clear that better object masks enhance 3D reconstruction results by preventing unnecessary background parts from being processed. In this section, we apply the image segmentation method to obtain a detailed object segmentation boundary. We adopt the seeded segmentation method proposed by Kim et al. [25]. It is a generative image segmentation algorithm based on the Random Walks with Restart (RWR), and can efficiently solve the weak boundary problem and texture problem.

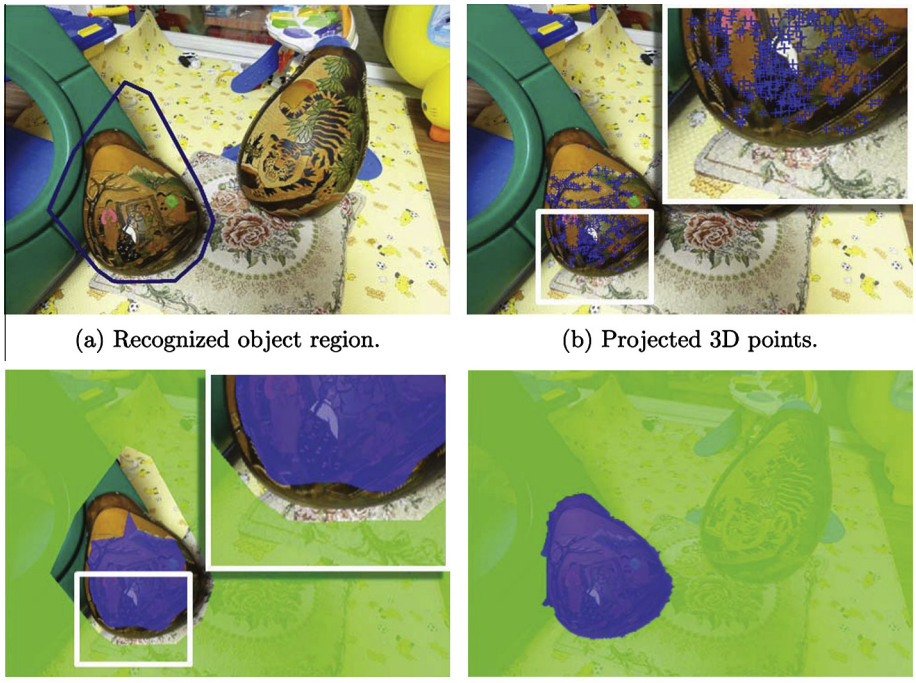
First, we construct a weighted graph in an image. The graph consists of pixel nodes and the edges connecting the neighborhood pixels. The edge weights encode image color similarity between connected nodes. Then, the random walkers traverse the graph with the probability proportional to the weights on the edges. We compute the steady-state probability for every pixel that a random walker starting at a seed point stays at the pixel. Finally, the most probable label is assigned to each pixel.

The RWR algorithm requires initial seed points for segmentation, and for this, the user provides scribbles as starting pixels of each label's random walker on the weighted graph. For binary labeling between object and background, seeds on the target object and background are required. Unlike interactive segmentation method [25], we aim to generate seed points automatically using RWR as a non-interactive segmentation method.

The key idea behind providing reliable seed points is utilizing an object's geometric information. Although discontinuity of visual pattern is typically observed at the object boundary, sometimes it is ambiguous to decide whether a pixel is on the object or not by the photometric observation only. As explained in Section 3.2, the sparse 3D points as well as the camera matrices are extracted by SfM under the consideration of 3D geometry. This implies that projected locations of sparse 3D points on an image plane are most likely to lie on the object's surface. The sparse 3D points $\mathbf{X}_k$ of object k are projected on the image n by the projection matrix $\mathbf{P}_{nk}$ as follows:

$$\{\mathbf{x}_{nk} = \mathbf{P}_{nk}\mathbf{X}_k; n = 1, \ldots, N, k = 1, \ldots, K\}. \quad (11)$$

In addition to the projected points $\mathbf{x}_{nk}$, we apply 2D alpha shapes [26] to them to obtain more stable seeds for segmentation. The alpha shape is a polygon derived from the point set with parameter α controlling the desired level of detail. The alpha shape generated from $\mathbf{x}_{nk}$ fills the empty space between the projected points. The seed is now given as a polygon area instead of each projected point. We empirically determined the optimal α value to adapt to the scale and texture density of the objects by following



(a) Recognized object region. (b) Projected 3D points.

(c) Object and background seeds. (d) Refined object boundary.

Fig. 5. Object boundary refinement by RWR segmentation. (a) The object boundary provided by co-recognition is marked in blue line. (b) Blue crosses deNote 3D points projected on the image. White box is scaled for display. (c) Blue mask represents *object* seeds produced by alpha shape, and green mask denotes *background* seeds given by object recognition. (d) Refined object region after proposed non-interactive RWR segmentation.

procedure. We select the 6 nearest neighbors of every points and calculate the average distance from the selected points. The α value is set to twice the average distance.

For the background seeds, we simply mark all the points outside the object boundary obtained from co-recognition. Fig. 5 shows the segmentation process with the automatically generated seeds. The object boundaries are determined to pixel-level precision through the RWR segmentation stage.

3.4. 3D reconstruction

The final stage of our system aims to reconstruct a 3D model for each object. Given object segmentation masks and dense correspondences with camera projection matrices, the condition for running 3D reconstruction algorithm is satisfied. One can use any reconstruction method to obtain 3D models of objects and background. In the paper, we adopt the publicly available multi-view stereo software PMVS [27], considered as the state-of-the-art algorithm.

4. Experiments and results

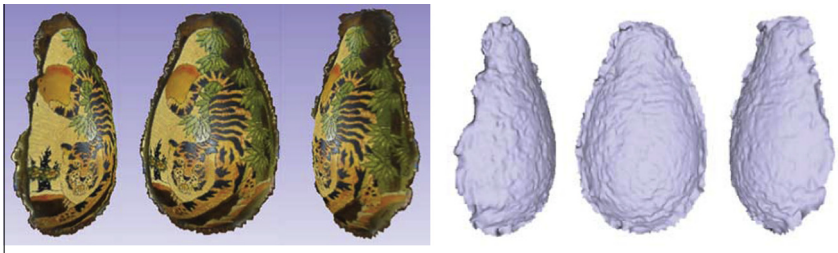
In this section, we demonstrate the experimental results of our approach for multiple object reconstruction in dynamic scenes. We demonstrated the performance of our algorithm on several test image sets containing objects that exhibited varying geometric configuration across frames, on both different and same backgrounds. For the experiments on video data, video sequences captured from movies and video clips downloaded from the Internet are used, as well as the video taken in the Lab. Comparisons are drawn with some prior works that have similar goals with our approach. We also performed quantitative evaluations of pixel-wise segmentation accuracy and 3D reconstruction correctness. To show the process and result of our approach more effectively, we uploaded the supplementary video material on our web site: http://cv.snu.ac.kr/research/MORDS/video_MORDS.wmv.

4.1. Qualitative results

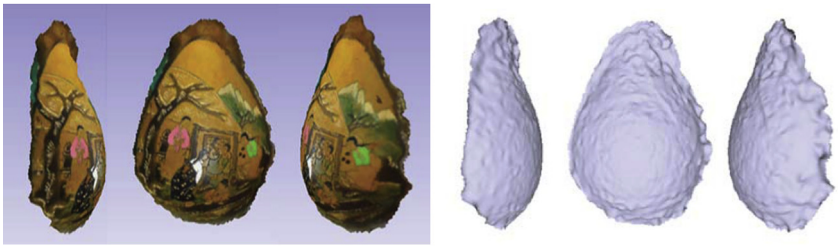
Dynamic scenes with different backgrounds: We first performed experiments on the sets of images, which capture multiple target objects on different backgrounds in each shot. We took the *Gourd* and *Tea* dataset which are comprised of 4 images as shown in Figs. 4 and 7(a), respectively. Our goal is to reconstruct the common objects which appear in all images. The scene continuity between consecutive frames is a crucial assumption that the tracking-based methods rely on. However, in this setting of experiment, every image has its unique configuration of scenes. Any permutation of input images will yield abrupt change of object poses and positions. The target objects are shown in various poses and positions in the scenes. Given that they have no consistent background, our algorithm utilized visual information from the foreground objects only. Figs. 6 and 7(b) show the reconstruction results of *Gourd* and *Tea*, respectively. Considering that the target objects occupy a small part of the images and only 4 images are used to reconstruct the 3D models, the effectiveness of our system is quite convincing.

Dynamic scenes with constant background: The second experiment is designed for reconstructing the foreground objects and background parts. For this purpose, we captured *Houses* dataset. Fig. 8(a) shows some sample images from the *Houses* dataset. *Houses* is a sequence of 25 images. Two objects have independent abrupt motions with a consistent background, while the camera moves left and right. Since the images contain consistent background, our algorithm separates the foreground and background by detecting them as distinct objects. The 3D model of the background part is also reconstructed as well as the foreground objects. Fig. 8(b) and (c) shows the reconstructed background and full 3D shape of each object, respectively. The explicit segregation of the object and background has enabled reconstruction of occluded background. Due to the occluding objects, some part of the background is not seen from the camera's view, however, images taken from other viewpoints compensate for the missing part. Note that this is different from the crowded scene reconstruction presented previously [27,28], which treat occluding objects as obstacles that have to be filtered out.

Identical objects in one image: Interestingly, the proposed method is applicable to the reconstruction from a single image if the single image contains multiple shots of identical objects. Fig. 9 shows our example of *Milk*. The repeated visual pattern induced by the multiple instances of identical object is frequently observable in the real world. Each object region in the image is



(a) *Gourd* object 1



(b) *Gourd* object 2

Fig. 6. Reconstruction results of two objects of *Gourd* dataset.

(a) Input images contain three objects in different configuration and background.

(b) Reconstruction results of three objects.

Fig. 7. *Tea* dataset and the reconstruction results.

equivalent to each shot of same object taken from different viewpoint. The relative camera position varies as the objects have different poses seen by the single camera.

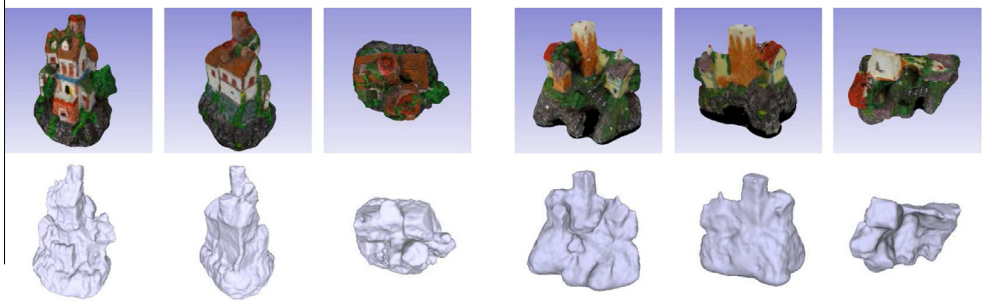
Here, we decompose each instances of the object in the recognition stage, and they are treated as multiple shots of same object. In such a case, the image matching is done only within the single image itself. To perform the single image reconstruction, we carry out a little modification to the initialization step of co-recognition. We allow the local features to find initial correspondences from the feature points extracted from the same image. The self-matched regions grow to all of the identical object regions. As shown in the reconstruction result in Fig. 9, our method provides good reconstruction result from a single image.

Dynamic scene in video clips: In the fourth experiment, we performed experiments on the *Racing* and *Dolls* video. We captured the *Racing* video from YouTube, and the *Dolls* was taken in our Lab.

In the *Racing* video clip, the camera is fixed and the viewpoint does not change. Instead, two cars appear and disappear in the sequence as they move across the circuit. This video scene contains consistent background, but it does not have relative camera motion. Although our system runs without image ordering, we exploit the ordering information of video frames to increase efficiency and overcome the high computational complexity of matching all possible pairs from the combinations. The frame at time t is matched with the frame at time $t+3$ and $t+6$ sequentially. A total of 36 frames were used in the experiment. The reconstruction result is shown in Fig. 10. Despite the blurry low texture of the body and window glass, the two cars are separated and modeled to 3D shapes successfully.

In the *Dolls* video clip, two dolls revolve around each other in the background. This scene is captured by a moving camera. The two objects occlude each other when one is located between the other and the background, then the occluded object reappears as the revolution continues. We observe partial/full occlusions of the target objects in the video. This video raises several challenging issues such as scale change, treatment of consistent but occluded background, mutual occlusion of objects, and re-identification of disappeared object. We apply the same strategy used in *Racing* video to match the sequences. A total of 160 frames are used in the experiment. Fig. 11 shows the reconstruction results of two doll

(a) Sample images from 25 input images of *Houses*.



(b) Reconstructed 3D model of each object.

(c) Reconstructed background

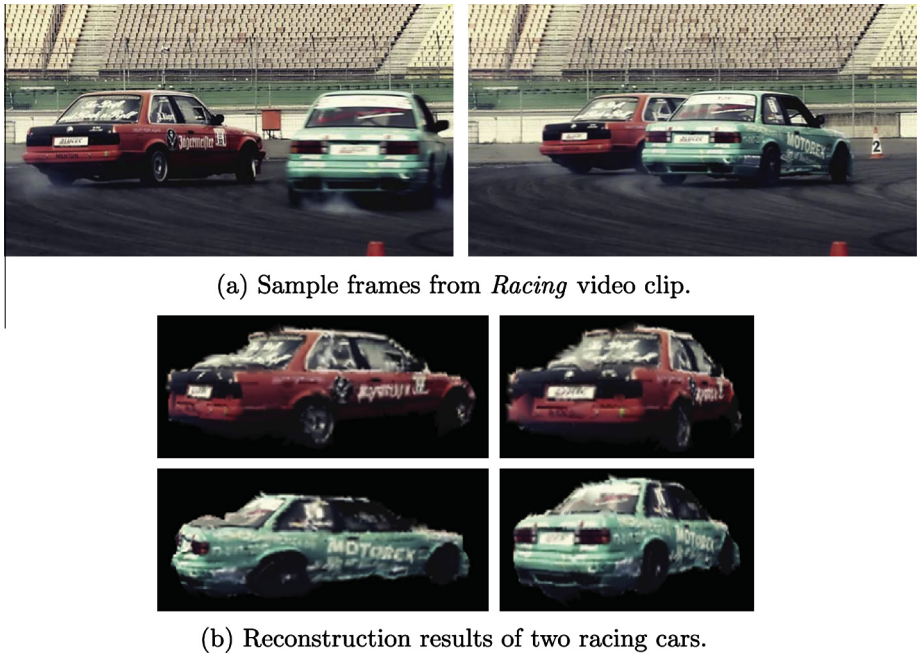
Fig. 8. *Houses* dataset and reconstruction result of two objects and background.

(a) Input image of *Milk*.

(b) Detected object instances as sets of local regions.

(c) Reconstructed 3D model.

Fig. 9. Reconstruction of identical object from a single image.



(a) Sample frames from *Racing* video clip.

(b) Reconstruction results of two racing cars.

Fig. 10. *Race* video clip and the reconstruction results of two cars. Note that the consistent background part is not reconstructed since the camera is fixed.

objects and the background. The results reveal that proposed method successfully overcomes the aforementioned challenges, which are known to be difficult issues of conventional methods.

Comparison with [2]: We performed an experiment on the same video in [2]. A scene where a van moves on the road as the camera pans right was extracted from the movie *Groundhog day*. A total of 30 frames were used in our experiment. Rothganger et al. [2] modeled the object as a set of affine covariant surface patches extracted from the feature detectors. As their algorithm runs only on the given patches, the reconstruction result is limited to the sparsely and unevenly distributed interest regions induced by the detectors. However, our algorithm explores new regions, which were not included in the initial output of detectors, through the expansion process. The results of our method and [2] are displayed in Fig. 12 for comparison.

Comparison with [12]: To contrast robustness with abrupt motion, we compared the object detection performance of our method with the work of Wills et al. [12]. Their method was intended to overcome the large inter-frame motion disparity, which was the biggest problem of tracking-based motion segmentation algorithms. We selected an image pair from *Tea* and applied both methods on the same image pair.[1] The object boundary determined by our method is displayed in Fig. 7(b) and (a) reports the failure of [12]. We got similar results from any combination of image pair. Although the method in [12] has been developed to handle large abrupt motions, their approach is weak in terms of scale change and rotational transformation. They have not explicitly modeled local affine transformation between the true inlier matches. Moreover, their assumption of planar motion is not appropriate for 3D objects. When a 3D object has out-of-plane rotation, each of its local region undergoes different movement. Our algorithm adapts to the deformation of surfaces as well as arbitrary positioning of objects.

[1] We used the code provided by the authors (http://joshwills.com/projects/www_code.html).

4.2. Quantitative results

4.2.1. Evaluation of segmentation accuracy

As a matter of quantitative evaluation, we measured the segmentation accuracy of target objects after each step of co-segmentation, RWR, and 3D reconstruction. For co-recognition and RWR, detected regions of the objects were compared with ground truth. For 3D reconstruction step, we then re-projected 3D models on 2D image planes to obtain the segment.

To measure segmentation performance, we manually labeled the target object region's ground truth pixels. We measured the segmentation accuracy by three criteria. The hit ratio was calculated as the ratio of truly detected pixels to the ground truth pixels, $HitRatio = |Result \cap GT|/|GT|$. The background ratio refers to the ratio of false positive pixels to result pixels, $BkgRatio = |Result - GT|/|Result|$. The overlap ratio measures the degree of overall correctness of segmentation, as the ratio between intersection and union of the result and ground truth, $OverlapRatio = |Result \cap GT|/|Result \cup GT|$. The higher hit, overlap ratio and lower background ratio means we have obtained better segmentation results.

We took the measurements of the *Gourd, Tea,* and *Milk* dataset, which consist of image shots. Their results are shown in Tables 1–3, respectively. As shown in the Tables, the RWR segmentation step has significantly increased the segmentation accuracy. The tendency has shown that co-recognition detects relatively larger regions than the target object. This result explains that co-recognition detects each object region as a cluster of overlapping circular grid, which has expanding properties, while RWR finds pixel-level object boundary segmentation. The expanding nature of co-recognition yields high hit ratio, but also increases background ratio. The RWR mostly refined the object boundary by filtering out the background part. The RWR segmentation decreased the background ratio with little degradation of the hit ratio. Interestingly, the 3D reconstruction step sometimes shows slight decrease in the measured segmentation accuracy in terms of intersection ratio: 0.94–0.93 for *Gourd* and, 0.97–0.95 for *Milk*. The reason for this is that 3D reconstruction uses relatively conservative criteria to model objects. For the reliability of 3D model, part

(a) Sample frames from *Dolls*.

(b) Reconstruction results of two dolls.

(c) Reconstruction result of background part.

(d) One object is partially occluded by the other object. We can estimate the occluded part of the object by utilizing the built model. Left: input frame. Right: 3D model of partially occluded object superimposed on the image.

Fig. 11. *Dolls* video clip and reconstruction results.

of an object is reconstructed when it can be seen at least three different views. The small degradation of re-projected 3D reconstructed model is easily explained by the fact that ground truth segmentation is made solely along the object boundary observed in each view.

4.2.2. Evaluation of reconstruction accuracy

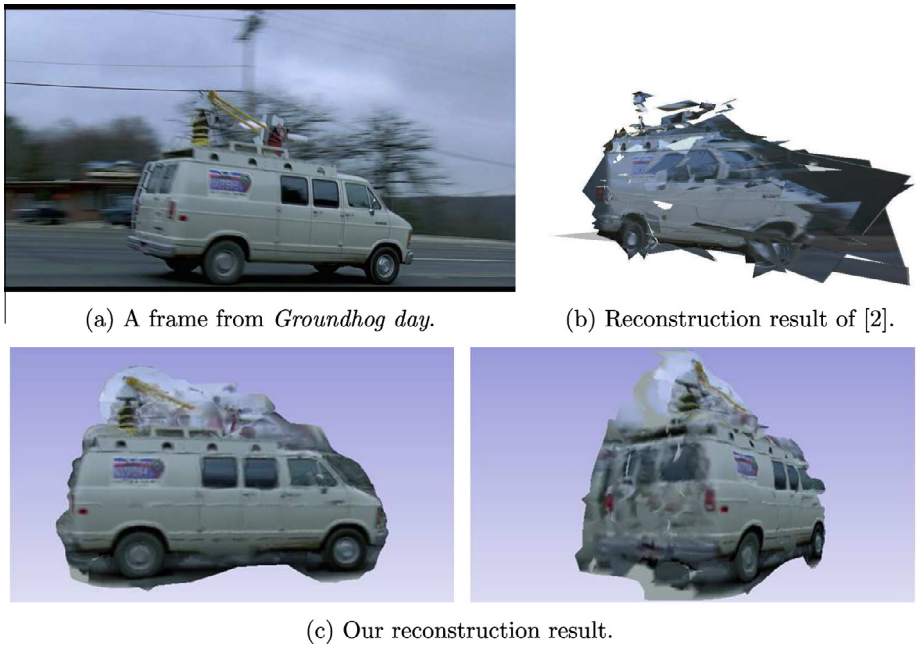
It is difficult to evaluate the reconstruction accuracy without the absolute, dense ground truth of the 3D object surface. However, some knowledge on the target scene's geometric relationships can be used to measure the accuracy of the built 3D model indirectly [3] (see Fig. 13).

In our experiment, we used the measured angle between two perpendicular planes of an object as the reconstruction accuracy measure. As shown in Fig. 14, we took two different sides of the *Milk* object and fitted a plane on each of its surfaces in a total least square sense. The principle component analysis (PCA) was used to fit the plane models to the 3D coordinates of the points on each reconstructed planes. Assuming that the points constitute a plane, the coefficients of the 3rd principle component define the normal vector of the plane by the underlying theory of the PCA. The deviation of the angle between the normal vectors of the two reconstructed planes shown in Fig. 14 from 90° was measured as 4.92°. This shows fair accuracy in spite of each object instance's small occupancy in the single image and limited viewpoints.

4.3. Analysis

Occlusion: A feature of the proposed algorithm is robustness against object occlusion. Conventional approaches [2,3] are based on the feature tracking method where the algorithm requires explicit treatment of occluded objects. Given that the tracked features show discontinuity at the occlusion boundary, the disconnected features have to be saved and restored to deal with the occlusion. However, our system recognized the object identities instead of following them. Thus, occlusion handling does not require additional explicit process. Furthermore, our recognize-integrate scheme enables us to deal with scenes containing full occlusion or missing object. Disappeared objects can maintain the same identities only if they keep similar appearance.

The *Dolls* video was intended to capture the severe occlusion situation. When an object passes behind the other object, the occluded object disappears completely from the camera's view. However, if an object is partially occluded as much as the calibration is



(a) A frame from *Groundhog day*. (b) Reconstruction result of [2].

(c) Our reconstruction result.

Fig. 12. Experimental results on a scene of the movie *Groundhog day*. Our approach shows better result then [2].

Table 1
Segmentation performance (*Gourd*).

	Object 1			Object 2			Total		
	Hit	Bkg	Overlap	Hit	Bkg	Overlap	Hit	Bkg	Overlap
After co-recognition	.998	.309	.690	.994	.272	.727	.996	.295	.704
After RWR segmentation	.966	.017	.950	.965	.025	.941	.965	.020	.946
3D model reprojection	.961	.016	.946	.938	.034	.908	.952	.023	.931

Table 2
Segmentation performance (*Tea*).

	Object 1			Object 2			Object 3			Total		
	Hit	Bkg	Overlap	Hit	Bkg	Overlap	Hit	Bkg	Overlap	Hit	Bkg	Overlap
After Co-recognition	1.000	.725	.275	.998	.553	.446	.999	.460	.540	.999	.591	.409
After RWR segmentation	.990	.105	.887	.962	.123	.848	.965	.100	.871	.971	.107	.870
3D model reprojection	.979	.018	.962	.899	.023	.880	.900	.017	.886	.922	.019	.906

Table 3
Segmentation performance (*Milk*).

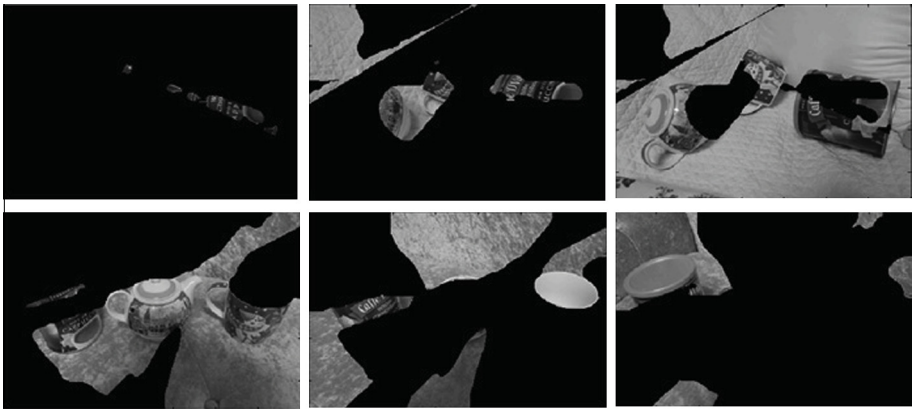
	Object 1		
	Hit	Bkg	Overlap
After co-recognition	.992	.318	.679
After RWR segmentation	.997	.070	.972
3D model reprojection	.972	.016	.957

available using the observable part, we can estimate the hidden part of the object by projecting the built object model. Fig. 11(d) shows an examples of occlusion.

Computational complexity: We performed the experiments on a computer with 3.3 GHz processor. The co-recognition and RWR are implemented in Matlab, while SfM and dense 3D reconstruction are written in C++.

In our framework, the majority of computational time was spent on the co-recognition and the dense 3D reconstruction steps. For instance, the co-recognition step took a total of 1380 s to match the *Gourd* images. The *Gourd* data consists of 4 images ($N = 4$), yielding 6 pairs of image matching sub-problems. On average, the running time of each pair-wise sub-problem was 230 s with standard deviation of 23.1 s. The camera calibration step took 23 and 22 s for the first and second objects, respectively, totally 45 s to run the SfM algorithm on both of them. The object boundary refinement step took less than 1 s for each image, consuming the smallest computation time in the whole pipeline. The dense 3D reconstruction step required 143 s to build the 3D model of the first object, 103 s to the second object.

The computation time of each co-recognition sub-problem varies according to the size of the foreground objects images. The algorithm converges relatively quickly in the case of small objects, since only small number of iterations are needed to find the object regions. On the other hand, larger objects require more time to be recovered by the iterative region growing process. For the *Race* dataset, more than 500 s were needed in each co-recognition sub-problem, since the region growing expanded to the whole image region. The parameters that control the reconstruction density mainly determine the execution time of the dense 3D reconstruction.



(a) Result of [12], where each column corresponds to each object.

(b) Our result, where each color represents identity of each object.

Fig. 13. Comparison of object detection performance with [12] reveals the superiority of our method in abrupt motion of 3D objects. Detected object boundary on an image pair of *Tea* is displayed.

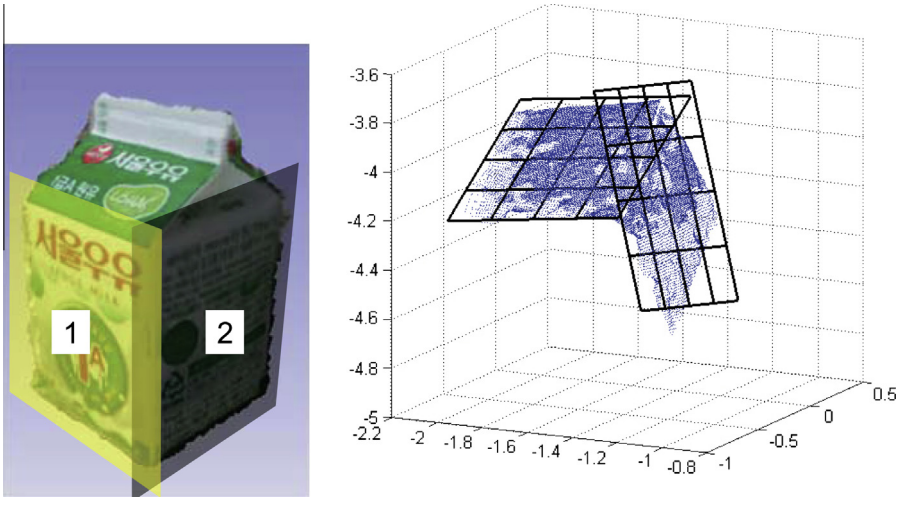


Fig. 14. Perpendicular planes in the object are used to measure the accuracy of the 3D reconstruction. Two sides of the *Milk* object are fitted on two planes and their relative angle is calculated using normal vectors.

Limitations: Although our system presents a robust framework against challenges such as occlusion (*House, Racing*, and *Dolls*), affine or perspective view changes (*Tea*), it also has limitations. Empirically, the structure-from-motion step is the weakest part of the flow. Objects with planar or shallow-depth structure can raise degeneracy to SfM. In such cases, despite the success of object recognition step, the objects cannot be reconstructed accurately. Next, the object recognition step requires sufficient texture on the surface of target objects since each local patch is matched by texture information on it. For instance, since the teapot object of *Tea* images contains low-textured handle and lid, these parts cannot be recognized successfully. The texturedness also affects the quality of dense reconstruction. The reconstruction results in Figs. 10 and 12 show that blurry low-textured body and window glass are reconstructed ruggedly.

5. Conclusion

In this paper, we have presented a reconstruction framework for multiple objects in dynamic scenes. We designed a system based on the object recognition approach, which solves the problem of estimating object number and feature matching issues at

once. Our object-centered approach grounds on the fact that many of vision problems can be interpreted as correspondence problems. Thus, unlike the conventional flow-based approaches, the proposed method utilized the correspondence information acquired from the unsupervised co-recognition method. We presented our work as an integrated framework, which includes unsupervised object recognition, segmentation, and 3D model reconstruction from continuous or discontinuous dynamic scenes. Experimental results on various data have demonstrated the effectiveness of our approach, especially in the presence of abrupt motion of objects.

For our future work, we shall attempt to reconstruct a variety of scenes containing non-rigid objects with more complex motions. We hope this will eventually lead us to a practical technique, such as 3D conversion of old classic movies.

References

[1] N. Thakoor, J. Gao, V. Devarajan, Multibody structure-and-motion segmentation by branch-and-bound model selection, IEEE Transactions on Image Processing 19 (6) (2010) 1393–1402.

[2] F. Rothganger, S. Lazebnik, C. Schmid, J. Ponce, Segmenting, modeling, and matching video clips containing multiple moving objects, IEEE Transaction on Pattern Analysis and Machine Intelligence 29 (3) (2007) 477–491.

[3] K.E. Ozden, K. Schindler, L. van Gool, Simultaneous segmentation and 3D reconstruction of monocular image sequences, in: IEEE International Conference on Computer Vision, 2007.

[4] A.W. Fitzgibbon, A. Zisserman, Multibody structure and motion: 3-d reconstruction of independently moving objects, in: European Conference on Computer Vision, 2000.

[5] B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: International Joint Conference on Artificial Intelligence, 1981.

[6] T. Brox, J. Malik, Object segmentation by long term analysis of point trajectories, in: European Conference on Computer Vision, 2010.

[7] W.-C. Lu, Y.-C.F. Wang, C.-S. Chen, Learning dense optical-flow trajectory patterns for video object extraction, in: IEEE International Conference on Advanced Video and Signal Based Surveillance, 2010.

[8] K. Ozden, K. Cornelis, L.V. Eycken, L.V. Gool, Reconstructing 3D trajectories of independently moving objects using generic constraints, Computer Vision and Image Understanding 96 (3) (2004) 453–471.

[9] J. Wang, E.H. Adelson, Layered representation for motion analysis, in: IEEE Computer Vision and Pattern Recognition, 1993.

[10] S. Ayer, H.S. Sawhney, Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL Encoding, in: IEEE International Conference on Computer Vision, 1995.

[11] J. Wang, E.H. Adelson, Representing moving images with layers, IEEE Transactions on Image Processing 3 (5) (1994) 625–638.

[12] J. Wills, S. Agarwal, S. Belongie, What went where, in: IEEE Converence on Computer Vision and Pattern Recognition, 2003.

[13] J. Xiao, M. Shah, Accurate motion layer segmentation and matting, in: IEEE Computer Vision and Pattern Recognition, 2005.

[14] J. Costeira, T. Kanade, A multi-body factorization method for motion analysis, in: IEEE International Conference on Computer Vision, 1995.

[15] R. Vidal, S. Sastry, Optimal segmentation of dynamic scenes from two perspective views, in: IEEE Computer Vision and Pattern Recognition, 2003.

[16] Q. Ke, T. Kanade, A subspace approach to layer extraction, in: IEEE Computer Vision and Pattern Recognition, 2001.

[17] M. Cho, Y.M. Shin, K.M. Lee, Unsupervised detection and segmentation of identical objects, in: IEEE Computer Vision and Pattern Recognition, 2010.

[18] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: British Machine Vision Conference, 2002.

[19] K. Mikolajczyk, C. Schmid, An affine invariant interest point detector, in: European Conference on Computer Vision, 2002.

[20] F. Schaffalitzky, A. Zisserman, Multi-view matching for unordered image sets, or 'How Do I Organize My Holiday Snaps?', in: European Conference on Computer Vision, 2002.

[21] M. Brown, D. Lowe, Unsupervised 3D object recognition and reconstruction in unordered datasets, in: 3-D Digital Imaging and Modeling, 2005.

[22] M. Vergauwen, L.V. Gool, Web-based 3D reconstruction service, Machine Vision and Applications 17 (6) (2006) 411–426.

[23] D. Martinec, T. Pajdla, Robust rotation and translation estimation in multiview reconstruction, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[24] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1998.

[25] T.H. Kim, K.M. Lee, S.U. Lee, Generative image segmentation using random walks with restart, in: European Conference on Computer Vision, 2008.

[26] H. Edelsbrunner, D.G. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, IEEE Transactions on Information Theory 29 (4) (1983) 551–559.

[27] Y. Furukawa, J. Ponce, Accurate, dense, and robust multi-view stereopsis, IEEE Transaction on Pattern Analysis and Machine Intelligence 32 (8) (2010) 1362–1376.

[28] N. Snavely, S.M. Seitz, R. Szeliski, Modeling the world from internet photo collections, International Journal of Computer Vision 80 (2) (2008) 189–210.