Semantic Structure from Motion

Sid Yingze Bao and Silvio Savarese
Department of Electrical and Computer Engineering, University of Michigan at Ann Arbor
{yingze,silvio}@eecs.umich.edu

Abstract

Conventional rigid structure from motion *(SFM) addresses the problem of recovering the camera parameters (motion) and the 3D locations (structure) of scene points, given observed 2D image feature points. In this paper, we propose a new formulation called* Semantic Structure From Motion *(SSFM). In addition to the geometrical constraints provided by SFM, SSFM takes advantage of both semantic and geometrical properties associated with objects in the scene (Fig. 1). These properties allow us to recover not only the structure and motion but also the 3D locations, poses, and categories of objects in the scene. We cast this problem as a max-likelihood problem where geometry (cameras, points, objects) and semantic information (object classes) are simultaneously estimated. The key intuition is that, in addition to image features, the measurements of objects across views provide additional geometrical constraints that relate cameras and scene parameters. These constraints make the geometry estimation process more robust and, in turn, make object detection more accurate. Our framework has the unique ability to: i) estimate camera poses only from object detections, ii) enhance camera pose estimation, compared to feature-point-based SFM algorithms, iii) improve object detections given multiple uncalibrated images, compared to independently detecting objects in single images. Extensive quantitative results on three datasets – LiDAR cars, street-view pedestrians, and Kinect office desktop – verify our theoretical claims.*

1. Introduction

Understanding the 3D spatial and semantic structure of complex scenes from images is one of the critical capabilities of an intelligent visual system. Consider the photographs in Fig. 1(a). These show the same environment observed from a handful of viewpoints. Even if this is the first time you (the observer) have seen this environment, it is not difficult to infer: i) the spatial structure of the scene and the way objects are organized in the physical space; ii) the semantic content of the scene and its individual components. State-of-the-art methods for object recognition [9, 20, 10, 19] typically describe the scene with a list of class labels (e.g., a chair, a desk, etc...) along with their 2D location and scale, but are unable to account for the 3D spatial structure of the scene and object configurations (Fig. 1(b)). On the other hand, reconstruction methods (e.g., those based on SFM) [25, 8, 30, 23, 31] produce metric recovery of object and scene 3D structure (3D point clouds) but are mostly unable to infer the semantic content of its components (Fig. 1(c)).

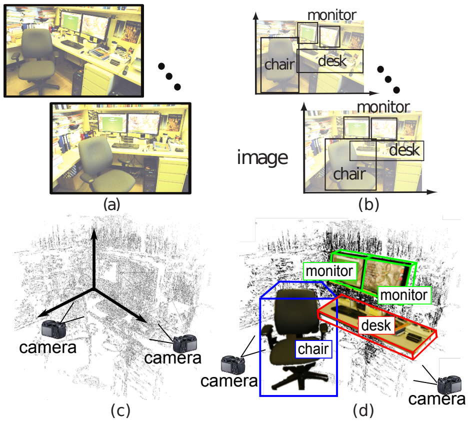


Figure 1: Main objective of SSFM. (a) Input photos showing the same environment observed from a handful of viewpoints. (b) Traditional object recognition algorithms identify objects in 2D without reasoning about the 3D geometry. (c) SFM returns 3D scene reconstruction (3D point clouds) with no semantic information attached to it. (d) SSFM aims to jointly recognize objects and reconstruct the underlying 3D geometry of the scene (cameras, points and objects).

In this paper we seek to fill this representation gap and propose a new framework for jointly recognizing objects as well as discovering their spatial organization in 3D (Fig. 1(d)). The key concept we explore in this work is that measurements across viewpoints must be semantically and geometrically consistent. By measurements, we refer to the set of objects that can be detected in the image (e.g. a chair or monitor in Fig. 1), their x,y location in the image, their scale (approximated by a bounding box) and their pose. Given a set of measurements from one view point, we expect to see a set of corresponding measurements (up to occlusions) from different view points which must be consistent with the fact that the view point has changed. For instance, the chair in Fig. 1(a) appears in two views and its location, scale and pose variation across the two views must be consistent with the view point transformation. In this work we exploit this property and introduce a novel joint probability model where object detection and 3D structure estimation are solved in a coherent fashion.

Our proposed method has the merit of enhancing both 3D reconstruction and visual recognition capabilities in two ways: i) *Enhancing 3D reconstruction*: Our framework can help overcome a crucial limitation of scene/object modeling methods. State-of-the-art SFM techniques mostly fail when dealing with challenging camera configurations (e.g., when the views are too few and the view baseline is too large). This failure occurs as it is very hard to establish correct feature correspondences for widely separated views. For instance, the 3D reconstruction in Fig. 1(c) was obtained using a state-of-the-art SFM algorithm [13] using 43 densely-sampled pictures of an office. The same algorithm would not work if we just used the two images in Fig. 1(a) for the reasons mentioned above. By reasoning at the semantic level, and by establishing object correspondences across views, our framework creates the conditions for overcoming this limitation. We show that our framework has the ability to estimate camera poses from object detections only. Moreover, our framework can still exploit traditional SFM constraints based on feature correspondences to make the 3D reconstruction process robust. We show that our method can significantly outperform across-view feature matching SFM algorithms such as [30, 22] (Tab. 1). ii) *Enhancing visual recognition*: Traditional recognition methods are typically prone to produce false alarms when appearance cues are not discriminative enough and no contextual information about the scene is available. For instance, the cabinet in Fig. 1(a) can be easily confused with a monitor as they both share similar appearance characteristics. By reasoning at the geometrical level, our framework is able to identify those hypotheses that are not consistent with the underlying geometry and reduce their confidence score accordingly. Our model leads to promising experimental results showing improvements in object detection rates compared with the state-of-the-art methods such as [9] (Fig. 4 and Tab. 2). Also, we show that we can automatically establish object correspondence across views.

Recently, a number of approaches have explored the idea of combining semantic cues with geometrical constraints for scene understanding. Notable examples are [14, 29, 21, 32, 16]. These focus on single images and, unlike our work, they do not attempt to enforce consistency across views. Moreover, they make restrictive assumptions on the camera and scene configuration. Other methods have been proposed to track objects with multi-view geometry [18], but they assume that the underlying scene geometry is available. A large number of works have proposed solutions for interpreting complex scenes from 3D data [11, 17, 27, 26] or a combination of 3D data and imagery [3]. However, in most of these methods 3D information is either provided by external devices (e.g. 3D scanning systems such as LiDAR) or using traditional SFM techniques. In either case, unlike our framework, the recognition and reconstruction steps are separated and independent. [5] attempts joint estimation using a "cognitive loop" but requires a dedicated stereo-camera architecture and makes assumptions about camera motion. To our best knowledge, this is the first work that seeks to make these two steps coherent within a setting that requires only images with uncalibrated cameras (up to internal parameters) and arbitrary scene-camera configurations.

2. The Semantic Structure from Motion Model

Conventional rigid *structure from motion* (SFM) addresses the problem of recovering camera parameters $\mathbf{C}$ and the 3D locations of scene points $\mathbf{Q}$, given observed 2D image feature points. In this paper, we propose a new formulation where, in addition to the geometrical constraints provided by SFM, we take advantage of both the semantic and geometrical properties associated with objects in the scene in order to recover $\mathbf{C}$ and $\mathbf{Q}$ as well as the 3D locations, poses, and category memberships of objects $\mathbf{O}$ in the scene. We call this *semantic structure from motion* (SSFM). The key intuition is that, in addition to image features, the measurements of objects across views provides additional geometrical constraints that relate camera and scene parameters. We formulate SSFM as a maximum likelihood estimation (MLE) problem whose goal is to find the best configuration of cameras, 3D points and 3D objects that are compatible with the measurements provided by a set of images.

2.1. Problem Formulation

In this section we define the SSFM problem and formulate it as an MLE problem. We first define the main variables involved in SSFM, and then discuss the MLE formulation.

Cameras. Let $\mathbf{C}$ denote the camera parameters. $\mathbf{C} = \{C^k\} = \{K^k, R^k, T^k\}$ where K is the camera matrix capturing the internal parameters, R rotation matrix, and T translation vector with respect to a common world reference system. K is assumed to be known, whereas $\{R, T\}$ are *unknown*. Throughout this paper, the camera is indexed by k as a superscript.

3D Points Q and Measurements $\mathbf{q}, \mathbf{u}$. Let $\mathbf{Q} = \{Q_s\}$ denote a set of 3D points Q_s. Each 3D point Q_s is specified by (X_s, Y_s, Z_s) describing the 3D point location in the world reference system. $\mathbf{Q}$ is an *unknown* in our problem. Denote by $\mathbf{q} = \{q_i^k\}$ the set of point *measurements* (image features) for all the cameras. Namely, q_i^k is the i^{th} point measurement in image (camera) k. A point measurement is described by the measurement vector $q^k = \{x, y, a\}_i^k$,

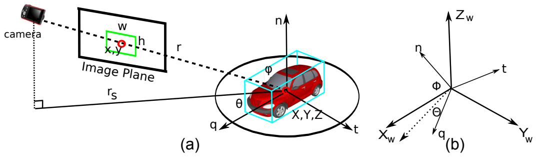


Figure 2: 3D object's location and pose parametrization. (a) Assume an object is enclosed by the tightest possible bounding cube. The object 3D location X, Y, Z is the centroid of the bounding cube. The object's pose is defined by the bounding cube's three perpendicular surface's norms that are n, q, t and parametrized by the angles Θ,Φ in a given world reference system (b). r is the ray connecting O and the camera center. Let zenith angle ϕ be the angle between r and n, and azimuth angle θ be the angle between q and r_S , where r_S is the projection of r onto the plane perpendicular to n. We parametrize an object measurement in the image by the location x, y of tightest bounding box enclosing the object, the width w and height h of the bounding box (object 2D scale), the object pose θ, ϕ, and class c.

where x, y describe the point image location, and a is a local descriptor that captures the local neighborhood appearance of the point in image k. These measurements may be obtained using feature detectors and descriptors such as [22, 34]. Since each image measurement $\{q_i^k\}$ is assumed to correspond to a certain physical 3D point Q_s, we model such correspondence by introducing an indicator variable u_i^k, where $u_i^k = s$ if $\{q_i^k\}$ corresponds to Q_s. A similar notation was also introduced in [7]. A set of indicator variables $\mathbf{u} = \{u_i^k\}$ allows us to establish feature correspondences across views and to relate feature matches with 3D point candidates (Sec. 2.3). Unlike [7], we assume the feature correspondences can be measured by feature matching algorithms such as [22]. Throughout this paper, Q and q are indexed by s and i respectively and they appear as subscripts.

3D Objects O and Measurements o. Let $\mathbf{O} = \{O_t\}$ denote a set of 3D objects O_t. The t^{th} 3D objects O_t is specified by a 3D location (X_t, Y_t, Z_t), a pose (Θ_t, Φ_t), and a category label c_t (e.g, *car*, *person*, etc...). Thus, a 3D object is parametrized by $O_t = (X, Y, Z, \Theta, \Phi, c)_t$ (Fig. 2). The set $\mathbf{O}$ is an *unknown* in our problem. Denote by $\mathbf{o} = \{o_j^k\}$ the set of object *measurements* for all the cameras. Thus, o_j^k is the j^{th} measurement of an object in image (camera) k. An object measurement is described by the following measurement vector $o_j^k = \{x, y, w, h, \theta, \phi, c\}_j^k$ (Fig. 2). As discussed in Sec. 2.2, these measurements may be obtained using any state-of-the art object detector that can return the probability that certain location x, y in an image is occupied by an object with category c, scale h, w, and pose θ, ϕ (e.g. [28]) [1]. Similar to the 3D point case, since each object measurement $\{o_j^k\}$ from image k is assumed to correspond to some physical 3D object O_t, such correspondence may be modeled by introducing an indicator variable v_j^k, where $v_j^k = t$ if o_j^k corresponds to 3D object O_t. However, for the object case, we seek to estimate object correspondence automatically as part of our inference process (Sec. 3). Thus, from this point on, we assume 3D object observations are given by $\mathbf{o}$. We denote 3D object and 2D object using the subscript index t and j respectively.

MLE formulation. Our goal is to estimate a configuration of $\mathbf{Q}$, $\mathbf{O}$ and $\mathbf{C}$ that is consistent with the feature point measurements $\mathbf{q}$, $\mathbf{u}$ and the object measurements $\mathbf{o}$. We formulate this estimation as the one of finding $\mathbf{Q}, \mathbf{O}, \mathbf{C}$ such that the joint likelihood is maximized:

$$\begin{aligned}\{\mathbf{Q}, \mathbf{O}, \mathbf{C}\} &= \arg\max_{Q,O,C} \Pr(\mathbf{q}, \mathbf{u}, \mathbf{o}|\mathbf{Q}, \mathbf{O}, \mathbf{C}) \\ &= \arg\max_{Q,O,C} \Pr(\mathbf{q}, \mathbf{u}|\mathbf{Q}, \mathbf{C}) \Pr(\mathbf{o}|\mathbf{O}, \mathbf{C}) \quad (1)\end{aligned}$$

where the last expression is obtained by assuming that, given $\mathbf{C}$, $\mathbf{Q}$ and $\mathbf{O}$, the measurements associated with 3D objects and 3D points are conditionally independent. In the next two sections we show how to estimate the two likelihood terms $\Pr(\mathbf{q}, \mathbf{u}|\mathbf{Q}, \mathbf{C})$ (Eq. 4 or 5) and $\Pr(\mathbf{o}|\mathbf{O}, \mathbf{C})$ (Eq. 3).

2.2. Object Likelihood $\Pr(\mathbf{o}|\mathbf{O}, \mathbf{C})$

$\Pr(\mathbf{o}|\mathbf{O}, \mathbf{C})$ measures the likelihood of object measurements $\mathbf{o}$ given the the camera and object configurations $\mathbf{O}, \mathbf{C}$. This term can be estimated by computing the *agreement* between predicted measurements and actual measurements. Predicted measurements are obtained by introducing a mapping $\omega_t^k = \omega^k(O_t) = \omega^k((X, Y, Z, \Theta, \Phi, c)_t)$ that relates the parameters describing the 3D object O_t to the image of camera C^k. Thus, ω_t^k is a parameter vector that contains the predicted location, pose, scale and category of O_t in C^k. Next, we present expressions for predicting the measurements and relating them to actual measurements and for obtaining an estimate of the likelihood term.

Computing Predicted Measurements. The transformation $\omega_t^k = \omega^k(O_t)$ can be computed once cameras $\mathbf{C}$ are known. Specifically, let us denote by X_t^k, Y_t^k, Z_t^k the 3D location of O_t in the reference system of C^k and by Θ_t^k, Φ_t^k its 3D pose (these can be obtained from $X_t, Y_t, Z_t, \Theta_t, \Phi_t$ in the world reference system by means of a (known) rigid transformation). Predicted location (x_t^k, y_t^k) and pose (ϕ_t^k, θ_t^k) of O_t in camera C^k can be computed by using the camera projection matrix [15] as $[x_t^k, y_t^k, 1]' = K^k[X_t^k, Y_t^k, Z_t^k]'/Z_t^k$ and $[\phi_t^k, \theta_t^k] = [\Phi_t^k, \Theta_t^k]$. Predicting 2D object scales in the image requires a more complex geometrical derivation that goes beyond the scope of this paper. We introduce an approximated simplified mapping defined as follows:

$$\begin{cases} w_t^k = f_k \cdot W(\Theta_t^k, \Phi_t^k, c_t)/Z_t^k \\ h_t^k = f_k \cdot H(\Theta_t^k, \Phi_t^k, c_t)/Z_t^k \end{cases} \quad (2)$$

where w_t^k, h_t^k denote the predicted object 2D scale (similar to Fig. 2), f_k is the focal length of the k^{th} camera. $W(\Theta_t^k, \Phi_t^k, c_t)$ or $H(\Theta_t^k, \Phi_t^k, c_t)$ is a (scalar) mapping that describes the typical relationship between physical ob-

[1] State of the art object detectors such as [9] can be modified so as to enable pose classification, as discussed in Sec.4.1.

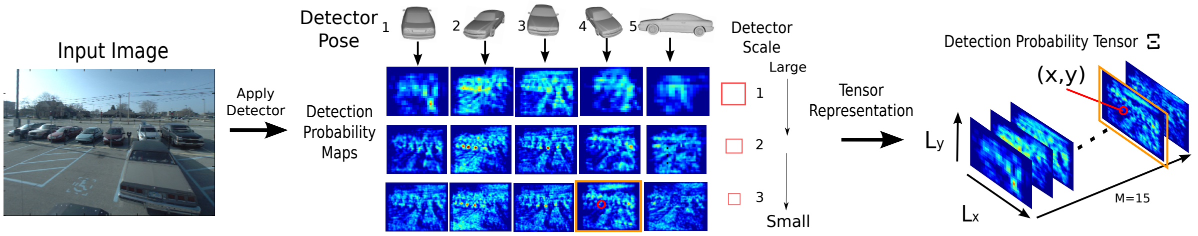


Figure 3: Multi-pose and multi-scale object detection illustration. The "probability maps" are obtained by applying car detector with different scales and poses on the left image. The color from red to deep blue indicates the detector response from high to low. We used LSVM [9] (Sec. 4.1) to obtain these probability maps. In this example, Ξ has dimensions $L_x \times L_y \times 15$. If the scale=3 (small), pose=4, and category=car, Π will return the index $\pi = 14$ (the red circle). Thus, $\Xi(x, y, 14)$ will return the confidence of detecting a car at small scale and pose=4 at location x, y in the image (the orange rectangle).

ject bounding cube and object image bounding box. This mapping is a function of the object pose Θ_t^k, Φ_t^k and category c_t. It can be learned by using ground truth 3D object bounding cubes and corresponding observations using ML regressor [1]. In conclusion, the equations above allow us to fully estimate the object prediction vector $\omega_t^k = \{x, y, w, h, \phi, \theta, c\}_t^k$ for object O_t in camera C^k.

Measurements as Probability Maps. $\Pr(\mathbf{o}|\mathbf{O}, \mathbf{C})$ can be now estimated by computing the *agreement* between predicted measurements and actual measurements. Such agreement is readily available using the set of probability values returned by object detectors such as [9] applied to images (Fig. 3). The output of this detection process for the image of C^k is a tensor Ξ^k of M probability maps wherein each map captures the likelihood that an object of category c with scale w, h and pose θ, ϕ presents at location x, y in the image. Thus, we can interpret Ξ^k as one $L_x \times L_y \times M$ tensor, where L_x and L_y are the image width and height and M adds up to the number of object categories, scales and poses. Let us denote by $\Pi : \{w, h, \phi, \theta, c\} \rightarrow \pi \in 1 \ldots M$ the *indexing function* that allows retrieval from Ξ^k the detection probability at any location x, y given a set of values for scale, pose and category. Fig. 3 shows an example of a set of 15 probability maps for only one object category (i.e., the *car* category), three scales and five poses associated with a given image. Notice that since measurements can be extracted directly from Ξ^k once the mapping 3D-object-image ω is computed, the 2D objects of the k^{th} image are automatically associated with the 3D objects. As a result, across-view one-to-one object correspondences are also established.

Estimating the likelihood term. The key idea is that the set Ξ^k of probability maps along with π can be used to estimate $\Pr(\mathbf{o}|\mathbf{O}, \mathbf{C})$ given the predicted measurements. To illustrate this, let us start by considering an estimation of the likelihood term $\Pr(o|O_t, C^k)$ for O_t observed from camera C^k. Using ω_t^k, we can predict the object's scale $\{w, h\}_t^k$, pose $\{\phi, \theta\}_t^k$ and category c_t^k. This allows us to retrieve from Ξ^k the probability of detecting an object at the predicted location $\{x, y\}_t^k$ by using the indexing function π_t^k, and in turn estimate $\Pr(o|O_t, C^k) = \Xi^k(x_t^k, y_t^k, \pi(w_t^k, h_t^k, \phi_t^k, \theta_t^k, c_t^k))$. Assuming that objects are independent from each other and camera configurations are independent, the joint likelihood of objects and cameras can be approximated as:

$$\Pr(\mathbf{o}|\mathbf{O}, \mathbf{C}) \propto \prod_t^{N_t} \Pr(\mathbf{o}|O_t, \mathbf{C}) \propto \prod_t^{N_t} (1 - \prod_k^{N_k} (1 - \Pr(o|O_t, C^k))) \quad (3)$$

where N_t is the number of objects and N_k is the number of cameras. N_t is in general unknown, but it can be estimated using detection probability maps (Sec.3.1). Notice that this term does not penalize objects that are observed only by a portion of images while they are truncated or occluded in other images. $\Pr(\mathbf{o}|O_t, \mathbf{C})$ is only partially affected by an occluded or truncated object O_t in the k^{th} image even if the object leads to a low value for $\Pr(o|O_t, C^k)$.

2.3. Points Likelihood $\Pr(\mathbf{q}, \mathbf{u}|\mathbf{Q}, \mathbf{C})$

$\Pr(\mathbf{q}, \mathbf{u}|\mathbf{Q}, \mathbf{C})$ measures the likelihood of the 3D points and cameras given the measurements of 3D points and their correspondences across views. This likelihood term can be estimated by computing the *agreement* between predicted measurements and actual measurements. Similar to the 3D object case, predicted measurements are obtained by introducing a mapping from 3D points to the images.

Predicted Measurements. Predicted measurements can be easily obtained once the cameras $\mathbf{C}$ are known. We indicate by q_s^k the predicted measurement of the s^{th} point Q_s in camera C^k which can be computed by using the camera C^k projection matrix, similar to Sec. 2.2. Recall that q_s^k is expressing a location x, y in pixels in the image. Since we know which point is being projected, we have a prediction for the indicator variable as well. This is equivalent to predicting the projection q_s^k of Q_s to camera C^k and the projection q_s^k of Q_s to C^k are in correspondence.

Point Measurements. Point measurements are denoted by $q_i^k = \{x, y, a\}_i^k$, where x, y describe the point location in image k of measurement i, and a is a local descriptor that captures the local appearance of the point in a neighborhood of image k. We obtain location measurements $\{x, y\}_i^k$ using a DOG detector equipped with a SIFT descriptor for estimating a_i^k [22]. Measurements for feature correspondences (matches) across images are obtained by matching the point features.

Estimating the likelihood term. $\Pr(\mathbf{q}, \mathbf{u}|\mathbf{Q}, \mathbf{C})$ can be estimated by computing the agreement between predicted measurements and the actual measurements. To illustrate this, let us start by considering the likelihood term $\Pr(q|O_s, C^k)$ for one point Q_s and for camera C^k. As introduced in [7], one possible strategy for computing such agreement assumes that the location of measurements and predictions are equal up to a noise n - that is, $q_i^k = q_s^k + n$, where $s = u_i^k$. If we assume zero mean Gaussian noise, we can estimate $\Pr(q_i^k | Q_s, C^k) \propto \exp(-(q_i^k - q_{u_i^k}^k)^2/\sigma_q)$, leading to the following expression for the likelihood:

$$\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C}) = \prod_i^{N_Q} \prod_k^{N_k} \exp(-(q_i^k - q_{u_i^k}^k)^2/\sigma_q) \quad (4)$$

where N_k is the number of cameras, N_Q is the number of points, and σ_q is the variance of 2D point projection measurement error. This is obtained by assuming independence among points and among cameras.

We also propose an alternative estimator for $\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C})$. While this estimator leads to a coarser approximation for the likelihood, it makes the inference process more efficient and produces more stable results. This new estimator exploits the epipolar constraints relating camera pairs. Given a pair of cameras C^l and C^k, we can estimate the fundamental matrix $F_{l,k}$. Suppose q_i^k, q_j^l are from C^k and C^l respectively, and the matching algorithm predicts that q_i^k and q_j^l are in correspondence. $F_{l,k}$ can predict the epipolar line $\xi_i^{l,k}$ (or $\xi_j^{k,l}$) of q_i^k (or q_j^l) in image C^l (or C^k). If we model the distance[2] $d_{j,i}^{l,k}$ between $\xi_i^{l,k}$ and q_j^l as zero-mean Gaussian with variance σ_u, $\Pr(q_i^k, q_j^l | Q_s, C_l, C_k) \propto \exp(-d_{j,i}^{l,k}/\sigma_u)$. Notice that this expression does not account for appearance similarity between matched features – that is the similarity between the descriptors a_i^k and a_j^l. We model appearance similarity as $\exp(-\frac{\alpha(a_i^k, a_j^l)}{\sigma_\alpha})$ where $\alpha(\cdot, \cdot)$ captures the distance between two feature vectors and σ_α the variance of the appearance similarity. Overall, we obtain the following expression for the likelihood term:

$$\begin{aligned} \Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C}) &\propto \prod_{k \neq l}^{N_k} \prod_{i \neq j}^{N_s} \Pr(q_i^k, q_j^l | Q_s, C_l, C_k) \\ &\propto \prod_{k \neq l}^{N_k} \prod_{i \neq j}^{N_s} \exp(-\frac{d_{j,i}^{l,k}}{\sigma_u}) \exp(-\frac{\alpha(a_i^k, a_j^l)}{\sigma_\alpha}) \quad (5) \end{aligned}$$

Eq. 5 is obtained by assuming that feature locations and appearance are independent. During the learning stage, we learn the variance σ_u and σ_α using an ML estimator on a validation set. Notice that $\Pr(\mathbf{q}, \mathbf{u} | \mathbf{Q}, \mathbf{C})$ is no longer a function of Q_s. This significantly reduces the search space for solving the MLE problem.

3. Max-Likelihood Estimation with Sampling

Our goal is to estimate camera parameters, points, and objects so as to maximize Eq. 1. Due to the high dimensionality of the parameter space, we propose to sample $\mathbf{C}, \mathbf{Q}, \mathbf{O}$ from $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$ similar to [7]. This allows us to approximate the distribution of $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$ and find the $\mathbf{C}, \mathbf{Q}, \mathbf{O}$ that maximize the likelihood. In Sec. 3.1 we discuss the initialization of the sampling process, and in Sec. 3.2 we describe a modified formulation of the Markov Chain Monte Carlo (MCMC) sampling algorithm for solving the MLE problem.

3.1. Parameter Initialization

Appropriate initialization of cameras, objects, and points is a critical step in the sampling method. We initialize camera configurations (i.e. estimate camera configurations that are geometrically compatible with the observations) using feature point matches and object detections.

Camera Initialization by Feature Points. We follow [23] to initialize (estimate) $\mathbf{C}$ from image measurements $\mathbf{q}$. Due to the metric reconstruction ambiguity, we scale the estimated camera translation with several random values to obtain several camera pose initializations.

Camera Initialization by Objects. We use a standard object detector [9] to detect 2D objects and estimate object pose and scale (Sec. 4.1). Next, we use such object detections to form possible object correspondences and use these to estimate several possible initial camera configurations. It can be shown [1] that object pose and scale consistency constraints (across views) can be used to reduce the large number of possible initial configurations.

Points and Objects Initialization. Camera configurations obtained by using points and objects form the initialization set. For each of these configurations, object detections are used to initialize objects in 3D using the mapping in Eq. 2. If certain initialized 3D objects are too near to others (location and pose-wise), they are merged to a single one. Similarly, for each camera configuration, feature points $\mathbf{q}$ are used to initialize 3D points $\mathbf{Q}$ by triangulation[15]. Correspondences between $\mathbf{q}$ and $\mathbf{Q}$ are established after the initialization. We use index r to indicate one out of R possible initializations for objects, cameras and points $(\mathbf{C}_r, \mathbf{O}_r, \mathbf{Q}_r)$.

3.2. Sample and Maximize the Likelihood

We sample $\mathbf{C}, \mathbf{O}, \mathbf{Q}$ from the underlying $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$ using a modified Metropolis algorithm [12] (Algo. 1). Since the goal of the sampling is to identify a maximum, the samples should occur as near to $\max \Pr(\mathbf{q}, \mathbf{u}, \mathbf{o} | \mathbf{Q}, \mathbf{C}, \mathbf{O})$ as possible, so as to increase the efficiency of the sampling algorithm. Thus, we only randomly sample $\mathbf{C}$, while the best configuration of $\mathbf{O}$ and $\mathbf{Q}$ given the proposed $\mathbf{C}$ are estimated during each sampling step. In step 3, the estimation of $\mathbf{O}'$ is obtained by greedy search within a neighborhood of the objects proposed during the previous sampling step. Since the object detection scale and pose are highly quantized, the greedy search yields efficient and robust results in practice. The estimation of $\mathbf{Q}$ is based on the minimization of the projection error (similar to [33]).

Following Algo. 1 and using the r^{th} initialization, we

[2] To account for outliers, we set a threshold on $d_{j,i}^{l,k}$. Namely, if $\bar{d}_{j,i}^{l,k}$ is the measurement, we set $d_{j,i}^{l,k} = \min(\bar{d}_{j,i}^{l,k}, \Gamma)$. We learn the outlier threshold Γ using a validation set.

Algorithm 1 MCMC sampling from r^{th} initialization. See [1] for details

```
1: Start with $r$th proposed initialization $\mathbf{C}_r, \mathbf{O}_r, \mathbf{Q}_r$. Set counter $v = 0$.
2: Propose new camera parameter $\mathbf{C}'$ with Gaussian probability whose mean is the previous sample and the co-variance matrix is uncorrelated.
3: Propose new $\mathbf{O}'$ within the neighborhood of previous object's estimation to maximize $\Pr(\mathbf{o}|\mathbf{O}', \mathbf{C}')$.
4: Propose new $\mathbf{Q}'$ with $\mathbf{C}'$ to minimize the point projection error.
5: Compute the acceptance ratio $\alpha = \frac{\Pr(\mathbf{q},\mathbf{u},\mathbf{o}|\mathbf{C}',\mathbf{O}',\mathbf{Q}')}{\Pr(\mathbf{q},\mathbf{u},\mathbf{o}|\mathbf{C},\mathbf{O},\mathbf{Q})}$
6: If $\alpha \geqslant \varrho$ where $\varrho$ is a uniform random variable $\varrho \sim U(0,1)$, then accept $(\mathbf{C}, \mathbf{O}, \mathbf{Q}) = (\mathbf{C}', \mathbf{O}', \mathbf{Q}')$. Record $(\mathbf{C}, \mathbf{O}, \mathbf{Q})$ as a sample in $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$.
7: $v = v + 1$. Goto 2 if $v$ is smaller than the predefined max sample number; otherwise return $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$ and end.
```

generate the sample $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$. From all of the samples, we estimate the maximum of $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o}|\mathbf{Q}, \mathbf{C}, \mathbf{O})$ as follows. We concatenate $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}_r$ from different initializations into one sample point set $\{\mathbf{C}, \mathbf{O}, \mathbf{Q}\}$. Next, the frequency of the samples will provide an approximation of the distribution of $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o}|\mathbf{Q}, \mathbf{C}, \mathbf{O})$. To identify the maximum, the MeanShift algorithm [4] is employed to cluster the samples. The center of the cluster with the highest sample number is regarded as the approximation of the maximum of $\Pr(\mathbf{q}, \mathbf{u}, \mathbf{o}|\mathbf{Q}, \mathbf{C}, \mathbf{O})$ and thus taken as the solution of the final estimation of $\mathbf{C}, \mathbf{O}, \mathbf{Q}$.

Algo. 1 can be applied using either Eq. 4 or 5. If Eq. 5 is used, the estimation is greatly simplified as $\mathbf{Q}$ no longer appears in the optimization process. Hence step 4 is no longer required. Our experiments use the latter implementation.

4. Evaluation

In this section we qualitatively demonstrate the ability of the SSFM model to jointly estimate the camera pose and improve the accuracy in detecting objects. We test SSFM on three datasets: the publicly available Ford Campus Vision and LiDAR Dataset[24], a novel Kinect office dataset[3], and a novel street-view pedestrian stereo-camera dataset. Although SSFM does not use any information from 3D points, the calibrated 3D points from LiDAR and Kinect allows us to easily obtain the ground truth information. The typical running time for one image pair with our Matlab single-thread implementation is ~20 minutes. Benchmark comparisons with the state-of-the-art baseline detector *Latent SVM* [9] and point-based SfM approach *Bundler* [30] demonstrate that our method achieves significant improvement on object detection and camera pose estimation results.

To evaluate the object detection performance, we plot precision-recall (PR) curves and compare the average-precision (AP) value with baseline detector LSVM [9]. Detection performance is computed by projecting the estimated 3D object bounding cube into each image and by measuring the overlap ratio (>50%) between such projection and ground truth bounding box. LSVM baseline detector is applied to each image used by SSFM. Thus PR values are computed for each image for fair comparison.

To evaluate the camera pose estimate, we compare the camera pose estimation of SSFM with the state-of-the-art point-based structure-from-motion approach Bundler [30]. Bundler first employs the SIFT feature, five-points algorithm [23] and RANSAC to compute the fundamental matrix, and then applies Bundle Adjustment [33]. In certain configurations (e.g. wide baseline) RANSAC or Bundle Adjustment fail to return results. In such cases we take the camera pose estimation of five-points algorithm as the results for comparison. We follow the evaluation criteria in [23]. When comparing the camera pose estimation, we always assume the first camera to be at the canonical position. Denote R_{gt} and T_{gt} as the ground truth camera rotation and translation, and R_{est} and T_{est} the estimated camera rotation and translation. The error measurement of rotation e_R is the minimal rotating angle of $R_{gt}R_{est}^{-1}$. The error measurement of translation e_T is evaluated by the angle between the estimated baseline and the ground truth baseline, and $e_T = \frac{T_{gt}^T R_{gt}^{-T} R_{est}^{-1} T_{est}}{|T_{gt}| \cdot |T_{est}|}$. For a fair comparison, the error results are computed on the second camera.

We also analyze the performance of SSFM as a function of the number of cameras (views). A testing set is called N-view set if it contains M groups of N images. The testing sets with smaller number of views are first generated (i.e. 2-view set is the very first). If one N-view set is used, the $N+1$-view testing set is generated by adding one additional random view to each of the M groups of N images.

4.1. Implementation Details

SSFM requires an object detector that is capable of determining the object pose. We use the state-of-the-art object detector [9] and treat object poses as extra-classes for each object category.

4.2. Ford Campus Vision Dataset[24]

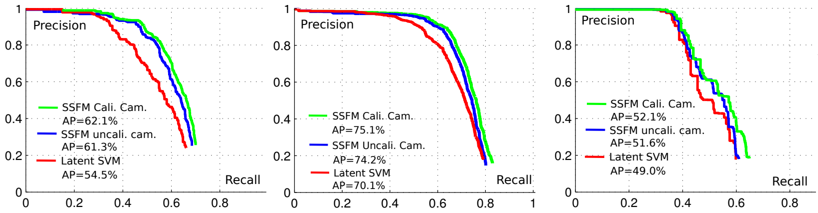
The Ford Campus Vision dataset consists images of cars aligned with 3D scans obtained using a LiDAR system. Ground truth camera parameters are also available. Our training / testing set contains 150 / 200 images of 4 / 5 different scenarios. We randomly select 350 image pairs out of the testing images with the rule that every pair of images must capture the same scene. The training set for the car detector is the 3D object dataset [28]. This training set consists of 8 poses.

Camera Pose Estimation: SSFM obtains smaller translation estimation error than Bundler and comparable rota-

Dataset	$\bar{e}_T$ Bundler/SSFM	$\bar{e}_R$ Bundler/SSFM
Ford Campus Car	26.5/$\mathbf{19.9}^\circ$	$\mathbf{0.47}^\circ$/0.78°
Street Pedestrian	27.1°/$\mathbf{17.6}^\circ$	21.1°/$\mathbf{3.1}^\circ$
Office Desktop	8.5°/$\mathbf{4.7}^\circ$	9.6°/$\mathbf{4.2}^\circ$

Table 1: Evaluation of camera pose estimation for two camera case. $\bar{e_T}$ represents the mean of the camera translation estimation error, and $\bar{e}_R$ the mean of the camera rotation estimation error.

[3] available at the author's web-page [1]



(a) Car (b) Pedestrian (c) Office

Figure 4: Detection PR results by SSFM with calibrated cameras (green), SSFM with uncalibrated cameras (blue) and LSVM [9] (red). Fig. 4c shows average results for mouse, keyboard and monitor categories. SSFM is applied on image pairs randomly selected from the testing set (unless otherwise stated). Calibration is obtained from ground truth.

tion estimation error (Tab. 1).

Object Detection: The PR by SSFM and the baseline detector are plotted in Fig. 4a. Since ground truth annotation for small objects is difficult to obtain accurately, in this dataset we only test scales whose bounding box areas are larger than 0.6% of the image area. SSFM improves the detection precision and recall.

Camera Baseline Width v.s. Pose Estimation: We analyze the effect of baseline width on the camera pose estimation. Since the rotation estimations of both Bundler and SSFM contain little error, we only show the translation estimation error v.s. camera baseline width (Fig. 5a). This experiment confirms the intuition that a wider baseline impacts more dramatically the performance of methods based on low level feature matching than does on methods such as SSFM where higher level semantics are used.

Comparison for Different Number of Cameras: Tab. 2 shows the camera pose estimation error and the object detection AP as a function of the number of views (cameras) used to run SSFM. As more cameras are available, SSFM tends to achieve better object detection result and camera translation estimation.

3D Object Localization Performance: Due to the metric-reconstruction ambiguity, we use calibrated cameras in this experiment to enforce that the coordinates of 3D objects have a physical meaning. We manually label the 3D bounding boxes of cars on the LiDAR 3D point cloud to obtain the ground truth car 3D locations. We consider a 3D detection to be true positive if the distance between its centroid and ground truth 3D object centroid is smaller than a threshold ∂ (see figure captions). The 3D object localization for one camera (single view) is obtained by using its 2D bounding box scale and location [2]. SSFM performance increases as the number of views grows (Fig. 5b).

Object-based Structure from Motion: We disable the feature point detection and matching, thus no 2D points are used (i.e. just maximize $\Pr(\mathbf{o}|\mathbf{C}, \mathbf{O})$). For the two-view case, the detection AP increases from the baseline 54.5% to 55.2%, while the error of camera pose estimation is $\bar{e}_T = 81.2°$ and $\bar{e}_R = 21.2°$. To the best of our knowledge, this is the first time SFM has been tested based only on high-level cues (objects) rather than low-level / middle-level cues (e.g. points, lines, or areas).

Camera #	2	3	4
Det. AP (Cali. Cam.)	62.1%	63.6%	64.2%
Det. AP (Uncali. Cam.)	61.3%	61.7%	62.6%
$\bar{e}_T$	$19.9°$	$16.2°$	$13.9°$

Table 2: Camera pose estimation errors and object detection AP v.s. numbers of cameras on the Ford-car dataset. The baseline detector AP is 54.5%.

4.3. Kinect Office Desktop Dataset[1]

We use Microsoft's Kinect to collect images and corresponding 3D range data of several static indoor office environments. The ground truth camera parameters are obtained by aligning range data across different views. We manually identify the locations of ground truth 3D object bounding cubes similarly to the way we process Ford dataset. The objects in this dataset are monitors, keyboards, and mice. The testing and training sets contain 5 different office desktop scenarios respectively and each scenario has ~50 images. From each scenario, we randomly select 100 image pairs for testing or training. SSFM performance is evaluated using the ground truth information and compared against baseline algorithms. We show these results as Fig. 4c, Tab. 1, Fig. 5c, and Fig. 5d. Refer to the figure captions for more details.

4.4. Stereo Street-view Pedestrian Dataset

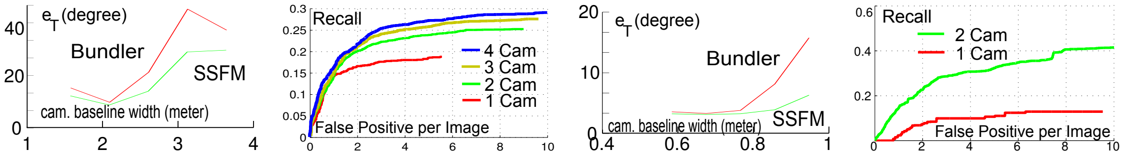
We collected this dataset by simultaneously capturing pairs of images of street-view pedestrians. The two cameras are pre-calibrated so that the ground-truth camera poses are measured and known. The object category in this dataset is pedestrian. The training set of object detector is INRIA pedestrian dataset [6] with no pose label. The two cameras are parallel and their relative distance is $4m$. The typical object-to-camera distance is $5 \sim 10m$. The training set contains 200 image pairs in 5 different scenarios. The testing set contains 200 image pairs in 6 other scenarios. SSFM attains smaller camera pose estimation error compared to Bundler (Tab. 1) and better detection rates than LSVM (Fig. 4b). Notice in this dataset the baseline width of the two cameras is fixed thus we cannot analyze the camera pose estimation error v.s. camera baseline width and cannot carry out experiments with multiple cameras.

5. Conclusion

This paper presents a new paradigm called the semantic structure from motion for jointly estimating 3D objects, 3D points and camera poses from multiple images. We see this work as a promising step toward the goal of coherently interpreting the geometrical and semantic content of complex scenes.

Acknowledgments

We acknowledge the support of NSF CAREER #1054127 and the Gigascale Systems Research Center. We wish to thank Mohit Bagra for his help in collecting the Kinect dataset and Min Sun for helpful feedback.



(a) T est. error v.s. the baseline (b) 3D obj. localization. $\eth = 2.5m$ (c) T est. error v.s. the baseline. (d) 3D obj. localization. $\eth = 0.1m$.

Figure 5: System analysis of SSFM on Ford Car Dataset (a)(b) and Kinect Office Dataset (c)(d).

References

[1] http://www.eecs.umich.edu/vision/projects/ssfm/index.html.

[2] S. Y.-Z. Bao, M. Sun, and S. Savarese. Toward coherent object detection and scene layout understanding. In *CVPR*, 2010.

[3] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *In Proc. 10th ECCV*, 2008.

[4] Y. Cheng. Mean shift, mode seeking, and clustering. *PAMI*, 1995.

[5] N. Cornelis, B. Leibe, K. Cornelis, and L. Gool. 3d urban scene modeling integrating recognition and reconstruction. *IJCV*, 78(2-3):121–141, 2008.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] F. Dellaert, S. Seitz, S. Thrun, and C. Thorpe. Feature correspondence: A markov chain monte carlo approach. In *NIPS*, 2000.

[8] A. R. Dick, P. H. S. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *IJCV*, 60(2):111–134, 2004.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2009.

[10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, pages 264–271, 2003.

[11] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, pages 224–237, 2004.

[12] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall, 1996.

[13] M. Golparvar-Fard, F. Pena-Mora, and S. Savarese. D4ar- a 4-dimensional augmented reality model for automating construction progress data collection, processing and communication. In *TCON Special Issue: Next Generation Construction IT*, 2009.

[14] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

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

[16] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1), 2008.

[17] D. Huber. Automatic 3d modeling using range images obtained from unknown viewpoints. In *Int. Conf. on 3-D Digital Imaging and Modeling*, 2001.

[18] S. M. Khan and M. Shah. A multi-view approach to tracking people in dense crowded scenes using a planar homography constraint. In *ECCV*, 2006.

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[20] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV 2004 workshop on statistical learning in computer vision*, 2004.

[21] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009.

[22] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[23] D. Nister. An efficient solution to the five-point relative pose problem. *TPAMI*, 2004.

[24] G. Pandey, J. R. McBride, and R. M. Eustice. Ford campus vision and lidar data set. *International Journal of Robotics Research*, 2011.

[25] M. Pollefeys and L. V. Gool. From images to 3d models. *Commun. ACM*, 45(7):50–55, 2002.

[26] M. Reynolds, J. Doboš, L. Peel, T. Weyrich, and G. J. Brostow. Capturing time-of-flight data with confidence. In *CVPR*, 2011.

[27] R. Rusu, Z. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11), 2008.

[28] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[29] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 31(5):824–840, 2009.

[30] N. Snavely, S. M. Seitz, and R. S. Szeliski. Modeling the world from internet photo collections. *IJCV*, (2), 2008.

[31] S. Soatto and P. Perona. Reducing "structure from motion": a general framework for dynamic vision. part 1: modeling. *International Journal of Computer Vision*, 20, 1998.

[32] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Depth from familiar objects: A hierarchical model for 3d scenes. In *CVPR*, 2006.

[33] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbob. Bundle adjustment: a modern synthesis. In *Vision Algorithms: Theory and Practice*, 1999.

[34] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference*, 2000.

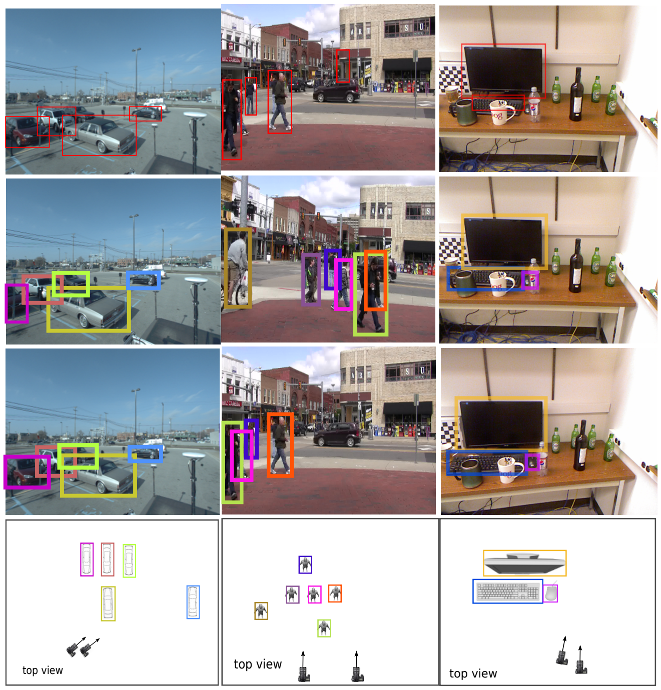


Figure 6: Typical Examples. Row 1: Baseline object detection in the 1^{st} image; row 2,3: the final joint object detections projected in the 1^{st} and 2^{nd} image; row 4: the top view of the scene. Colors in the last three rows show the object correspondences established by SSFM. See [1] for more results.