Semantic Structure From Motion with Object and Point Interactions

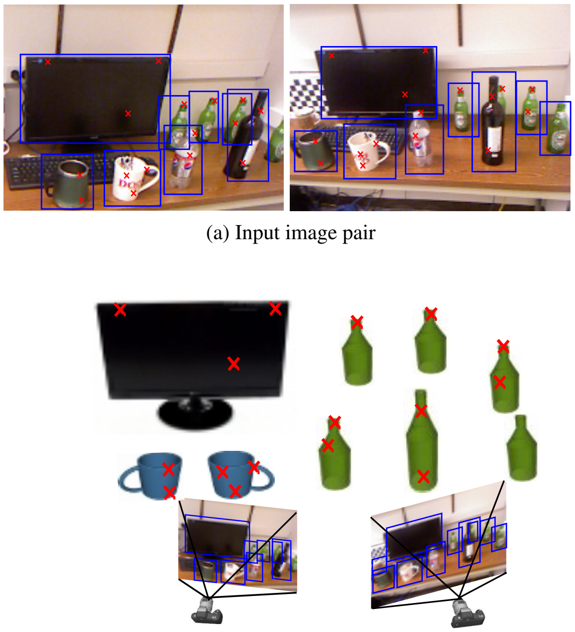
Sid Yingze Bao, Mohit Bagra, Silvio Savarese
University of Michigan at Ann Arbor, USA
{yingze,mohitbag,silvio}@eecs.umich.edu
http://www.eecs.umich.edu/vision/projects/ssfm/index.html

Abstract

We propose a new method for jointly detecting objects and recovering the geometry of the scene (camera pose, object and scene point 3D locations) from multiple semi-calibrated images (camera internal parameters are known). To achieve this task, our method models high level semantics (i.e. object class labels and relevant characteristics such as location and pose) and the interaction (correlations) of objects and feature points within the same view and across views. We validate our algorithm against state-of-the-art baseline methods using two public datasets - Ford Car dataset and Kinect Office dataset [1] - and show that we: i) significantly improve the camera pose estimation results compared to point-based SFM algorithm; ii) achieve better 2D and 3D object detection accuracy than using single images separately. Our algorithm is critical in many application scenarios including object manipulation and autonomous navigation.

1. Introduction

Interpreting the geometrical and semantic content of a scene is a key problem in robotics and computer vision. Not only a visual system would need to have the capability to recognize the important semantic phenomena in the scene (e.g. objects), but also to localize such components in the 3D physical space. This would be critical as the robotic agent navigates through the environment (obstacle avoidance, path planning, etc..) as well as interacts with it (object grasping and manipulation). Most of the existing methods have attempted to: 1) estimate the geometry alone (structure from motion (SFM), e.g. [11, 20, 5, 25]) without being able to recognize and identify the objects within the scene; 2) just infer the semantics (e.g. [6, 15, 7, 18]) without having the capability of determining the underlying geometry; 3) recognize objects from 3D range data [8, 14, 22] by assuming that an underlying 3D reconstruction of the environment is available in the camera (robot) reference system. Recent works propose to simultaneously estimate geometry and semantics from a single image [9, 13, 26, 2, 24, 3, 4, 10] or multiple images [1]. In particular, the novelty of [1] is to use high level features such as objects and they geometrical characteristics (pose and scale) as key building blocks to establish geometrical constraints across different observations. Authors named this approach as *Semantic Structure From Motion* (SSFM). While promising, however, [1] is not able to seamlessly combine rigid geometrical con-



(a) Input image pair

(b) Reconstructed Scene

Figure 1: We propose a new method for jointly detecting objects and recovering the geometry of the scene (camera pose, object and scene point 3D locations) from two or multiple un-calibrated images of the scene. We achieve this by leveraging across-view i) object location and pose consistency, ii) feature correspondences and iii) interactions (correlations) between objects (see colored bounding boxes) and feature points (see red crosses). (a) shows two images of the same scene. (b) shows our 3D reconstruction and object detection results within the cameras 3D reference system.

straints (e.g. those used in SFM methods which leverage feature points lying on both objects and scene elements) with relationships induced by the object semantics. In other words, [1] handles objects and feature points as independent entities. In this paper, we explicitly exploit the correlation (interaction) between high-level elements (objects) and low-level ones (image features) to extend [1] and coherently solve the SFM and the object detection problems (Fig. 1). We propose to model this correlation by learning the typical transformation that object appearance elements (features) undergo as objects in the scene are observed from different view points. We show that this transformation can be learned in a statistical sense for each object class without the need of explicitly determining the 3D shape of the objects in the scene. Quantitative and qualitative experimental results on two data sets demonstrate that by modeling such object-feature interactions our method obtains more accurate reconstruction and recognition results than if objects and features are assumed independent as in [1]. Our method works under the condition that camera internal parameters are known.

2. Problem Formulation

Rigid *structure from motion* (SFM) is defined as the problem of recovering the camera parameters $\mathbf{C}$ and scene's 3D structure $\mathbf{Q}$ from the observations in the input images. The observations are the observed locations and features of 2D image points $\mathbf{q}$. Following [1], we extend the SFM problem and explore a new problem called *semantic structure from motion* (SSFM). In SSFM the unknowns that we aim to estimate are the 3D feature points $\mathbf{Q}$, camera locations and poses $\mathbf{C}$, and high level elements $\mathbf{O}$ (object in the scene). The observations are the image features $\mathbf{q}$ and the observations of the objects in the image. Observations of the objects may be obtained using standard object detectors and are denoted by $\mathbf{o}$. The goal is to estimate the unknown parameters ($\mathbf{Q}$, $\mathbf{O}$ and $\mathbf{C}$) given the observations ($\mathbf{q}$ and $\mathbf{o}$). We formulate this estimation problem as likelihood maximization using the graphical model in figure 2.

2.1. Notations

Cameras C. Let $\mathbf{C}$ denote the camera parameters. $\mathbf{C} = \{C^k\} = \{K^k, R^k, T^k\}$ where K is the internal parameter, 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*.

3D Points Q. 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 *unknown*.

Image Point Observations $\mathbf{q}$. Denote by $\mathbf{q} = \{q_i^k\}$ the set of point *observations* (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 $q_i^k = (x_i^k, y_i^k, a_i^k, u_i^k)$, where (x_i^k, y_i^k) is image location, a_i^k the image feature, and u_i^k the index of corresponding 3D point of q_i^k. Point observations are obtained by a feature detector such as [28, 12, 16, 21]. We also assume that correspondences of features across views are estimated by feature matching algorithm [28, 16]. This allows us to determine the 3D points Q by triangulation and to establish values for the index u.

3D Objects O. Let $\mathbf{O} = \{O_t\}$ denote a set of 3D objects O_t. Each 3D objects O_t is specified by a 3D location (X_t, Y_t, Z_t), a pose $\Gamma_t = (\Theta_t, \Phi_t)$, and a category label c_t (e.g, car, bottle, etc...). (X_t, Y_t, Z_t) is the center of the 3D bounding cube tightly enclosing the object. The pose Γ_t is identified as a point on the unit viewing sphere [2, 1]. A 3D object is parametrized by $O_t = (X, Y, Z, \Gamma, c)_t$. The set $\mathbf{O}$ is *unknown* in our problem.

Image Objects Observations $\mathbf{o}$. Denote by $\mathbf{o} = \{o_j^k\}$ the set of object observations for all the cameras. o_j^k is the j^{th} observed object in image (camera) k. An object observations is described by the following measurement vector $o_j^k = \{x, y, w, h, \gamma, c, a, v, p\}_j^k$, where x, y are the object location in the image, h, w the object scale (bounding box height and width), $\gamma = (\theta, \phi)$ the object pose, a the image feature, and c the category label (e.g, car). These measurements can be obtained by any object detectors (e.g. [6]) that return the probability p that certain location (x, y) in an image is occupied by an object with category c, scale h, w, and pose ψ[1]. Each true detection o_j^k is assumed to correspond to certain physical 3D object O_t, and such correspondence is modeled by v_j^k. Let $v_j^k = t$ if o_j^k corresponds to 3D object O_t. Note that detection results are noisy, so $\mathbf{o}$ contains many object detection false positives. A false positive is not associated to any of the 3D objects and we model this by setting $v_j^k = 0$.

Indices. k super-indexes which camera a variable belongs to (e.g. C^k). s sub-indexes which 3D point a variable belongs to (e.g. Q_s). t sub-indexes which 3D object a variable belongs to (e.g. O_t). i sub-indexes which 2D point a variable belongs to (e.g. q_i^k). j sub-indexes which 2D object a variable belongs to (e.g. o_j^k).

2.2. Max-Likelihood Formulation

Following [1], we formulate the SSFM problem as a *max-likelihood estimation* (MLE) problem:

$$\{\mathbf{Q}, \mathbf{O}, \mathbf{C}\} = \arg \max_{Q,O,C} \Pr(\mathbf{q}, \mathbf{o} | \mathbf{Q}, \mathbf{O}, \mathbf{C}) \qquad (1)$$

In [1], $\Pr(\mathbf{q}, \mathbf{o} | \mathbf{Q}, \mathbf{O}, \mathbf{C})$ is decomposed into object terms and point terms separately by assuming that the objects and points are conditionally independent. One of the key contributions of our work is that we claim points and objects are correlated and this correlation should be explicitly accounted for. We propose a more general method of

[1] State of the art object detectors such as [6] can be modified so as to enable pose classification [1].

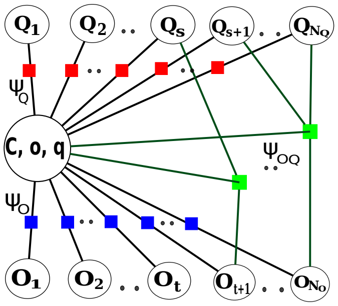


Figure 2: Factor Graph Representation.$\mathbf{C}, \mathbf{o}, \mathbf{q}$ are the camera parameters and image observations respectively. Q (top) are 3D points. O (bottom) are 3D objects. Ψ_Q (red) captures the camera-point relationship (conventional SFM). Ψ_Q (red) and Ψ_O (blue) captures the camera-point relationship and camera-object relationship separately ([1]). In this work we model the object-point relationship and formulate it as Ψ_{OQ} (green). The number of points correlated with each object is not fixed. For instance, object O_{N_o} is correlated with Q_{N_Q},Q_{s+1}, and O_{t+1} is correlated with Q_s.

decomposing the likelihood term using a factor graph formulation in Fig. 2:

$$\begin{aligned}\Pr(\mathbf{q}, \mathbf{o}|\mathbf{Q}, \mathbf{O}, \mathbf{C}) \quad = \quad & \frac{1}{Z}\prod_t \Psi_O(O_t; \mathbf{o}, \mathbf{C}) \prod_s \Psi_Q(Q_s; \mathbf{q}, \mathbf{C}) \\ & \prod_t \Psi_{OQ}(O_t, \mathbf{Q}_t; \mathbf{o}, \mathbf{q}, \mathbf{C}) \qquad (2)\end{aligned}$$

where Z is a normalization constant, Ψ_O captures the object likelihood, Ψ_Q captures the point likelihood, Ψ_{OQ} captures the object-point correlation likelihood. We will detail the computation of each likelihood terms in Sec. 2.3~2.5. Note that if the last correlation term is ignored (i.e. $\Pr(\mathbf{q}, \mathbf{o}|\mathbf{Q}, \mathbf{O}, \mathbf{C}) \propto \prod_t \Psi_O(O_t, \mathbf{o}, \mathbf{C}) \prod_s \Psi_Q(Q_s, \mathbf{q}, \mathbf{C})$), we will obtain exactly the same formulation in [1]. Furthermore, if the last two terms of Eq. 1 are ignored, Eq. 1 degenerates into $\{\mathbf{Q}, \mathbf{C}\} = \arg\max_{\mathbf{Q},\mathbf{C}} \prod_s \Psi_Q(Q_s, \mathbf{q}, \mathbf{C})$, which is equivalent to solving the conventional point-based rigid SFM problem. This suggests that if no objects exists in images, our model is still capable of solving the SFM problem in the same manner as conventional point-based SFM does.

2.3. Object Likelihood Ψ_O

The factor function $\Psi_O(O_t; \mathbf{o}, \mathbf{C})$ evaluates the likelihood (up to a normalization constant) of 3D object O_t given the camera hypothesis $\mathbf{C}$ and image object detection observations $\mathbf{o}$. We compute $\Psi_O(O_t; \mathbf{o}, \mathbf{C})$ by following two intuitions: i) the projections onto different images of the same 3D object should have similar appearance, ii) the probability of detecting an object in proximity to the projection of a 3D object to the image plane for each camera should be high. Thus, we decompose $\Psi_O(O_t; \mathbf{o}, \mathbf{C})$ as:

$$\Psi_O(O_t; \mathbf{o}, \mathbf{C}) = P_t^{det} \cdot P_t^{app} \qquad (3)$$

where P_t^{det} captures the object detection probability of the projection of O_t into each camera, and P_t^{app} captures the similarity of the image appearance of the projections of O_t across cameras. Denote by $\mathbf{o}_t = \{o_j^k | v_j^k = t\}$ the set of 2D projections of O_t in every images.

Computing P_t^{det}. We use object detectors to compute P_t^{det}. Denote by p_j^k the object detection probability of o_j^k. Suppose o_j^k is the 2D projection of O_t in image k. i.e. $v_j^k = t$. We propose to compute P_t^{det} as $P_t^{det} = (1 - \prod_k (1 - p_{j_k}^k))$ where $v_{j_k}^k = t$. This expression makes the computation of P_t^{det} less sensitive to configurations where objects are occluded or partially occluded from some of the views.

Computing P_t^{app}. P_t^{app} is computed by checking the similarity of projection appearances. If $o_{j_1}^1$ and $o_{j_2}^2$ are true projections of the same 3D object O_t, they should share similar image appearance, i.e. the appearance feature vectors $a_{j_1}^1$ and $a_{j_2}^2$ (Sec. 2.1) are close to each other. We propose to approximate P_t^{app} by considering appearance similarity for each camera pairs (and by assuming that these pairwise terms are independent): $P_t^{app} = \prod_{k_1 \neq k_2} \boldsymbol{N}(a_{j_{k_1}}^{k_1} - a_{j_{k_2}}^{k_2})$, where $\boldsymbol{N}(\cdot)$ is a Gaussian distribution whose mean is zero and covariance is learned from the training set by max-likelihood.

2.4. Point Likelihood Ψ_Q

The factor function $\Psi_Q(Q_s; \mathbf{q}, \mathbf{C})$ evaluates the likelihood (up to a normalization constant) of 3D point Q_s given the camera hypothesis $\mathbf{C}$ and image point detection and matching $\mathbf{q}$. We compute $\Psi_Q(Q_s; \mathbf{q}, \mathbf{C})$ by following two intuitions: i) the probability of detecting a feature point in proximity of the projection of Q_s into the image plane should be high, ii) the projections of Q_s into different images should have similar features (appearance). Thus, we decompose $\Psi_Q(Q_s; q, \mathbf{C})$ as:

$$\Psi_Q(Q_s; \mathbf{q}, \mathbf{C}) = P_s^{point} \cdot P_s^{feat} \qquad (4)$$

where P_s^{point} captures the location agreement between the projection of Q_s and its corresponding image point, and P_s^{feat} captures the feature (appearance) similarity of the corresponding points of Q_s across camera images. Since each 3D point Q_s is calculated by the triangulation of matched 2D points, the correspondence of 2D point and 3D point is established and fixed after the 2D feature point matching. Denote by q_s^k the projection of Q_s into image k, and $q_{i_k}^k$ the corresponding observation of Q_s.

Computing P_s^{point}. We assume that q_s^k and q_i^k overlap up to a zero-mean noise. Thus, we propose to compute $P_s^{point} = \prod_k \boldsymbol{N}(q_{i_k}^k - q_s^k)$ where again $\boldsymbol{N}(\cdot)$ is a Gaussian distribution whose mean is zero and covariance is learned from the training set by max-likelihood.

ComputingP_s^{feat}. We assume those feature vectors are equal up to a zero-mean noise: $P_s^{feat} = \prod_{k_1 \neq k_2} \boldsymbol{N}(a_{i_{k_1}}^{k_1} - a_{i_{k_2}}^{k_2})$, where $\boldsymbol{N}(\cdot)$ is a Gaussian distribution whose mean is zero and covariance is learned from the training set by max-likelihood.

2.5. Object-Point Correlation Likelihood Ψ_{OQ}

Before we introduce the object-point correlation likelihood $\Psi_{O,Q}$, we first define the *correlation* between object and point. If a 3D point Q_s lies on the surface of a 3D object O_t, the corresponding feature and object projections in each camera are said to be correlated. Denote by $\mathbf{Q}_t$ the set of 3D points that lie on O_t. The factor function $\Psi_{OQ}(O_t, \mathbf{Q}_t; \mathbf{o}, \mathbf{q}, \mathbf{C})$ captures how well the correlation between object O_t and points $\mathbf{Q}_t$ is supported by the image observations (i.e. $\mathbf{o}, \mathbf{q}$) and camera parameters (i.e. $\mathbf{C}$). We will discuss the details of $\Psi_{O,Q}$ in the Sec. 3.

2.6. Inference with Monte Carlo Markov Chain

The estimation of camera parameters, points, and objects is equivalent to maximizing Eq. 1. Due to the high dimension of the parameter space, we sample $\mathbf{C}, \mathbf{Q}, \mathbf{O}$ from $\Pr(\mathbf{q}, \mathbf{o}|\mathbf{Q}, \mathbf{C}, \mathbf{O})$ using Monte Carlo Markov Chain (MCMC) similarly to [1]. The proposal distribution of this MCMC process is $\Pr(\mathbf{q}, \mathbf{o}|\mathbf{Q}, \mathbf{C}, \mathbf{O})$. A key to MCMC is the procedure for initializing the sampling process. Here we adopt the same approach as detailed in [1]. For camera pose, we have two sources of initializations: 1) feature point based SFM result, 2) roughly estimation of camera pose from object detection pose and scale in different images. With different camera pose initializations, we obtain different 3D object initializations by projecting 2D object detections into 3D according to their 2D scales and poses [2]. For different initializations, MCMC provides different samples. We combine all the samples to find the maximum value of the probability. The single sample that maximizes the likelihood $\Pr(\mathbf{q}, \mathbf{o}|\mathbf{Q}, \mathbf{C}, \mathbf{O})$ is taken as the solution of the final estimation of $\mathbf{C}, \mathbf{O}, \mathbf{Q}$.

3. Modeling the Object-Point Correlation $\Psi_{O,Q}$

3.1. Identifying Correlation From Images

The concept of *correlation* among features and object observations is associated with the physical 3D phenomenon of having features points to lie on 3D objects. Since the inputs of our algorithm are only 2D images, we identify the correlation of 3D points and 3D objects from images. Denote by o_t^k the projection of O_t in image k. Suppose Q_s lies on O_t. The corresponding 2D feature point observation of Q_s must fall into the 2D bounding box of o_t^k. Notice that, however, not all the 2D points within the bounding box of o_t^k are correlated with O_t – e.g. the bounding box of o_t^k may include features from the background which are not part of the object foreground O_t. Moreover, features points that are visible in one view may not be visible in another view because of 3D object self-occlusions (Fig. 3a).

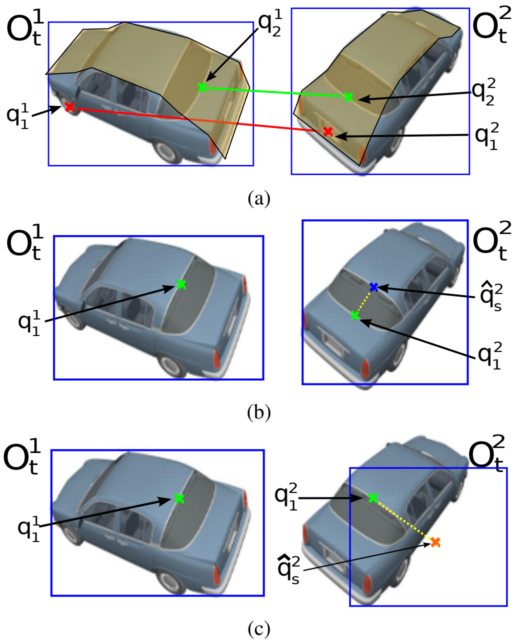


Figure 3: Fig. 3a: Given object category and poses, L predicts a pair of decision regions (yellow polygons in top figure) that may contain consistent feature correspondences. If a pair of points both fall into these regions (e.g. q_2^1 and q_2^2), it is likely to be true match (so that $L = true$). Otherwise, it (e.g. q_1^1,q_1^2) is likely to be a false match (so that $L = false$). Fig. 3b: U predicts the matched point of q_1^1 to be $\hat{q}_s^2$. The yellow dashed line indicates the distance d between $\hat{q}_s^2$ and q_1^1's matched point $q_{i_2}^1$. d should be zeros ideally. We use d to evaluate the likelihood that whether or not a pair of matched points (e.g. q_1^1 and q_1^2) is correct. Fig. 3c: If the object detection hypothesis is wrong (right blue rectangle), the prediction of U will be deviated from the matched point. Although the point match is correct, the correlation likelihood will also drop due to the incorrect object detection.

We introduce a function $L = \{true, false\}$ to estimate whether or not two points (in different images) are correlated with an object. L returns "true" if a pair of matched feature points $q_{i_1}^1, q_{i_2}^2$ correspond to two observations of the same 3D point Q_s lying on the 3D physical object O_t (points and objects are correlated). L returns "false" otherwise. It is clear that this prediction can be made if the 3D shape of the object is available and the camera geometry is given (by back-projection). In practice, however, the actual 3D shape is unknown (we only have an hypothesis of the object category label) and recovering it goes beyond the scope of this paper.

We propose to learn this mapping L implicitly from a set of training data. Specifically, L is modeled as a classifier which is learned to associate input data points (i.e., the locations of the matched features in each image pair) to a "true" or "false" label. This association is learned for each (discretized) object pose and object class label. This association can be made *quasi* independent from the camera configurations by normalizing the feature coordinates with respect to the detected object bounding box. The classifier will give "true" labels if matched features are geometrically consistent with the object class and the object pose transformation; whereas it will associate "false" labels to "inconsistent" configurations which typically stems from self-occlusions or background regions. The classifier L can be expressed as follows: $L_{\gamma_t^1,\gamma_t^2,c}(\frac{x_{i_1}^1-x_t^1}{w_t^1}, \frac{y_{i_1}^1-y_t^1}{h_t^1}, \frac{x_{i_2}^2-x_t^2}{w_t^2}, \frac{y_{i_2}^2-y_t^2}{h_t^2}) = \{true, false\}$ where the coordinates $\frac{x_{i_1}^1-x_t^1}{w_t^1}, \frac{y_{i_1}^1-y_t^1}{h_t^1}, \frac{x_{i_2}^2-x_t^2}{w_t^2}, \frac{y_{i_2}^2-y_t^2}{h_t^2}$ are normalized with respect to the detected object bounding box (whose size scale are w_t and h_t) in each image, the variables associated with the 2D object projection are sub-indexed by t, and L's sub-indexes γ_t^1,γ_t^2,c capture the dependency of L on the object poses (γ_t^1,γ_t^2) and class labels (c). We implement L using a non-linear SVM. The classifier predicts a pair of decision regions (in the image plane) that may contain consistent feature correspondences (i.e. those labeled as "true"). As Fig. 3a shows, regions associated to a "true" label are highlighted in yellow. If $L_{\gamma_t^1,\gamma_t^2,c}(\frac{x_{i_1}^1-x_t^1}{w_t^1}, \frac{y_{i_1}^1-y_t^1}{h_t^1}, \frac{x_{i_2}^2-x_t^2}{w_t^2}, \frac{y_{i_2}^2-y_t^2}{h_t^2}) = false$, the match between $q_{i_1}^1$ and $q_{i_2}^2$ will result in $\Psi_{O,Q} = 0$. To simplify the notation, we use $L(q_{i_{k_1}}^{k_1}, q_{i_{k_2}}^{k_2})$ to denote the function L.

Learning L. We learn from the training set the parameters of L (i.e the coefficient of the SVM classifier). In our training set, images and corresponding depth maps are available. From these we can easily obtain ground truth feature matches on 3D objects across view points. We learn L for different pose pairs and for different object classes.

3.2. Evaluating Correlated Points

While L indicates if feature matches are geometrically consistent with the object class and object pose transformation (via binary classification), it does not measure the degree of this consistency. In order to do so, we introduce a function U which is capable of predicting the location of matched feature points (lying on the object O_t) across views given the object 2D projections in two images, and the 2D image location of one of the two matched points. This prediction can be made deterministic given the object 3D shape and camera configurations. As discussed earlier, however, we assume the object 3D shape is not available and we rather aim at learning this prediction using a training set. Similarly to L, by normalizing the feature coordinates with respect to the detected object bounding box, we can make U function of the object class and pose transformation only and express U as:

$$U_{\gamma_t^1,\gamma_t^2,c}(\frac{(x_{i_1}^1 - x_t^1)}{w_t^1}, \frac{(y_{i_1}^1 - y_t^1)}{h_t^1}) = (\frac{\hat{x_i^2} - x_t^2}{w_t^2}, \frac{\hat{y_i^2} - y_t^2}{h_t^2})^T$$

where γ_t^1, γ_t^2 are 2D object poses, x_t^1,y_t^1,x_t^2,y_t^2 are 2D object projections, w_t^1,h_t^1,w_t^2,h_t^2 are 2D object projection scales, $x_{i_1}^1$,$y_{i_1}^1$ are the observed point in the 1^{st} image, and $\hat{x}_i^2$,$\hat{y}_i^2$ are the predicted image point in the 2^{nd} image. In order to limit the number of parameters required to learn U, we use low-order Taylor series to approximate U. This makes the learning tractable and the prediction more robust to data noise. Assume the exact prediction based on 3D information has Taylor series expansion $\xi_{\gamma^1,\gamma^2,c}$ (2 by ∞ matrix) so that $U_{\gamma^1,\gamma^2,c}(x,y) = \xi_{\psi^1,\psi^2,c} \cdot (1,x,y,xy,x^2,y^2...)^T$. We use the quadratic terms to approximate U, i.e. $U_{\gamma^1,\gamma^2,c}(x,y) \approx \alpha_{\gamma^1,\gamma^2,c} \cdot (1,x,y,xy,x^2,y^2)^T$, where the matrix α is the first several columns of ξ. To simplify the notation, we use $U(q_{i_{k_1}}^{k_1})$ to denote the function U that predicts the location of $\hat{q}_s^{k_2}$ (i.e. the matched point of $q_{i_{k_1}}^{k_1}$ in image k_2).

Learning U. We perform linear regression to learn the parameters of U (i.e. α) for different object pose pairs and categories.

Using U. U predicts the matched point. We use the distance between this prediction and image observation to evaluate their consistency (as a part of computing $\Psi_{O,Q}$). Define $d_{i_{k_1},i_{k_2}}^{k_1,k_2}$ as the distance between the observation $q_{i_{k_2}}^{k_2}$ (matched point of $q_{i_{k_1}}^{k_1}$ by point feature) and the prediction $U(q_{i_{k_1}}^{k_1})$.

3.3. Computing $\Psi_{O,Q}$ with L and U

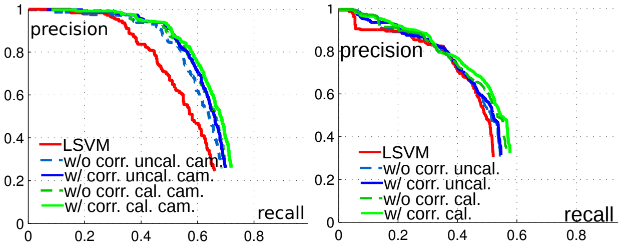
We compute $\Psi_{O,Q}$ under two assumptions: i) given the cameras and 3D objects, 3D points correlated to the same objects are independent to each other, ii) joint correlation likelihood term can be decomposed into pair-wise terms. Hence, $\Psi_{OQ}(O_t, \mathbf{Q}_t; \mathbf{o}, \mathbf{q}, \mathbf{C})$ is decomposed:

$$\begin{aligned} & \Psi_{OQ}(O_t, \mathbf{Q}_t; \mathbf{o}, \mathbf{q}, \mathbf{C}) \\ = \; & \Pi_{Q_s \in \mathbf{Q}_t} \Psi_{O,Q}(\mathbf{q}_s, \mathbf{o}_t; O_t, Q_s, \mathbf{C}) \\ = \; & \prod_{Q_s \in \mathbf{Q}_t} \prod_{u_{j_1}^{k_1} = u_{j_2}^{k_2} = s} \Psi_{O,Q}(q_{j_1}^{k_1}, q_{j_2}^{k_2}, o_t^1, o_t^2; C^{k_1}, C^{k_2}) \end{aligned} \qquad (5)$$

where again $\mathbf{q}_s$ is the projection of Q_s on different images, and $u_{j_1}^{k_1} = u_{j_2}^{k_2} = s$ indicates $q_{j_1}^{k_1}, q_{j_2}^{k_2}$ both correspond to Q_s. Assuming that $d_{i_{k_1},i_{k_2}}^{k_1,k_2}$ satisfies a 2D zero-mean Gaussian distribution, the object-point correlation likelihood can be computed as :

$$\begin{aligned} & \Psi_{O,Q}(q_{j_1}^{k_1}, q_{j_2}^{k_2}, o_t^1, o_t^2; C^{k_1}, C^{k_2}) \\ = & \begin{cases} 0 & \text{if } L(q_{i_{k_1}}^{k_1}, q_{i_{k_2}}^{k_2}) = false \\ N(d_{i_{k_1},i_{k_2}}^{k_1,k_2}) & \text{o.w.} \end{cases} \end{aligned} \qquad (6)$$

where $N(\cdot)$ is a Gaussian density function whose mean is zero and covariance matrix is learned from the training set in a max-likelihood manner.



(a) Ford Car dataset. Average-precision: LSVM 54.5%; w/o corr. uncal. cam. 61.3%; w/ corr. uncal. cam. 62.2%; w/o corr. cal. cam. 62.1%; w/ corr. cal. cam. 62.8%.

(b) Kinect Office. Average-precision: LSVM 42.9%; w/o corr. uncal.45.0%; w/ corr. uncal. cam. 45.3%; w/o corr. cal. cam. 45.7%; w/ corr. cal. cam. 46.4%.

Figure 4: Object detection precision-recall. "cal". stands for calibrated; "uncal." stands for uncalibrated; "corr." stands for object-point correlation. LSVM is our baseline object detection methods [6].

4. Evaluation

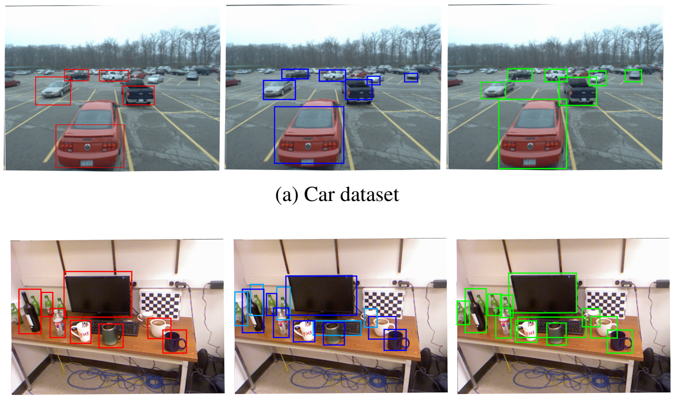
We evaluated our method on two data sets: Ford Car (Sec. 4.1) and Kinect Office (Sec. 4.2). We benchmark our method with the state-of-the-art baseline detector Latent SVM [6] and point-based SFM approach Bundler [25]. When we evaluate the 2D or 3D object detection performance of our framework, we test it with two configurations: with or without known camera pose (external parameters). We show if the camera pose is given, our method will detect objects more accurately than if camera pose is unknown. In such case, camera pose is estimated by our algorithm.

We also test our method with and without the object-point correlation, i.e. enabling or disabling the Ψ_{OQ} term when we compute the joint likelihood in Eq. 1. If we disable the Ψ_{OQ} term our algorithm generates results similar to [1]. The typical running time for one image pair with Matlab single-thread implementation is ~10 minutes.

2D Object Detection. Since our algorithm jointly formulates object detections in multiple images, we can: i) recover missed positives by using the object detections from other images, ii) remove the false positives by checking the consistency of object detection's pose and scale across multiple images. We evaluate the object detection performance and show precision-recall (PR) in Fig. 4 and Tab. 1. We compare our algorithm with a baseline LSVM [6]. Our detection performance is computed by projecting the estimated 3D object bounding cube into each image and measuring the overlap ratio (>50%) between such projection and ground truth bounding box. Thus, PR obtained by LSVM is computed for all single images for fair comparison. Fig. 5 shows the examples of the detection result, which visualizes the benefit of using object-point correlations as opposed to the case when objects and points are independent in [1]. More anecdotal results are shown in Fig. 7c,7d.

Category	Car	Monitor	Keyboard	Mouse	Bottle	Mug
LSVM[6]	0.544	0.900	0.177	0.101	0.361	0.476
Ours Uncali. Cam.	0.622	0.903	0.201	0.096	0.376	0.508
Ours Cali. Cam.	0.628	0.903	0.220	0.102	0.385	0.521

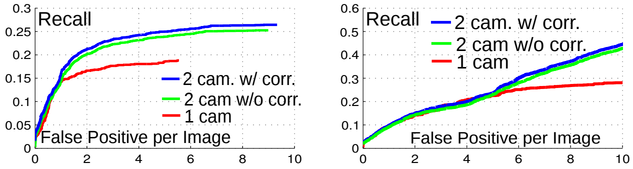
Table 1: 2D object detection average-precision.



(a) Car dataset

(b) Office dataset

Figure 5: Comparison among LSVM, [1], and this paper. The first column shows the detection results by LSVM. The second column shows the detection results by [1] where no object/point correlation is used. Notice that some of the bounding boxes do not fit the objects well (cyan). The third column shows the results by using SSFM with object/point correlation (this paper). The detected bounding boxes are more precise than [1] .



(a) Car Dataset. $\eth = 2.5m$

(b) Office Dataset. $\eth = 0.1m$

Figure 6: 3D Object Detection Precision v.s. False Positives per Image.

3D Object Detection. Compared to methods that only use a single image for scene layout estimation [2], our algorithm achieves better performances in localizing objects in 3D. This is because we can better handle the object scale variation by estimating the object location and size from multiple images. 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 the center of its bounding cube and center of the cube of a ground truth 3D object is smaller than a threshold $\eth$ (see Fig. 6). Due to the metric-reconstruction ambiguity, we use calibrated cameras in this experiment to enforce that the 3D objects coordinates are defined up a similarity transformation. The 3D object localization for one camera is obtained by using its 2D bounding box scale and location [13]. Results are shown in figure 6.

Camera Pose Estimation. We show that the ability of our method to model object and point correlations leads to

	Ours with / without corr.	SFM
Car $\bar{e}_T$	$\mathbf{15.3}^\circ$/19.9°	26.5°
Car $\bar{e}_R$	$< 0.1^\circ$/$< 0.1^\circ$	$< 0.1^\circ$
Office $\bar{e}_T$	$\mathbf{4.4}^\circ$/4.7°	8.5°
Office $\bar{e}_R$	$\mathbf{3.7}^\circ$/4.1°	9.5°

Table 2: Camera Pose Estimation. We report SFM result with the implementation provided by [25]. The numbers shown are the average value for all our testing image pairs.

more accurate camera pose estimation results. We compare our method with the state-of-the-art point-based SFM approach Bundler [25]. Bundler employs the SIFT feature, five-points algorithm [17], and Bundle Adjustment [27] to estimate the camera pose. 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 [17]. Ground truth depth and camera parameters are known for both data sets (see Sec. 4.1 and 4.2). 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. Results are shown in Tab. 2.

4.1. Ford Car dataset

The Ford Campus Vision dataset [19] consists of 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 [23]. This training set consists of 8 poses. Some typical experimental results are shown as Fig. 7a,7b.

4.2. Kinect Office dataset

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. Some typical experimental results are shown as Fig. 7c,7d.

5. Conclusion

We have proposed a new approach for jointly detecting objects and recovering the geometry of the scene from two or multiple un-calibrated images of the scene. Compared to previous contributions where high level features (e.g. objects) and low level features (e.g. points) are processed independently, our method explicitly models the correlation between objects and points across views. Quantitative and qualitative experimental results have shown that this additional piece of information is critical for improving both detection and reconstruction accuracy.

Acknowledgment

We acknowledge the support of NSF CAREER #1054127 and the Gigascale Systems Research Center.

References

[1] S. Y. Bao and S. Savarese. Semantic structure from motion. In *CVPR*, 2011. 1, 2, 3, 4, 6

[2] S. Y. Bao, M. Sun, and S. Savarese. Toward coherent object detection and scene layout understanding. In *CVPR*, 2010. 1, 2, 4, 6

[3] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. *ECCV*, 2008. 1

[4] N. Cornelis, B. Leibe, K. Cornelis, and L. Gool. 3d urban scene modeling integrating recognition and reconstruction. *IJCV*, 2008. 1

[5] A. Dick, P. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *IJCV*, 60(2):111–134, 2004. 1

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2009. 1, 2, 6

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, pages 264–271, 2003. 1

[8] A. Frome, D. Huber, R. Kolluri, T. BÃŒlow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, 2008. 1

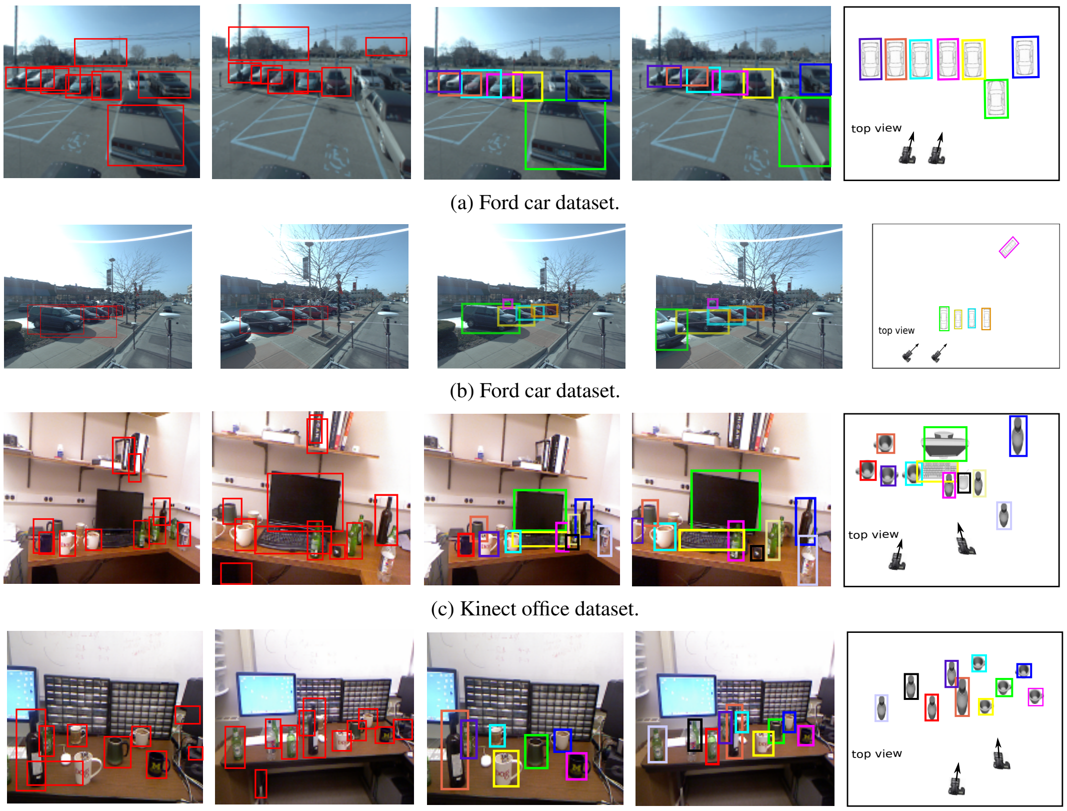
[9] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *CVPR*, pages 1–8. IEEE, 2009. 1

[10] A. Gupta, A. Efros, and M. Hebert. Blocks World Revisited: Image Understanding using Qualitative Geometry and Mechanics. *ECCV*, 2010. 1

[11] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. 2003. 1

[12] B. Herbert, T. Tinne, and V. Luc. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 2

[13] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 80(1):3–15, 2008. 1, 6



(a) Ford car dataset.

(b) Ford car dataset.

(c) Kinect office dataset.

(d) Kinect office dataset.

Figure 7: Result Examples. Column 1 and 2: Baseline object detection in two images. There quite a few missed positives and false alarms due to the occlusion and variation of view point. Column 3,4: the final joint object detections projected in the 1st and 2nd image. SSFM recovers missed positives in one view by using the object detection in other views. SSFM also reduces the probabilities of false alarms that have no correspondence in other views. Column 5: the top view of the scene, which shows the reasoning result of SSFM in 3D space. Colors in the last three columns show the object correspondences established by SSFM.

[14] D. Huber. Automatic 3d modeling using range images obtained from unknown viewpoints. In *Int. Conf. on 3-D Digital Imaging and Modeling*, 2001. 1

[15] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *In ECCV workshop on statistical learning in computer vision*, pages 17–32, 2004. 1

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004. 2

[17] D. Nister. An efficient solution to the five-point relative pose problem. *IJCV*, 2004. 7

[18] M. Ozuysal, V. Lepetit, and P.Fua. Pose estimation for category specific multiview object localization. In *CVPR*, Miami, FL, June 2009. 1

[19] G. Pandey, J. R. McBride, and R. M. Eustice. Ford campus vision and lidar data set. *International Journal of Robotics Research*, 2011. In Press. 7

[20] M. Pollefeys and L. Gool. From images to 3D models. *Communications of the ACM*, 45(7):50–55, 2002. 1

[21] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *PAMI*, 32:105–119, 2010. 2

[22] 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*, 2008. 1

[23] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007. 7

[24] A. Selvatici, A. R. Costa, and F. Dellaert. Object-based visual slam: How object identity informs geometry. In *IV Workshop de Visao Computacional Bauru Brazil*, 2008. 1

[25] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2):189–210, 2008. 1, 6, 7

[26] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Depth from familiar objects: A hierarchical model for 3D scenes. In *CVPR*, volume 2, pages 2410–2417. IEEE, 2006. 1

[27] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, 2000. 7

[28] T. Tuytelaars and L. V. Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC*, 2000. 2