Semantic Structure From Motion with Points, Regions, and Objects

Sid Yingze Bao, Mohit Bagra, Yu-Wei Chao, Silvio Savarese
Department of Electrical and Computer Engineering, University of Michigan at Ann Arbor
{yingze,mohitbag,ywchao,silvio}@eecs.umich.edu

Abstract

Structure from motion (SFM) aims at jointly recovering the structure of a scene as a collection of 3D points and estimating the camera poses from a number of input images. In this paper we generalize this concept: not only do we want to recover 3D points, but also recognize and estimate the location of high level semantic scene components such as regions and objects in 3D. As a key ingredient for this joint inference problem, we seek to model various types of interactions between scene components. Such interactions help regularize our solution and obtain more accurate results than solving these problems in isolation. Experiments on public datasets demonstrate that: 1) our framework estimates camera poses more robustly than SFM algorithms that use points only; 2) our framework is capable of accurately estimating pose and location of objects, regions, and points in the 3D scene; 3) our framework recognizes objects and regions more accurately than state-of-the-art single image recognition methods.

1. Introduction

One core problem in computer vision is to recover the structure of a scene and estimate the pose and location of the observer from images. Typical algorithms that address this problem are called *structure from motion* (SFM). Most SFM algorithms [29, 28] represent the structure of a scene as a set of 3D points. However, in many applications such as robotic manipulation and autonomous navigation, a point-based representation is not sufficient and both structure and semantic information are required. Given a scene as in Fig. 1, one wishes to answer questions such as: is there a cup, a table, or a bottle? Where are they in 3D? What are their poses? Unfortunately, most SFM algorithms cannot provide an answer to these questions.

In this paper, we propose a novel framework to jointly recover the structure of the scene and estimate the pose of cameras from a few input images (Fig. 1). By "recovering the structure of the scene", we mean recognizing and estimating the location of points, as well as location and pose of objects (e.g. bottles, cups, and monitors), and regions (e.g. table, wall, and road) in the scene. There are two main contributions in our framework.

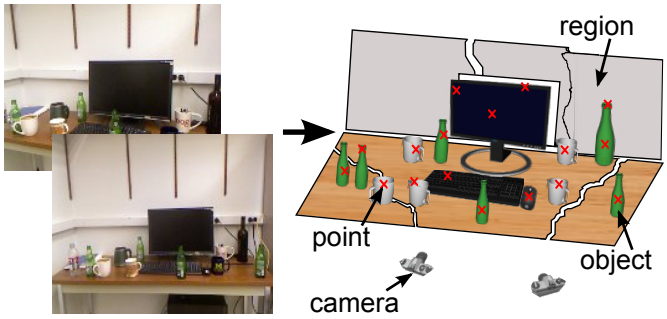


Figure 1: Our goal is to recognize semantic elements (e.g. cups, bottles, desk, wall, etc), localize them in 3D, and estimate camera pose from a number of semi-calibrated images. We propose to achieve this goal by modeling interactions among 3D points, regions, and objects.

First, we propose to represent a scene as a set of points, objects and regions, whereas SFM algorithms usually represent the scene just as a set of points. Points, regions, and objects have different and complementary properties (See Tab. 1). For instance, points do not carry semantic information, but can be robustly matched across views (e.g. by [22]). However, unless local affine information is available [23], a large number of point correspondences are required to robustly estimate the camera pose. Regions (e.g. a portion of road) carry weak semantic information, and can be used to impose stronger geometrical constraints for estimating camera poses, but are harder to match across views than points. Objects carry rich semantic information, and can be used to impose even stronger geometrical constraints for estimating camera poses (if object's pose and scale are estimated), but are even more difficult to match across views than regions (due to self occlusions, etc). By jointly modeling points, regions, and objects, we leverage all these properties and seek to take advantage of the best of each of them.

Second, we propose to take advantage of *interactions* among points, objects, and regions to help regularize the estimation of the unknown parameters. An *interaction* mod-

	Degree of semantics	Matching robustness	Geometrical richness
Points	Low	High	Low
Regions	Medium	Medium	Medium
Objects	High	Low	High

Table 1: Properties of points, regions, and objects. Degree of semantics: degree of semantic information that a component carries. Matching robustness: degree of robustness in matching the same component across views. Geometrical richness: the number of geometrical constraints that a component matched across views can impose for estimating camera poses.

els the relationship between pairs of *scene components* in terms of location, pose, and semantics. We refer to points, regions, and objects as *scene component*s. Fig. 5 and Sec. 5 discuss in details the types of interactions we model in this work. Experimental results demonstrate that modeling the interactions among scene components is critical for robustly detecting and localizing them in 3D.

Our framework shows advantages in three aspects: 1) the estimation of the camera poses is more accurate than SFM algorithms that only use points; 2) it is capable of estimating the objects, regions, and points in a 3D scene whereas most SFM algorithms only estimate points; 3) the recognition and localization of objects and regions is more accurate than recognition methods that work on a single image. The reason for an increase in region classification accuracy is two-fold. First, our framework associates the observations of the same physical region from different views, which enables us to integrate appearance information across views. Second, our framework estimates the 3D geometry (location and orientation) of a region, and therefore it can leverage both appearance and geometry information to assign a semantic label to it (e.g. a table is usually a "horizontal" surface with "wooden" texture).

We designed and executed different experiments to test our theoretical claims in three publicly available datasets. Our framework consistently shows better performance than alternative state-of-the-art algorithms.

2. Related Works

There are several recent works addressing the problem of joint geometry estimation and semantic understanding. Among these works (e.g. [12, 15, 11, 25, 10, 20, 6, 13]), most assume that only a single image is available, whereas a handful of them model this problem with multiple images. Ladicky et al. [18] show promising results on joint reconstruction and segmentation from stereo pairs, but it works under the assumption of calibrated cameras with small baselines. Cornelis et al. [7] also assume a small-baseline calibrated stereo system, and it makes assumptions on the camera trajectories. Bao and Savarese [3] have recently introduced a formulation to integrate object recognition and SFM. The key idea of [3] is to use object recognition to guide the estimation of camera poses and simultaneously use the estimated scene geometry to help object recognition. However, [3] has a few limitations: 1) it focuses on modeling "structured" objects (e.g. cars, cups, and bottles), and it ignores "amorphous" scene components (e.g. road, sky, and table surfaces which can be hardly identified by standard object detectors such as [8, 21]). 2) it assumes that 3D points and 3D objects are independent given camera poses. A recent extension of [3] is presented in [2] wherein the idea of capturing the relationship between points and objects is explored. Compared to [3] and [2], we present a novel framework that coherently integrates regions and their interactions with objects and points. Finally, the idea of using scene component interactions to help estimate the scene layout is also proposed in several other works, but most of them only use limited types of interactions. For example, [27] uses point-region interaction, [19] uses point-object interaction, and [12, 15, 4] use object-region interaction. Our framework incorporates all types of interactions (point-region, point-object, object-region) and jointly use them to improve the scene layout estimation accuracy.

3. Framework Overview

In this section, we first introduce the unknowns and measurements in our framework. We next introduce our proposed framework and cast our problem as an energy maximization problem. We conclude this section with the inference algorithm for solving the maximization problem.

3.1. Measurements and Unknowns

We summarize the key notation in Tab. 2.

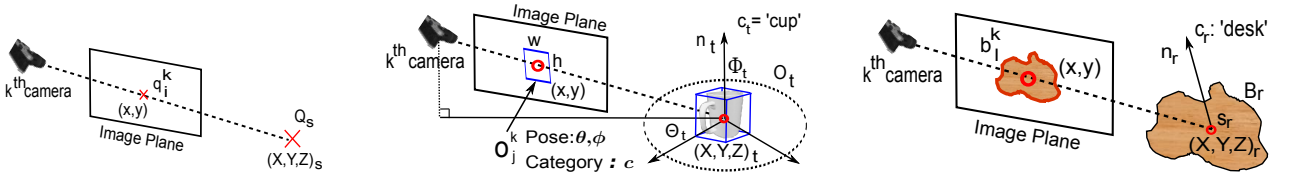
Images. Our inputs are a set of *images* $\mathbf{I} = \{I^1 \cdots I^k \cdots I^{N_c}\}$, where I^k is the k^{th} input image

Cameras. A *camera* is described by its internal parameter (known), rotation matrix (unknown), and translation vector (unknown) w.r.t world reference system. Let C^k be the k^{th} camera that captures image I^k.

Points (Fig. 2a)**.** The *point measurements* are detected interest points (e.g. by detectors such as [22, 30]). Denote by q_i^k the i^{th} interest point in image I^k. Let Q_s be the s^{th} 3D point within the scene. The correspondence between Q_s and point measurements is denoted by u_s, where $u_s = \{i^1, i^2, \cdots\}$ if Q_s corresponds to $q_{i^1}^1, q_{i^2}^2, \cdots$ respectively. Q_s and u_s are unknown.

Objects (Fig. 2b)**.** The *object measurements* are detected 2D objects (e.g. by detectors such as [8, 21]). Denote by o_j^k the j^{th} detected object in image I^k. Let O_t be the t^{th} 3D object within the scene. The correspondence between O_t and object measurements is denoted by v_t, where $v_t = \{j^1, j^2, \cdots\}$ if O_t corresponds to $o_{j^1}^1, o_{j^2}^2, \cdots$ respectively. O_t and v_t are unknown.

Regions (Fig. 2c)**.** The *region measurements* are segmented regions (Sec. 4.3.1). (e.g. by segmentation algorithms such as [26, 33]). We match segmented regions



(a) A point measurement q_i^k is described by its location (x, y). A 3D point (unknown) Q_s is described by its 3D location $(X, Y, Z)_s$ in the world reference system.

(b) An object measurement o_j^k is described by its location (x, y), bounding box scale h, w, pose θ, ϕ, and category label c. A 3D object (unknown) O_t is described by 3D centroid $(X, Y, Z)_t$, normal direction n_t, pose (Θ_t, Φ_t), and category label c_t (e.g, cup or bottle).

(c) A region measurement b_l^k is described by its boundary M_l^k (red boundary in image plane) and appearance descriptor a_l^k (e.g. color histogram and texture). A 3D region (unknown) B_r is described by 3D centroid $(X, Y, Z)_r$, normal direction n_r, area s_r, and category label c_r (e.g, sky, road, and wall).

Figure 2: Parametrization for the measurements and unknowns for points (a), objects (b), and regions (c).

across views based on their appearance and epipolar constraints. If a region is matched across views, we further estimate its location and orientation in 3D and assign it a semantic label (Sec. 4.3.1). Denote by b_l^k the l^{th} region in image I^k. Let B_r be the r^{th} 3D region within the scene. We assume that 3D regions are planar. The correspondence between B_r and regions measurements is denoted by g_r, where $g_r = \{l^1, l^2, \cdots\}$ if B_r corresponds to 2D regions $b_{l^1}^1, b_{l^2}^2, \cdots$ respectively. B_r and g_r are unknown.

Objects v.s. Regions. Two properties differentiate objects from regions. The first property is 3D volume. An object occupies a certain 3D volume and can be bounded by a 3D cube (Fig. 2b); conversely a region is a planar portion of surface that does not have 3D volume. The second property is whether or not the 3D location is predictable from one single image. The 3D location of an object can be (roughly) predicted from one detected 2D object in an image, if prior knowledge about the typical 3D object size is available [4]. The 3D location of a region cannot be predicted from only one image, since a region (e.g. a piece of sky) does not have "typical" size. Examples of objects include cars, bottles, persons. Examples of regions include the surface of a road or the sky.

3.2. Energy Maximization Framework

Given a set of input images, we seek to: i) recognize objects and classify regions; ii) estimate 3D locations and poses of points, regions, and objects; iii) estimate camera extrinsic parameters. Our framework follows two intuitions.

Intuition #1. The image projection of estimated 3D objects, regions, and points should be consistent with their image measurements (Fig. 4). Such consistency is measured w.r.t. location, scale, and pose.

Intuition #2. The interactions among estimated scene components should be consistent with the interactions learnt from the training set (Fig. 5).

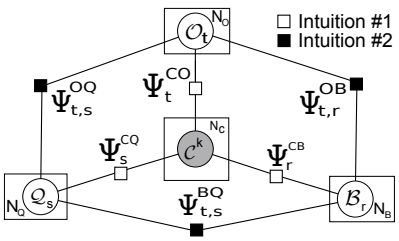


Figure 3: Factor node graph. The camera node $C^k = \{C^k, I^k\}$ is partially observed because the rotation and translation of C^k are unknown and I^k are input images.

Following these intuitions, we capture the relationship between measurements and unknowns by the factor graph in Fig. 3. The nodes Ψ_t^{CO}, Ψ_s^{CQ}, Ψ_r^{CB} are constructed following the intuition #1, and they evaluate how likely each single object, point, or region is consistent with 2D image measurements (Sec. 4). The nodes $\Psi_{t,r}^{OB}$, $\Psi_{t,s}^{OQ}$, $\Psi_{r,s}^{BQ}$ are constructed following the intuition #2, and they evaluate how likely the interaction among scene components (objects, points, or regions) are consistent with the learnt ones (Sec. 5). We formulate this estimation problem as the one of maximizing the energy:

$$\{\mathbb{Q}, \mathbb{O}, \mathbb{B}, \mathbf{C}\} = \arg\max_{\mathbb{Q},\mathbb{O},\mathbb{B},\mathbf{C}} \Psi(\mathbb{Q}, \mathbb{O}, \mathbb{B}, \mathbf{C}; \mathbf{I})$$

$$= \arg\max_{\mathbb{Q},\mathbb{O},\mathbb{B},\mathbf{C}} \prod_s \Psi_s^{CQ} \prod_t \Psi_t^{CO} \prod_r \Psi_r^{CB} \prod_{t,s} \Psi_{t,s}^{OQ} \prod_{t,r} \Psi_{t,r}^{OB} \prod_{r,s} \Psi_{r,s}^{BQ} \quad (1)$$

The definitions of $\mathbb{Q}, \mathbb{O}, \mathbb{B}, \mathbf{C}$ are in Tab. 2.

3.3. Solving the Estimation Problem

We use simulated annealing [16] to search for the solution for Eq. 1. Algorithm 1 summarizes the simulated annealing procedure for sampling the parameter space. Among all the samples in the parameter space, we identify the maximum. The parameters associated with the maximum sample are our final solution for Eq. 1, which correspond to the estimated cameras, points, objects, and regions.

4. Scene Components and Energy

One of our objectives is to estimate the scene components (3D points, 3D objects, and 3D regions) as well as their correspondences from the image measurements. In Sec. 3.1, we have discussed how to acquire these measurements from images. In this section, we explain how to estimate scene components based on these measurements.

4.1. Estimating 3D Objects

It has been shown by [15, 4] that it is possible to roughly estimate the 3D location of an object given its detection in the image, the camera focal length and prior knowledge about the object's physical scale. We obtain the initial set of 3D objects O_t based on detected object measurements with high detection scores (Fig. 4b). The correspondences $\{v_t\}$ are given by the projection of $\{O_t\}$ in images. Based on the initial set of 3D objects, we search for the best configuration of objects by $\mathbb{O} = \arg\max_{\mathbb{O}=\{O_t\}} \Pi_t \Psi_t^{CO}(\mathcal{O}_t, \mathbf{C}; \mathbf{I})$ given

	Camera	Point	Object	Region
Measurements	-	q_i^k (Fig.2a)	o_j^k (Fig.2b)	b_l^k (Fig.2c)
Set Notation	-	$\mathbf{q} : \{q_i^k\}$	$\mathbf{o} : \{o_j^k\}$	$\mathbf{b} : \{b_l^k\}$
3D Unknowns	C^k	Q_s (Fig.2a)	O_t (Fig.2b)	B_r (Fig.2c)
Set Notation	$\mathbf{C} : \{C^k\}$	$\mathbf{Q} : \{Q_s\}$	$\mathbf{O} : \{O_t\}$	$\mathbf{B} : \{B_r\}$
Correspondences	-	u_s	v_t	g_r
Pair Notation	$\mathcal{C}^k : \{C^k, I^k\}$	$\mathcal{Q}_s : \{Q_s, u_s\}$	$\mathcal{O}_t : \{O_t, v_t\}$	$\mathcal{B}_r : \{B_r, g_r\}$
Set Notation	$\mathbb{C} : \{\mathcal{C}^k\}$	$\mathbb{Q} : \{\mathcal{Q}_s\}$	$\mathbb{O} : \{\mathcal{O}_t\}$	$\mathbb{B} : \{\mathcal{B}_r\}$

Table 2: Key notations. Sub-indices s, t, r indicate 3D points, 3D objects, and 3D regions respectively (e.g. $(X, Y, Z)_s$ is the 3D location for a 3D point Q_s, and $(X, Y, Z)_t$ is the 3D centroid for a 3D object O_t). The sub-indices i, j, l indicate 2D measurements of points, objects, and regions respectively (e.g. $(x, y)_i$ is the image location of a 2D point q_i, and $(x, y)_l$ is the image location of the centroid of a 2D region b_l). The super-index (usually k) denotes which camera / image a variable is related to (e.g, q_i^k is the i^{th} 2D point in the k^{th} image). Bold and blackboard bold letters are used to denote set of variables.

camera hypothesis $\mathbf{C}$. The object energy is computed as $\Psi_t^{CO} = \Pr(\mathbf{o}, v_t | O_t, \mathbf{C})$, which is the conditional probability of observing the measurements of O_t. Please refer to [3] for more details about $\Pr(\mathbf{o}, v_t | O_t, \mathbf{C})$ and the optimization process.

4.2. Estimating 3D Points

In order to obtain 3D points $\mathbf{Q}$, we first establish the correspondences of detected interest points $\mathbf{q}$ across views. As in most SFM algorithms, a correspondence u_s implies the existence of a 3D point Q_s. A correspondence u is likely to be true only if: 1) the points linked by u have similar feature descriptors, 2) the location of the 2D points linked by u are compatible with the cameras $\mathbf{C}$ (e.g. epipolar line constraints). u_s can be established by any feature matching algorithm (e.g. [22, 30]). The corresponding Q_s is estimated by triangulation (Fig. 4a). Given Q_s and u_s, we compute $\Psi_s^{CQ} = \Pr(\mathbf{q}, u_s | Q_s, \mathbf{C})$, which is the conditional probability of observing the measurements of Q_s. We search for the best configuration of points by solving $\mathbb{Q} = \arg\max_{\mathbb{Q}=\{\mathcal{Q}_s\}} \Pi_s \Psi_s^{CQ}(\mathcal{Q}_s, \mathbf{C}; \mathbf{I})$ given camera hypothesis $\mathbf{C}$. Please refer to [3] for more details about $\Pr(\mathbf{q}, u_s | Q_s, \mathbf{C})$ and the optimization process.

4.3. Estimating 3D Regions

In order to obtain 3D regions, similarly to points, we first establish the correspondences of region measurements across views. A correspondence implies the existence of a 3D region B_r. Sec. 4.3.1 explains how to obtain an initial set $\{B_r\}$ of 3D regions given the proposed correspondences. Sec. 4.3.2 explains how to compute the energy for 3D regions using the initial set. We select $\mathbb{B}$ as the solution of $\mathbb{B} = \arg\max_{\mathbb{B}=\{\mathcal{B}_r\}} \Pi_r \Psi_t^{CB}(\mathcal{B}_r, \mathbf{C}; \mathbf{I})$.

Algorithm 1 Sampling parameters to search for the solution for Eq. 1.

```
Propose initial guesses of camera poses [1].
FOR C ∈ the set of initial guesses
  C_0 = C;
  FOR n = 1 : M  (M is predefined)
    C_n = C_{n-1} + C' where C' is 0-mean Gaussian r.v.
    O'_n ⇐ arg max_{O={O_t}} Π_t Ψ_t^{CO}(O_t, C_n; I);  (Sec. 4.1)
    Q'_n ⇐ arg max_{Q={Q_s}} Π_s Ψ_s^{CQ}(Q_s, C_n; I);  (Sec. 4.2)
    B'_n ⇐ arg max_{B={B_r}} Π_r Ψ_t^{CB}(B_r, C_n; I);  (Sec. 4.3)
    {O_n, Q_n, B_n} ⇐ arg max Π_{t,s} Ψ_{t,s}^{OQ} Π_{t,r} Ψ_{t,r}^{OB} Π_{r,s} Ψ_{r,s}^{BQ} by
    gradient descent starting from {O'_n, Q'_n, B'_n}  (Sec. 5)
    α = Ψ(O_n,Q_n,B_n;C_n,I) / Ψ(O_{n-1},Q_{n-1},B_{n-1};C_{n-1},I);
    IF α < ϱ where ϱ ~ U(0,1)  (uniform random variable)
      {O_n, Q_n, B_n} = {O_{n-1}, Q_{n-1}, B_{n-1}};
    END
  END
END
```

4.3.1 Initializing 3D Regions Given 2D Regions

To initialize 3D regions, we first identify across-view correspondences among 2D regions. The 2D regions can be obtained from each image independently [26] or coherently [33]. We used [26] in our experiment. We use epipolar constraints and appearance matching (using color histograms and texture features) to find a set of potential matches. As visualized in Fig. 4c, given a set of matched 2D regions $\{b_{l^k}^k\}$ and camera hypothesis $\mathbf{C}$, we initialize $B_r = (X, Y, Z, n, s, c)_r$ as follows.

- Region Centroid $(X, Y, Z)_r$**.** The region centroid is obtained by triangulating the centroids of $\{b_{l^k}^k\}$.

- Region Normal n_r**.** Using the appearance and location of the region $b_{l^k}^k$, methods such as [32, 20, 14] can be employed to estimate the normal $n_{l^k}^k$ of $b_{l^k}^k$ from the camera C^k. For instance, [32] allows to estimate the region normal from the camera view point using properties of the rank of texture matrices. [14] classifies regions into a few discretized normal orientation categories (flat, front, side, etc...). [20] uses vanishing lines to estimate the region's normal w.r.t the camera. Given $n_{l^k}^k$ and C^k, we can estimate the normal n_r^k of $b_{l^k}^k$ in the world system. We obtain the normal n_r of B_r as $n_r = \frac{1}{N_c} \Sigma_k n_r^k$ where N_C is the number of images.

- Region Area s_r**.** Given $(X, Y, Z)_r$ and C^k, we can compute the distance d_r^k between B_r and C^k. Given n_r and C^k, we obtain the angle α_r^k between n_r and the camera C^k line of sight (Fig. 2c). Each region $b_{l^k}^k$ has an image area $s_{l^k}^k$. $s_r = \frac{1}{N_C} \sum_k \frac{d_r^k s_{l^k}^k}{cos(\alpha_r^k)}$ where N_C is the number of images.

- Region Class c_r**.** The class of B_r is estimated based on the appearance properties of its measurements and its geometrical properties. Given $a_{l^k}^k$ (the appearance of $b_{l^k}^k$), we use methods such as [17, 31, 5] to estimate the confidence $f_{app}(c_r = c; a_{l^k}^k)$ that B_r is of class c. The 3D geometry of B_r can be used as a cue for region classification. For example, most walls are vertical and the most desks are horizontal. Given the estimated geometry $\{n_r, s_r\}$, we learn a KNN classifier to estimate the confidence $f_{geo}(c_r = c; n_r, s_r)$ that B_r is of class c. We compute the probability $\Pr(c_r = c)$ that B_r is of class c as the weighted average of f_{geo} and f_{app}. Notice that estimating the class and ge-

4.3.1 Initializing 3D Regions Given 2D Regions

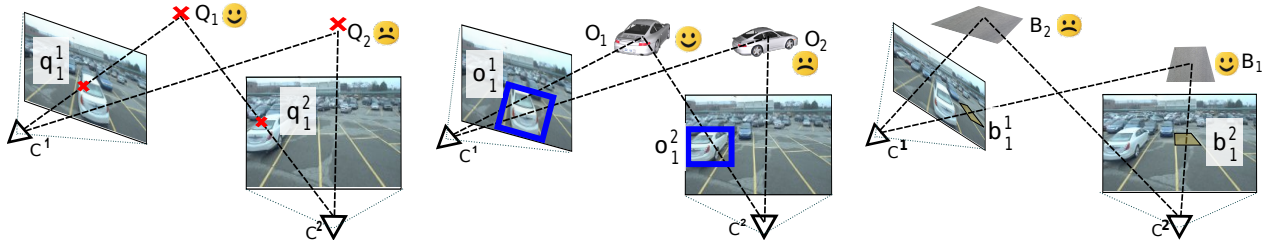
To initialize 3D regions, we first identify across-view correspondences among 2D regions. The 2D regions can be obtained from each image independently [26] or coherently [33]. We used [26] in our experiment. We use epipolar constraints and appearance matching (using color histograms and texture features) to find a set of potential matches. As visualized in Fig. 4c, given a set of matched 2D regions $\{b_{l^k}^k\}$ and camera hypothesis $\mathbf{C}$, we initialize $B_r = (X, Y, Z, n, s, c)_r$ as follows.

- Region Centroid $(X, Y, Z)_r$**.** The region centroid is obtained by triangulating the centroids of $\{b_{l^k}^k\}$.

- Region Normal n_r**.** Using the appearance and location of the region $b_{l^k}^k$, methods such as [32, 20, 14] can be employed to estimate the normal $n_{l^k}^k$ of $b_{l^k}^k$ from the camera C^k. For instance, [32] allows to estimate the region normal from the camera view point using properties of the rank of texture matrices. [14] classifies regions into a few discretized normal orientation categories (flat, front, side, etc...). [20] uses vanishing lines to estimate the region's normal w.r.t the camera. Given $n_{l^k}^k$ and C^k, we can estimate the normal n_r^k of $b_{l^k}^k$ in the world system. We obtain the normal n_r of B_r as $n_r = \frac{1}{N_c} \Sigma_k n_r^k$ where N_C is the number of images.

- Region Area s_r**.** Given $(X, Y, Z)_r$ and C^k, we can compute the distance d_r^k between B_r and C^k. Given n_r and C^k, we obtain the angle α_r^k between n_r and the camera C^k line of sight (Fig. 2c). Each region $b_{l^k}^k$ has an image area $s_{l^k}^k$. $s_r = \frac{1}{N_C} \sum_k \frac{d_r^k s_{l^k}^k}{cos(\alpha_r^k)}$ where N_C is the number of images.

- Region Class c_r**.** The class of B_r is estimated based on the appearance properties of its measurements and its geometrical properties. Given $a_{l^k}^k$ (the appearance of $b_{l^k}^k$), we use methods such as [17, 31, 5] to estimate the confidence $f_{app}(c_r = c; a_{l^k}^k)$ that B_r is of class c. The 3D geometry of B_r can be used as a cue for region classification. For example, most walls are vertical and the most desks are horizontal. Given the estimated geometry $\{n_r, s_r\}$, we learn a KNN classifier to estimate the confidence $f_{geo}(c_r = c; n_r, s_r)$ that B_r is of class c. We compute the probability $\Pr(c_r = c)$ that B_r is of class c as the weighted average of f_{geo} and f_{app}. Notice that estimating the class and ge-



(a) Estimating 3D Point from Measurements. Q_1 and Q_2 are candidate 3D points given measurements q_1^1 and q_1^2. Q_1 is a good candidate because its projection is consistent with the location of q_1^1 and q_1^2.

(b) Estimating 3D Object from Measurements. O_1 and O_2 are candidate 3D objects given measurements o_1^1 and o_1^2. O_1 is a good candidate because its projection is consistent with the pose, location, and scale of o_1^1 and o_1^2.

(c) Estimating 3D Region from Measurements. B_1 and B_2 are candidate 3D regions given measurements b_1^1 and b_1^2. B_1 is a good candidate because its projection is consistent with the pose, location, and scale of b_1^1 and b_1^2.

Figure 4: Estimating scene components from measurements. Notice that we do not estimate the object 3D model. 3D car in (b) is only for visualization.

ometry of 3D regions based on 2D measurements is usually a challenging problem. However, object-region and point-region interactions help us estimate 3D regions more accurately (Sec. 5).

4.3.2 Energy of 3D Regions

The energy Ψ_r^{CB} measures how likely $\mathcal{B}_r = \{B_r, g_r\}$ is consistent with the measurements. Ψ_r^{CB} can be decomposed as a product of two energy terms: $\bar{\Psi}_r^{CB}$ and $\widetilde{\Psi}_r^{CB}$. $\bar{\Psi}_r^{CB}$ captures how likely corresponding across-view measurements of B_r have similar appearance. $\widetilde{\Psi}_r^{CB}$ captures how likely the location and scale of the projection of B_r is consistent with its corresponding measurements.

Energy $\bar{\Psi}_r^{CB}$. The degree of appearance variability of a region as function of the view point depends on the region class. For example, the appearance of a portion of road will not change much as function of view point transformations. Thus,

$$\bar{\Psi}_r^{CB} \propto \sum_c \prod_k N(a_{l^k}^k - \bar{a_r}; 0, \Sigma^c) \Pr(c_r = c) \quad (2)$$

where $a_{l^k}^k$ is the appearance of $b_{l^k}^k$, $\bar{a_r} = \frac{1}{N_c}\sum_k a_{l^k}^k$ is the mean of appearances, $N(\bullet; a, \Sigma)$ is a Gaussian distribution whose mean is a and covariance is Σ, $\Pr(c_r = c)$ is the probability that B_r is of class c. We learn Σ^c from our training set using a max-likelihood estimator.

Energy $\widetilde{\Psi}_r^{CB}$. Given C^k and B_r, let b_r^k be the image projection of B_r. We model the energy $\widetilde{\Psi}_r^{CB}$ by evaluating if b_r^k and the measurement of B_r have: 1) similar image areas, 2) similar normal vectors computed within the camera reference system. Thus,

$$\widetilde{\Psi}_r^{CB} \propto \sum_c \prod_k N(\log \frac{s_r^k}{s_{l^k}^k}; 0, \sigma_{sc}) N(n_{l^k}^k; n_r^k, \sigma_{nc}) \Pr(c_r = c) \quad (3)$$

where $s_{l^k}^k$ (s_r^k) are image area of $b_{l^k}^k$ (b_r^k), $n_{l^k}^k$ is the region normal vector of $b_{l^k}^k$ (estimated by methods such as [32, 14, 20]), and n_r^k is the normal vector of B_r w.r.t. C^k. We chose to use log difference in Eq. 3 since it best fits real data distribution.

Overall Region Energy. Combining Eq. 2 and 3, we compute Ψ_r^{CB} as:

$$\Psi_r^{CB} = \bar{\Psi}_r^{CB} \widetilde{\Psi}_r^{CB} = \sum_c \prod_k N(a_{l^k}^k - \bar{a_r}; 0, \Sigma^c)$$

$$N(\log \frac{s_r^k}{s_{l^k}^k}; 0, \sigma_{sc}) N(n_{l^k}^k; n_r^k, \sigma_{nc}) \Pr(c_r = c)$$

5. Interactions Among Scene Components

The concept of *interactions* among scene components (objects, points, and regions) originates from the observation that scene components are related following certain geometrical or physical rules in 3D. For example, a 3D point may lie on a 3D object (Fig. 5b), and a 3D object may lie on a 3D region (Fig. 5a). Image cues can be used to validate the existence of such interactions (Fig. 5c and Fig 5d). A pair of scene components that are hypothesized to be interacting can be associated to an *interaction energy* (i.e. $\Psi_{t,s}^{OQ}, \Psi_{t,r}^{OB}, \Psi_{r,s}^{BQ}$). The interaction energy is a function of the scene components' 3D locations, poses, and semantic labels[1]. One step in Algorithm 1 is to search for the best configuration of $\mathbb{O}, \mathbb{Q}, \mathbb{B}$ so as to maximize interaction energy (i.e. $\{\mathbb{O}, \mathbb{Q}, \mathbb{B}\} = \arg\max \prod_{t,s} \Psi_{t,s}^{OQ} \prod_{t,r} \Psi_{t,r}^{OB} \prod_{r,s} \Psi_{r,s}^{BQ}$). This step refines the 3D location and pose for all scene components, and is achieved by a gradient descent algorithm. The initial values $\mathbb{O}', \mathbb{Q}', \mathbb{B}'$ for this gradient descent algorithm are provided by the individual estimation of each scene component, as discussed in Sec. 4.1, 4.2, and 4.3.

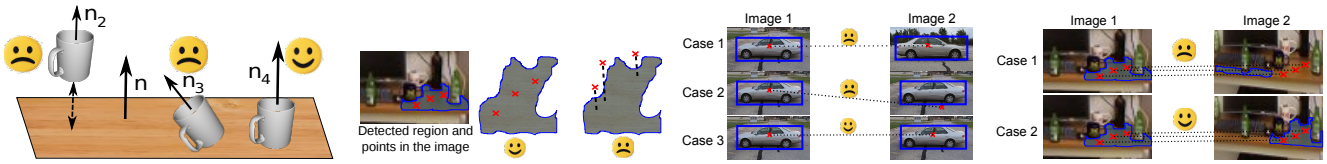
5.1. Object-point Interaction

A point Q_s and an object O_t interact if Q_s lies on the surface of O_t. This also implies that matched points should be consistent with matched objects across views (Fig. 5c). If this consistency criteria is verified, the energy $\Psi_{t,s}^{OQ}$ will have a large value. This type of interaction was explored in [2] and we refer to that paper for more details.

5.2. Object-region Interaction

An object O_t and a region B_r interact if: i) O_t physically sits on the surface defined by B_r; ii) O_t sits in the up-right pose (Fig. 5a). In other words, B_r is the supporting region for O_t. As shown in works such as [15, 4], the supporting region (surface) can be used to add constraints on object detection task. In turn, detected objects help the estimation of the geometry of supporting regions. The energy $\Psi_{t,r}^{OB}$ evaluates the interaction between B_r and O_t. Thus, $\Psi_{t,r}^{OB}$ has large value if the bottom face of O_t touches the surface of B_r (location consistency) and the pose of O_t is up-right (pose consistency) as shown in Fig. 5a.

[1] If two scene components do not interact, their interaction energy is 1.



(a) Object-region interaction. The estimated object (cup) should lie on the region (desk) and be in its up-right pose.

(b) Point-region interaction. The estimated 3D points (red crosses) should lie on the estimated 3D region (a portion of desk).

(c) Object-point interaction. Matched points (crosses) should be consistent with the measurement of the same instance (cars).

(d) Point-region interaction. Matched regions (a portion of desk) should be consistent with measurement of the same points (red crosses) across views.

Figure 5: Interactions among objects, points, and regions. (a,b): types of interactions in 3D. (c,d): types of interactions across views.

In order to reduce the size of an otherwise combinatorial problem, we construct a set of candidate object-region interactions. Each element (e.g. $\{O_t, B_r\}$) in this set is selected following two criteria: i) the bottom side of the bounding box of o_j^k (the image measurement of O_t) lies within b_l^k (the image measurement of B_r), ii) the region class of B_r and object class of O_t are compatible. This compatibility is expressed by an indicator function $I_{t,r}^{OB}(c_t, c_r) = \{0, 1\}$, which returns 1 (compatible) if an object of class c_t and a region of class c_r may interact, and returns 0 (incompatible) otherwise. For instance, $I_{t,r}^{OB}(c_t, c_r) = 0$ if $c_t =$ 'car' and $c_r =$ 'tree', because a car cannot 'sit' on a tree. $I_{t,r}^{OB}(c_t, c_r)$ is learnt to be 1 if in the training set an object of class c_t interacts with a region of class c_r, and it is learnt to be 0 if no object of class c_t interacts with regions of class c_r. If both criteria are met, then we say that a region B_r and an object O_t do interact and we evaluate their corresponding interaction energy.

Location Consistency. If O_t lies on B_r, their locations in the 3D space will be close to each other (Fig. 5a). Denote by $d_{t,r}$ the point-to-plane distance from the bottom of the bounding cube of O_t to the surface of B_r. Assuming a Gaussian measurement noise, $d_{t,r}$ follows a zero-mean Gaussian distribution controlled by a variance σ^{OB}. In general, σ^{OB} is a function of object category c_t and region class c_r. $d_{t,r}$ can be used to evaluate $\bar{\Psi}_{t,r}^{OB}$:

$$\bar{\Psi}_{t,r}^{OB} \propto \sum_c [I_{t,r}^{OB} N(d_{t,r}; 0, \sigma_d^{OB}(c_t, c_r)) + (1 - I_{t,r}^{OB})] \Pr(c_r = c) \tag{4}$$

where $\Pr(c_r = c)$ is the confidence that B_r is of class c (Sec. 4.3.1).

Pose Consistency. If O_t lies on B_r, the pose of O_t and B_r should be consistent (Fig. 5a): n_r (the normal of B_r) should be equal to n_t (the normal of O_t). The angle between n_r and n_t can also be used to evaluate $\tilde{\Psi}_{t,r}^{OB}$:

$$\tilde{\Psi}_{t,r}^{OB} \propto \sum_c [I_{t,r}^{OB} N(acos(n_r' n_t); 0, \sigma_n^{OB}(c_t, c_r)) + (1 - I_{t,r}^{OB})] \Pr(c_r = c) \tag{5}$$

where $\sigma_n^{OB}(c_t, c_r)$ is the angle variance for object category c_t and region category c_r.

Interaction Energy. Given Eq. 4 and 5, $\Psi_{t,r}^{OB}$ is computed as:

$$\Psi_{t,r}^{OB} = \bar{\Psi}_{t,r}^{OB} \tilde{\Psi}_{t,r}^{OB} = \sum_c [I_{t,r}^{OB} N(d_{t,r}; 0, \sigma_d^{OB}(c_t, c_r)) \\ N(acos(n_r' n_t); 0, \sigma_n^{OB}(c_t, c_r)) + (1 - I_{t,r}^{OB})] \Pr(c_r = c) \tag{6}$$

5.3. Point-region Interactions

A point Q_s and a region B_r interact if Q_s lies on the surface of B_r. This implies that: i) the image measurements of Q_s and B_r should be consistent (Fig. 5b); ii) the 3D locations of Q_s and B_r should be close to each other (Fig. 5d). As shown in works such as [27, 9], point-region interactions help improve the across-view matching accuracy of both points and regions, and help the estimation of 3D locations and poses of both points and regions. The energy $\Psi_{s,r}^{QB}$ evaluates the interaction between Q_s and B_r. Thus, $\Psi_{s,r}^{QB}$ has large value if Q_s is close to B_r in 3D (wrong match of points or regions will cause the estimated 3D point Q_s and region B_r to be far apart).

Similarly to object-region interaction, we construct a set of candidate point-region interactions. A pair $\{Q_s, B_r\}$ will be selected as an element in this set if q_i^k (image measurement of Q_s) lies within b_l^k (image measurement of B_r). Denote by $d_{s,r}$ the point-to-plane distance between Q_s to B_r. Assuming a Gaussian measurement noise, $d_{s,r}$ obeys a zero mean Gaussian distribution controlled by a variance $\sigma^{QB}(c_r)$, which is a function of region class. Thus, we have

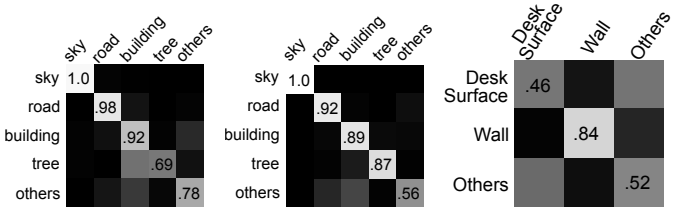
$$\Psi_{s,r}^{QB} = \sum_c N(d_{s,r}; 0, \sigma^{QB}(c_r)) \Pr(c_r = c) \tag{7}$$

6. Evaluation

We evaluate the performance of the proposed framework with the static-scene datasets of [3]. These include two outdoor datasets (car and person) and one indoor dataset (office). Such datasets contain pairs of images with various baselines of the same scene for a number of scenarios (Fig. 7). We use a training set to learn the variances in Eq. 2 ~ Eq. 7 by maximum likelihood estimation. We use the same set of pairs of images as [3].

	[28]	[3]	This paper
Car	26.5° / < 1°	19.9° / < 1°	**12.1°** / < 1°
Person	27.1° / 21.1°	17.6° / 3.1°	**11.4°** / **3.0°**
Office	8.5° / 9.5°	4.7° / 3.7°	**4.2°** / **3.5°**

Table 3: The error for camera pose estimation. We follow the evaluation criteria in [24]. The two numbers in each cell are translation error e_T / rotation errors e_R. Let R_{gt} and T_{gt} be the ground truth camera rotation and translation, and let R_{est} and T_{est} be the estimated camera rotation and translation. The rotation error e_R is the minimal rotation angle (in degree) of $R_{gt} R_{est}^{-1}$. The translation error e_T is the angle between the estimated baseline and the ground truth baseline, i.e. $e_T = acos(\frac{T_{gt}^T R_{gt}^{-T} R_{est}^{-1} T_{est}}{|T_{gt}| \cdot |T_{est}|})$. The reported errors are computed on the second camera, given that the first camera is at the canonical position.



(a) Car. Ours w/ [17]. (b) Person. Ours w/ [17]. (c) Office. Ours w/ [31].

Figure 6: Confusion table for region classification by our framework. "w/" = "with". Our framework can use the appearance-based classification confidence (i.e. f_{app} in Sec. 4.3.1) produced by either [17] or [31].

6.1. Quantitative Performance

Camera pose. We evaluate the ability of our framework to estimate the camera poses from input images (the internal parameters are known). The results are reported in Tab. 3. As our framework jointly uses points, objects and regions to estimate the camera poses, it shows superior performance over our baseline algorithms [28] (which only uses points) and [3] (which only uses points and objects).

Region recognition (classification). For car / person / office set, we train region classifier on car [3] / Camvid [6] / office [3] dataset. Fig. 6 shows the confusion table for the three datasets. Tab. 4 shows the average accuracy for region classification tasks.

3D region orientation. Tab.5 shows the average error in estimating region orientations. The baseline algorithms are Zhang et al. [32] and Hoiem et al. [14]. [32] estimates the region orientation by texture. Notice that [32] tends to fail if a region has little or irregular texture. [14] estimates the region orientation by its appearance and location in the image. Since our framework estimates the region orientation by object-region and point-region interactions in addition to the appearance of a region, it shows better performance in estimating region orientations.

3D region localization. Tab. 6 shows the average error for 3D region localization tasks. Notice the contribution of the interaction in improving the accuracy in localizing regions in 3D.

2D object detection. Tab. 7 shows the average precision for object detection. The baseline algorithms are Felzenszwalb et al. [8] and Bao and Savarese [3]. Since our framework detects objects from multiple images and leverages the interaction between scene components, it enables better object detection performance than [8] (which detect objects from each single image separately) and [3] (which does not consider the interactions).

Percentage %	Car	Person	Office
[17] / [31]	88.9 / 78.2	82.9 / 74.6	50.8 / 54.7
Ours with [17]/[31]	**90.2 / 80.0**	**84.4 / 75.4**	**51.2 / 59.0**

Table 4: Average accuracy for region classification tasks. The numbers are percentage of total pixels correctly labeled / total pixels in the reconstructed regions. We report the classification results using all the 2D regions that our framework is capable to match and reconstruct in 3D. In car / person / office dataset, our framework is able to match and reconstruct the regions that account for 14.4% / 9.6% / 11.3% of total area of all input images. Our framework uses the appearance-based classification confidence (i.e. f_{app} in Sec. 4.3.1) produced by either [17] or [31].

	Zhang et al.[32]	Hoiem et al.[14]	This Paper
Car	69.2°	27.0°	**26.1°**
Office	45.2°	40.5°	**37.9°**

Table 5: Average error in estimating region normals. Let n_{gt} be the ground truth normal of a region from range data. If the estimated region normal is n_{est}, the error is $acos(|n'_{gt}n_{est}|)$. Since the person dataset does not have range data, we do not use it for this evaluation.

with / without interaction	$median(e_d)$	$var(e_d)$
Car	0.281 / **0.175**	0.54 / 0.44
Office	0.033 / **-0.011**	0.182 / 0.189

Table 6: Average error in estimating region 3D locations. The two numbers in each cell are our results estimated with / without interactions. For a detected region, let d_{est} be its distance to camera, and let d_{gt} be its ground truth distance from the range data. We define the error as $e_d = \log \frac{d_{est}}{d_{gt}}$ that should be 0 if there is no error. Since the person dataset does not have range data, we cannot evaluate on it.

3D object detection. Tab. 8 shows the average precision for object detection in 3D space. Ground truth 3D objects are manually labeled from range data. If the distance between the centroid of a detected 3D object and the centroid of a ground truth object is less than δ (values in the caption of Tab. 8), the detected 3D object is counted as true. Since every detected 3D object can be associated to a confidence value, we can generate precision-recall curves and compute the average precision. Due to the metric reconstruction ambiguity, we use ground truth camera poses in this experiment. Our baselines are Hoiem et al. [15] and Bao and Savarese [3]. [15] uses the 2D bounding box to estimate the location of an object in 3D. [3] estimates 3D object locations without modeling interactions. Our framework shows much better performance over [15] and [3]. This demonstrates that the importance of modeling interaction to solve the camera and structure estimation problems.

7. Conclusion

We have proposed a novel framework for jointly estimating camera poses and detecting objects, points, and re-

	[8]	[3]	This Paper
Car	54.5%	61.3%	**62.8%**
Person	70.1%	75.1%	**76.8%**
Office	42.9%	45.0%	**45.7%**

Table 7: 2D object detection average precision. We follow the criteria as in the PASCAL VOC challenge. In the office dataset, we compute the overall average precision for 5 categories: monitors, mice, keyboards, bottles, and cups.

	Estimation by single image. [15]	Without interactions [3]	With interactions. Our full model.
Car	21.4%	32.7%	**43.1%**
Office	15.5%	20.2%	**21.6%**

Table 8: Average precision for 3D object detection. We set $\delta = 2.5/0.1$ meter for car / office dataset.

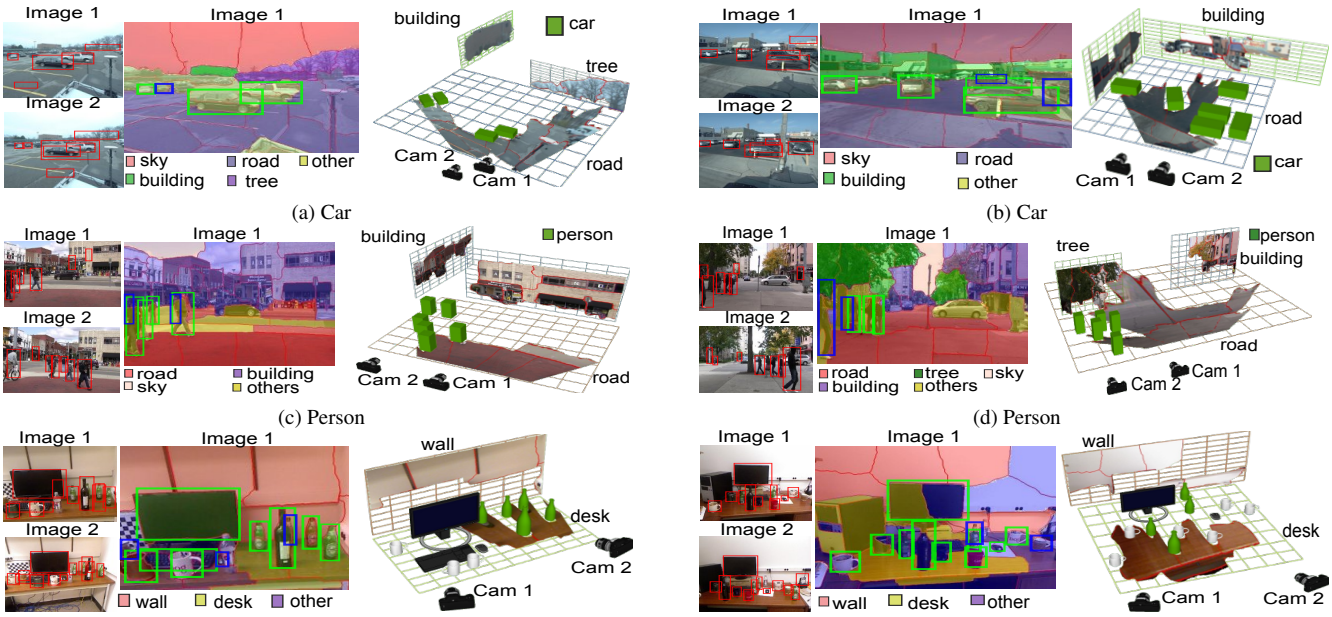


Figure 7: Each panel shows anecdotal results by our framework. **Left**: Input image pair and baseline object detection by [8]. **Middle**: Segmented regions by [26] (delimited by red boundary), region classes, and improved object detection (green and blue bounding boxes). Notice that many false alarms are removed and missed positive are recovered (blue bounding box). **Right**: Reconstructed 3D scene with objects and regions along with the estimation of the location and pose of cameras. Estimated planes appear perpendicularly w.r.t others because we use [14] to initialize 3D plane orientations.

gions from two or multiple semi-calibrated images. We have demonstrated that by modeling the interaction among points, regions, and objects, we obtain more accurate results than if these scene elements are estimated in isolation.

8. Acknowledgment

We acknowledge the support of NSF CAREER #1054127, the Gigascale Systems Research Center, and the KLA-Tencor Fellowship.

References

[1] Project web: http://www.eecs.umich.edu/vision/projects/ssfm/index.html. 4

[2] S. Y. Bao, M. Bagra, and S. Savarese. Semantic structure from motion with object and point interactions. In *IEEE Workshop on Challenges and Opportunities in Robot Perception (in conjunction with ICCV)*, 2011. 2, 5

[3] S. Y. Bao and S. Savarese. Semantic structure from motion. In *CVPR*, 2011. 2, 4, 6, 7

[4] S. Y. Bao, M. Sun, and S. Savarese. Toward coherent object detection and scene layout understanding. *Image and Vision Computing*, 29(9), 2011. 2, 3, 5

[5] A. Berg, F. Grabler, and J. Malik. Parsing images of architectural scenes. In *ICCV*, 2007. 4

[6] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008. 2, 7

[7] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3d urban scene modeling integrating recognition and reconstruction. *IJCV*, 2008. 2

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2009. 2, 7, 8

[9] V. Ferrari, T. Tuytelaars, and L. V. Gool. Wide-baseline muliple-view correspondences. In *CVPR*, 2003. 6

[10] R. Garg, S. Seitz, D. Ramanan, and N. Snavely. Where's waldo: Matching people in images of crowds. In *CVPR*, 2011. 2

[11] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 2

[12] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 2

[13] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010. 2

[14] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007. 4, 5, 7, 8

[15] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 2008. 2, 3, 5, 7

[16] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 1983. 3

[17] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010. 4, 7

[18] L. Ladicky, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *IJCV*, 2011. 2

[19] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *BMVC*, 2004. 2

[20] D. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 2, 4, 5

[21] 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. 2

[22] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 2, 4

[23] K. Mikolajczyk and C. Schmid. Scale affine invariant interest point detectors. *IJCV*, 2004. 1

[24] D. Nister. An efficient solution to the five-point relative pose problem. *TPAMI*, 2004. 6

[25] N. Payet and S. Todorovic. Scene shape from texture of objects. In *CVPR*, 2011. 2

[26] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, 2003. 2, 4, 8

[27] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *ICCV*, 2001. 2, 6

[28] N. Snavely, S. M. Seitz, and R. S. Szeliski. Modeling the world from internet photo collections. *IJCV*, (2), Nov. 2008. 1, 6, 7

[29] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbob. Bundle adjustment: a modern synthesis. In *Vision Algorithms: Theory and Practice*, 1999. 1

[30] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *British Machine Vision Conference*, 2000. 2, 4

[31] M. Varma and R. Garg. Locally invariant fractal features for statistical texture classification. In *ICCV*, 2007. 4, 7

[32] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma. Tilt: Transform invariant low-rank textures. *IJCV*, 2010. 4, 5, 7

[33] C. Zitnick, N. Jojic, and S. Kang. Consistent segmentation for optical flow estimation. In *ICCV*, 2005. 2, 4