Visual-Inertial Semantic Scene Representation
UCLA TR CSD160005

Stefano Soatto
May 20, 2016

Abstract

We describe a representation of a scene that captures geometric and semantic attributes of objects within, along with their uncertainty. Objects are assumed persistent in the scene, and their likelihood computed from intermittent visual data using a convolutional architecture, integrated within a Bayesian filtering framework with inertials and a context model. Our method yields a posterior estimate of geometry (attributed point cloud and associated uncertainty), semantics (identities and co-occurrence), and a point-estimate of topology for a variable number of objects within the scene, implemented causally and in real-time on commodity hardware.

1 Introduction

We design a representation of a scene, and infer it from visual and inertial data, to support interaction tasks, including localization (where am I?), navigation (where can I go?), ambient object detection (what is around me?), and prediction (where is it going?). Correspondingly, the representation maintains (posterior) estimates of geometric, topological, semantic, and dynamic attributes of the scene and objects within.

The task informs what are the nuisance factors in the data, which are shared among several tasks. We focus on illumination, viewpoint, and partial occlusion, in addition to calibration, alignment, and other sensor-specific characteristics. An ideal representation would be a minimal sufficient statistic (of the data, for the task) that is also invariant (to nuisance factors) [53].

Geometry. A gravity-aligned, sparse attributed Euclidean point cloud, with local illumination- and pose-invariant descriptors as attributes [15], is a minimal sufficient invariant for localization tasks, including relative localization of multiple sensor platforms/agents (Fig. 1). Unlike monocular vision-based SLAM, inertial measurements enable inference of global orientation and (Euclidean) scale. We leverage existing work [30, 40, 56] to approximate the posterior distribution of geometric attributes given visual and inertial measurements, causally and in real-time. This, however, is insufficient for navigation tasks where one must discriminate free space from photometrically unexciting surfaces.

Topology. Determining which points are close, not in the sense of the embedding topology but in the sense of belonging to the same surface, is necessary for navigation, but also to organize the scene into “objects”...
that have identities and relations that can affect high-level planning. We leverage on existing work [24] to infer surface topology given images and (MAP) point-estimates of camera poses as above (Fig. 2).

**Semantics.** We define objects as (compact) regions of 3D space that produce measurable correlates in

Figure 2: Topology estimation for some segments of the scenes in Figs. 3 and 7. Although significant gaps still exist, especially in the dark-mesh chair and black monitor, geometric entities in the representation are connected as belonging to the same surface, which supports both navigation (free space traversal) and semantic labeling.

images that are shared with high probability in a population of image regions labeled as equivalent by an oracle. Objects exist in the scene, not on the image, that only provides evidence (likelihood) of a given object being present in the scene. This can approximated using a convolutional architecture (CNN) [23, 45], interpreted not as an object detector, but as a likelihood function for objects being present in the scene. This is in line with [53], where CNNs are seen as computing marginal/profile likelihoods, which are minimal sufficient invariant statistics of a single image.

Since we are interested in interaction tasks, not in single-image interpretation, we wish to maintain a posterior estimate of objects in the scene given all past data, causally and in real-time. Objects are persistent; they do not flicker in and out of existence as we move smoothly through physical space.

**Context** In addition to identities, the representation should encode relations among objects. Geometric relations are already encoded in the geometric component of the representation, since each object has pose attributes. This enables answering questions about mutual position among objects (Fig. 1). Topological relations, such as co-visibility, are also captured in the representation (Fig. 2). Semantic relations, such as co-occurrence and relative pose and identities, are modeled by a graph with unary and pairwise potentials [10] with a Dirichlet prior.

**Dynamics** In addition to identities and relations, objects have dynamic properties, that affect how they are likely to move at future times. We put the emphasis on real-time, since a representation for interaction tasks must be inferred and updated in real-time to enable closing the loop. While there is a growing body of work on long-term motion (“intent”) prediction, at present none that we considered is suitable for real-time posterior update, and therefore we only address slow motion (relative to the sensor update rate) and defer long-term prediction to future work.

**Contributions and related work**

We design an inference procedure for a persistent scene representation from a monocular video and inertial data streams. It captures geometric, topological, and semantic properties, and rudimentary dynamics. To the best of our knowledge, this is the first system of its kind that can operate in real time at scale. We leverage on existing visual-inertial sensor fusion [30, 40, 56], topology estimation [24], object detection [23, 45], and context modeling [10].

We stress the importance of inertial sensors, not so much for dead-reckoning, but to validate object hypotheses in a Bayesian setting, by making global orientation and scale observable so that, for instance, a model car in our system is not classified as a car (Fig. 3), despite image evidence of the contrary.

Figure 3: Left-to-right: A model car (1), detected as a car by a CNN (2), dismissed by our model (3), because its geometric attributes (4) have small likelihood under the model.

---

1 We take a supervised approach, where identities are determined by the supervisor. Alternatively, one could define objects in terms of functional or topological characteristics that yield measurable correlates, for instance being “detachable,” [3], or use other sensory modality, such as touch.
There is a sizable literature on semantic scene understanding from a single image ([19] and references therein). We are instead interested in agents embedded in physical space, so the restriction to a single image is limiting. There is also a vast literature on scene segmentation, including dedicated workshops and special issues ([31] and references therein), mostly using range (RGB-D) sensors, although also from video. There is also work on 3D recognition ([35] [19] [43]), but again with no inertial frame and no motion. Some focus on real-time operation ([13]), but most operate off-line. None of the datasets commonly used in these works ([11] [11] [60] provide an inertial reference, except for KITTI on which we comment later.

This work, by its nature, relates to a vast body of work on “scene understanding,” which has been the focus of attention of the Computer Vision, Robotics ([39] [44]) and AI communities ([37]) for a considerable amount of time ([55]). Most recently, since the advent of cheap consumer range sensors, these communities have joined forces and there is a wealth of contemporary work ([41] [55] [59] [9] [51] [14] [26] [34] [46] [29] [5] [25] [27] [48] [57]). The use of RGB-D cameras unfortunately restricts the domain of applicability mostly indoors and at close range. Most of this work does not leverage on the similar recent accessibility to cheap inertial sensors.

There is also work that focuses on scene understanding from visual sensors, specifically video ([38] [1] [42] [50] [6] [61]), although none integrates with inertial sensors, although there is recently resurgent interest in sensor fusion ([62]). Additional related work includes ([28] [12] [7] [44]).

Finally, our work relates to parallel efforts to develop a theory of representations, including ([21] [2] [8], although focused on representing the scene and its three-dimensional geometry rather than representing images.

2 Modeling

2.1 Representations

A scene \( \theta \) is populated by a number of objects \( z_j \in \{z_1, \ldots, z_N \} \), each with geometric (position, shape) and semantic (label) attributes \( z_j = \{x_j, y_j, t_j\} \). Measurements (e.g., images) up to the current time \( t \), \( y^t = \{y_1, \ldots, y_t\} \) are captured from a sensor at pose \( g_t \). A representation of the scene in support of semantic tasks is the joint posterior \( p(\theta, z^j|y^t) \) for up to the \( j \)-th objects seen up to time \( t \), where sensor pose \( g_t \) and other nuisances are marginalized. The joint posterior can be decomposed as

\[
p(\theta, z^j|y^t) = p(\theta|z^j) p(z^j|y^t)
\]

with the first factor updated asynchronously each time a new object \( z_{j+1} \) becomes manifest starting from a prior \( P(\theta|\emptyset) \) and the second factor [2] updated synchronously as new measurements \( y_{t+1} \) become available starting from \( t = 0 \) and given \( p(z|\emptyset) \).

A representation of the scene in support of (geometric) localization tasks is the posterior \( p(g_t, x|y^t) \) over sensor pose \( g_t \) (which, of course, is not a nuisance for this task) and a sparse attributed point cloud \( x = \{x_1, \ldots, x_{N_x}\} \), given all measurements (visual and inertial) up to the current time. Conditioning the semantics on the geometry\(^5\) we can write \([3]\) as

\[
p(z^j|y^t) = \int p(z^j|g_t, x, y^t) dP(g_t, x|y^t)
\]

\(^2\)A scene instance can thus be thought of as a graph with nodes \( z^j = \{z_1, \ldots, z_j\} \) and edges capturing (geometric, positional and semantic) relations.

\(^3\)If the goal is to infer objects \( z \) from images \( y^t \), marginalizing the above would ignore the context \( \theta \). This would make it impossible to exploit context knowledge while learning objects directly from data. However, the context model can be learned from object knowledge (first factor), and used to validate object hypotheses (second factor). Assuming a mixture model for \( p(\theta|z) \), we can then exploit the context model while inferring objects via \( \text{max}_\theta p(z|\theta, y^t) \). Alternatively, an empirical Bayes approach would first infer \( \hat{\theta} = \text{arg max}_\theta p(\theta|y^t) \) and then perform inference on \( z \) using \( p(z|\hat{\theta}, y^t) \). We adopt the former method for simplicity.

\(^4\)Attributes include sparse geometry (position in the inertial frame) as well as local photometry (feature descriptor).

\(^5\)Note that these approximations allow us to separate the computation of the geometric representation (the “dorsal stream” to determine “where”) from the semantic representation (the “ventral stream” to determine “what”).
where the integrand can be updated as more data $y_{t+1}$ becomes available using
\[ p(z^j | g_{t+1}, x, y^{t+1}) \propto p(y_{t+1} | z^j, g_{t+1}, x) \int p(g_{t+1} | g_t, u_t) dP(z^j | g_t, x, y^t). \]  
\(3\)

In an active perception setting, we distinguish nuisances that can be (partially) controlled, like viewpoint/pose $g_t$, from those that cannot like contrast/illumination $\kappa_t$. Here $u_t$ can be a control action, e.g., translational and rotational velocity affecting the pose $g_{t+1} = g_{t+1}u_t$. The transition probability is trivial in this case since the next pose $g_{t+1}(u_t)$ is a deterministic function of the current pose $g_t$ and the controlled velocity $u_t$.

### 2.2 Approximations

The measure in (2) can be approximated to second-order by a simultaneous localization and mapping (SLAM) system: $p(g_t, x | y^t) \simeq N(g_t, x; \hat{g}_{t|t}, \hat{x}_{t|t}, P_{t|t})$. Eq. (2) is a diffusion around the mean/mode $\hat{g}_{t|t}, \hat{x}_{t|t}$; if the covariance $P_{t|t}$ is small, it can be ignored, so given
\[ \hat{g}_{t|t}, \hat{x}_{t|t} = \arg\max_{g_t, x} p(g_t, x | y^t) \]  
\(4\)
then $\hat{p}_{g,x}(z|y^t) \equiv p(z|g_t = \hat{g}_{t|t}, x = \hat{x}_{t|t}, y^t) \simeq p(z|y^t)$. Otherwise the marginalization in (2) can be performed using samples from the SLAM system. Either way, omitting the subscripts, we have
\[ \hat{p}(z|y^{t+1}) \propto \underbrace{p(y_{t+1} | z, \hat{g}_{t|t}u_t, \hat{x}_{t|t})}_{\text{CNN}} \underbrace{\hat{p}(z|y^t)}_{\text{EKF/PMF}} \]  
\(5\)
where the likelihood term is approximated by a convolutional neural network (CNN) as shown in Sect. 3.5 and the posterior is updated by a filter (EKF/PMF) as shown in Sect. 3.4. Finally, the first factor in (2)
\[ p(\theta | z^{j+1}) \propto \underbrace{p(z_{j+1} | \theta) P(\theta | z^j)}_{\text{RNN}} \]  
\(6\)
can be approximated using a recurrent neural network (RNN) as we show in Sect. 3.5.

#### Remark 1 (CNN as a likelihood function)
Approximating the likelihood $p(y|l, s, g, x)$ in (3) appears daunting because of the purported need to generate the data $y$ – including every pixel in the image, given object class $l$, shape $s$ and pose $g$ – and the need to normalize with respect to all possible images of the object. Fortunately, the latter is not needed since the product on the right-hand side of (3) needs to be normalized anyway, which can be done easily in a particle-based representation of the posterior by dividing by the sum of the weights of the particles. Generating actual images is similarly not needed. What is needed is a mechanism that, for a given image $y_{t+1}$, allows quantifying the likelihood that an object of any class $l$ with any shape $s$ be present in any portion of the image $b$ where $s$ projects to from the vantage point $g_t$. In Sect. 3.3 we will show how a discriminatively-trained CNN can be leveraged to this end.

#### Remark 2 (Interpretability of the representation)
While the representation $\theta$ is abstract and not interpretable, it is sufficient for predicting/hallucinating objects, which are interpretable. They have geometric and semantic attributes that can be verified independently. In particular, given any point estimate $\hat{\theta}$, for instance $\hat{\theta}(z^j) = \mathbb{E}(p(\theta | z^j))$ or $\arg\max_{\theta} p(\theta | z^j)$, we can predict unseen objects by sampling $p(z_{j+1} | \theta)$. We can also score configurations of objects $z^j$ for compatibility with a scene estimate by evaluating $p(z^j | \theta)$.

\[\text{[6]}\]Which phenomenon can be considered a nuisance, and which is controlled, depends on the task and the sensor platform available. In photometric stereo, one can control the light but not the vantage point. In a mobile robot, one can control the vantage point but not the light. In a mobile active-illumination setting, one can control both.
2.3 Information

Given measurements up to the current time, \( y^t \), the information a (future) measurement \( y_{t+1} \) contains on the scene/objects is, formally,

\[
\mathbb{I}(z, y_{t+1}|y^t) = \mathbb{H}(y_{t+1}|y^t) - \mathbb{H}(y_{t+1}|z, y^t).
\]  (7)

The first term is the uncertainty of future data given the past. It depends indirectly on the true scene, through future data it generates. The second term depends explicitly on the true collection of objects in the scene \( z \), which in general is non-identifiable (Rem. 3). By definition, nuisances are not informative of the scene (although see Rem. 4 for the case of controllable nuisances), so the information content of \( y_{t+1} \) is identical to that of \( \phi_\kappa(y_{t+1}) \) if \( \phi_\kappa \) is a maximal invariant to \( \kappa \). Actionable Information (AI) [52] captures the first term after uncertainty due to non-controllable nuisance variability \( \kappa \) is removed:

\[
\mathcal{H}_t(y_{t+1}) = \mathbb{H}(\phi_\kappa(y_{t+1})|y^t) \quad \text{(8)}
\]

We will show that (7) is equivalent to (8) for exploration, \textit{i.e.}, to determine the (controlled nuisances that yield the) most informative future data.

2.4 Exploration

For the purpose of this section, we assume a data formation model \( h \) that, given the true (objects in the) scene \( z \) and nuisance \( g \), could generate the data \( y \) up to an uninformative/white residual/noise \( n \): \( y = h(z, g) + n \).

In Sect. 2.1 we will relax this assumption to compute functions of the data that are sufficient to evaluate the likelihood. Before measuring \( y_{t+1} \), if we had samples from \( p(z|y^t) \) and \( p(g_t, x|y^t) \), we could determine the action \( u_t \) that would place the sensor at \( g_{t+1}(u_t) = g_t u_t \), so as to hallucinate measurements \( \hat{y}_{t+1} = h(z, g_{t+1}) \) that most decrease uncertainty \((\text{i.e., maximize information})\) on \( z \). From (7) we have

\[
u_t = \arg \max_u \mathbb{H}(y_{t+1}(u)|y^t) - \mathbb{H}(y_{t+1}(u)|z, y^t)
\]  (9)

where \( y_{t+1}(u) = h(z, g_{t+1}(u)) + n_{t+1} \).

Note that the difference between \( y_{t+1} \) and \( \hat{y}_{t+1} \) is uninformative and therefore the maximizer of the information in the two is the same. The second term in (9) is constant for \( y_{t+1} \), and zero for \( \hat{y}_{t+1} \), so it can be ignored in the maximization. It is relevant in the minimization involved in inference and will be revisited in Sect. 2.6.

Also, there is no gain in maximizing the uncertainty in \( y_{t+1} \) due to nuisances \( \kappa \) since, by definition, they are uninformative of the scene. Therefore, we only care to maximize the uncertainty of a \textit{maximal} \( \kappa \)-invariant of \( \hat{y}_{t+1}, \phi_\kappa(\hat{y}_{t+1}) = \phi_\kappa \circ h(z, g_{t+1}(u)) \): From (7) and (8) we have

\[
u_t = \arg \max_u \mathbb{I}(z, y_{t+1}(u)|y^t) = \arg \max_u \mathcal{H}_t(y_{t+1}(u))
\]  (10)

which shows that \textit{the most informative action is the one that maximizes Actionable Information}, hence the name. Controllable nuisances \( u \) are selected to be as informative as possible, whereas the uncontrollable ones \( \kappa \) are removed with no information loss if \( \phi_\kappa \) is a maximal invariant. This process is closely related to that proposed by [22]. We note that nuisances that have the structure of a group acting on the data can be removed without a loss and without the need to gather more data. Nuisances that form a group, however, cannot be eliminated in pre-processing.

Note that we need to simulate the effect of the controlled nuisances \( g(u) \) but not of the uncontrollable ones \( \kappa \), since it does not help hallucinating phenomena that are uninformative. In addition to illumination, cyclo-rotation (rotation about the optical axis) can also be moded-out by rectifying the image with an homography, for instance to gravity-align it using accelerometer measurements (Fig. 4).
A Turing postulate

We call Turing postulate the assumption that, for a compact scene, there exists a (possibly large but finite) set of data $y^t$ whose sufficient statistics are as good as the scene for the purpose of generating future measurements $y_{t+1}$ (or maximal invariants thereof $\phi_k(y_{t+1})$):

$$p(y_{t+1}|z) = p(y_{t+1}|h(z(y^t), g)) \quad \forall g$$

where $z(y^t) \sim p(z|y_{t+1}, x_{t+1}, y^t)$ are from the posterior, which is a minimal sufficient (Bayesian) statistic of $y^t$ for $z$ [53]. The Turing test is passed when the innovation residual $y_{t+1} - h(z(y^t), g)$ is uninformative, according to some empirical whiteness test. The assumptions made thus far are going to be relaxed in Sect. 6.1.

Remark 3 (The “true scene” cannot be known). The scene is, in general, not identifiable as there are infinitely-many scenes that generate the same set of images, no matter how many. In other words, for any given set of data $y^t$ there is an equivalence class (or a distribution) of scenes that can generate them (up to white noise). In this equivalence class, in addition to the “true” scene there are many others, including those that are deterministic or stochastic functions of past data $y^t$, the trivial example being the data itself. Since the task of “knowing the true scene” is undeceivable, we can be satisfied with knowing any scene in the equivalence class, which is sufficient to hallucinate (future) images $\hat{y}_{t+1}$ that are indistinguishable from the measured ones $y_{t+1}$, up to noise. More precisely, we do not care to hallucinate the effect of uninformative nuisances, so we care that the invariant residual $\phi_n(\hat{y}_{t+1}) - \phi_n(y_{t+1})$ be white/uninformative. Thus a representation is a statistic of past data $y^t$ that is (minimal) sufficient to hallucinate (invariants of) future data $y_{t+1}$ up to an uninformative residual [52, 53].

2.5 Exploitation

In some cases, one may wish to infer a point estimate $\hat{z}$ from the representation of the scene, which is the posterior in [3]. Focusing on the second term of (9), we have that

$$\mathbb{H}(y_{t+1}|z, y^t) = \mathbb{H}(y_{t+1}|z) \leq \mathbb{H}(y_{t+1}|\hat{z}(y^t))$$

for any estimator function $\hat{z}(\cdot)$. We can therefore choose the one that minimizes the right-hand side; this requires postulating a likelihood function $p(y_{t+1}|z)$, and computing the empirical entropy

$$\hat{z} = \arg\min_{z(\cdot)} \mathbb{E}_{p(y_{t+1}, y^t)}(-\log p(y_{t+1}|z(y^t)))$$

or, more in general, minimizing with respect to predictors $\hat{p}(y_{t+1}|y^t)$, which in our case is chosen of the form $p(y_{t+1}|\hat{z}(y^t))$. It should be noted that for a linear system with additive, zero-mean Gaussian noise, the above coincides with the maximum a-posteriori (MAP) estimate

$$\hat{z} = \arg\max_z p(z|y^{t+1}).$$

More in general, the relation between the two are described in [53, 17] for linear systems. To be practical, the inference scheme above must be coupled with restrictions on the classes of estimators and penalties on complexity. We adopt a MAP approach and simply provide as a point estimate, when required, the mode of the posterior.

For objects where the posterior converges to a single mode, we can replace the coarse shape with a high-quality model from the prior [40], if specific object recognized, or an iconic prior from the class. Topology estimated by an implicit surface following [24] for some representative objects is shown in Fig. 2. When object move slowly, they can still be represented in the static scene model. If multiple synchronized sensors are available, the static assumption is always satisfied, and therefore moving objects can be represented as if static (Fig. 8).
Figure 4: Gravity is encoded in the representation of the scene in which objects are embedded, and estimated in the moving frame along with the pose of the camera (blue line segment, center); alignment is then obtained by transforming current images by an homography, keeping the error rate approximately constant (right, blue line) despite cyclo-rotation, unlike direct usage of an R-CNN that degrades with the rotation angle (red line, for the object class “sofa”).

Figure 5: Objects exist in the scene (top, bottom), not on the images (middle): Image-based object detection based on an R-CNN (middle), projected onto the scene (top, for indoor and outdoor sequences; each of the 20 categories is color-coded) produces a confused picture since the CNN knows no scene topology. When object objects are native to the scene, however (bottom), so local compactness and persistence can be enforced ab-ovo, the organization of the scene into objects can benefit from the aggregation of the entire time history of measurements through a likelihood function, approximated by a CNN (bottom, showing representative classes detected). Note that the objects are present in the scene (bottom right, shown as bounding boxes embedded in the colored point cloud) even when they do not trigger responses by the CNN on the image (upper-left inset box).

3 Instantiation

Let $I_t$ be raw data captured at time $t$ and $I^t = \{I_0, \ldots, I_t\}$ the “history” of the data up to $t$. For example an image $I_t : D = [640 \times 480] \to [0,255] \forall t = 1, \ldots, T$ with domain $D$ on the pixel lattice. Let $x$ be a description of the (sparse) geometry of the scene, for instance a point cloud $x = [x_1, \ldots, x_{N_x}] \in \mathbb{R}^{3 \times N_x}$. Let $\phi : I \to \mathbb{R}^{N_\phi}$ be any statistic (deterministic function) of the raw data. If the image is restricted to a compact subset of $b \subset D$, $\phi(I_b)$ is sometimes called a “local (feature) descriptor.” A convolutional neural network is often used as a local descriptor $\phi(I_b)$ \cite{he2016deep}, with the entire image $b = D$ as a special case. The pose of the sensor is described by a reference frame $g_t \in SE(3)$; $\kappa_t$ are environmental effects and sensor calibration affecting the range space of the data, for instance monotonic continuous (contrast) transformations $\kappa_t : [0,255] \to [0,255]$ that affect the value $I$ point-wise: $I(x_{ij}) \mapsto \kappa_t(I(x_{ij})) \forall x_{ij} \in D$. Let $l \in \{l_1, \ldots, l_K\}$ be a discrete label or class, that denotes the identity of an object in the scene, $s \subset \mathbb{R}^3$ its shape, in the simplest case a parallelepiped or ellipsoid, and $\rho : s \to \mathbb{R}_+$ its appearance (a.k.a., radiance, reflectance, albedo, texture). An object is then described by $\{l, s, \rho\}$. A point $x \in s$ projects onto a pixel $x_{ij} \in D$ on the camera $g_t$ via an ideal perspective projection $\pi : \mathbb{R}^3 \to \mathbb{R}^2$; $x_{ij} = \pi(g_t x)$ if visible; The pre-image $\pi^{-1}_s(x_{ij}) = \{x \in s \mid \pi(x) = x_{ij}\}$ transforms via $\pi_{gs}(x_{ij}) = g^{-1}\pi_s(x_{ij}) = \{x \in s \mid \pi(gx) = x_{ij}\}$ and may include multiple objects, but only the closest is visible. An object is a set of points $s$, with shape $s$, albedo $\rho$ viewed under illumination $\kappa_t$ from camera $g_t$ generates an image $I_t$ at the pixel $x_{ij} \in D$ via

$$I_t(x_{ij}) = \kappa_t \circ \rho \circ g_t^{-1}\pi_s^{-1}(x_{ij}) + n_t(x_{ij}) \quad (15)$$

where $n_t \sim \mathcal{N}(n; 0, R_n)$ is a residual modeling error that aggregates the effects of all unmodeled phenomena (non-Lambertian reflectance, mutual illumination/inter-reflection, transparency, translucency, sensor calibration, noise, etc.), and is considered white, IID, homoscedastic, Gaussian and zero-mean with diagonal covariance $R_n$. With an abuse of notation, we lump all nuisance variability into the symbol $g_t$ (or, equivalently, we ignore contrast transformations $\kappa_t$ since they can be managed easily, as we show next), and write the

Figure 6: Effect of context: Imposing a scene prior that encodes co-occurrence helps reduce the likelihood of motorbike, aeroplane (left), and retrieve dining table and chair (right). The prior is learned from Pascal VOC2012.
image formation model formally as

\[ I_t(x_{ij}) = h_\rho(s, l, g_t) + n_t(x_{ij}). \]  \hspace{1cm} (16)

where \( x_{ij} = \pi(g_t x) \) for all visible \( x \in s \) and \( l \) is a latent (class) variable that is assumed to have measurable correlates in \( I \). For the entire visible portion of an object that occupies a region of space \( B \supseteq S \) that projects onto a region \( b_t \subset D \) of the image plane of the camera \( g_t \), we can write the image-formation process for all pixels \( x_{ij} \in b_t \) as

\[ I_t = h_\rho(s, l, g_t) + n_t, \]  \hspace{1cm} (17)

where the region \( B \) and its projection onto the image \( b_t \) are implicitly defined by \( s \) and \( g_t \). The data formation model above defines a probabilistic model of the form

\[ p(I_t|s, l, \rho, g_t) = \mathcal{N}(I_t; h_\rho(s, l, g_t), R_n) \]  \hspace{1cm} (18)

where the rendering function \( h_\rho \) is in general expensive to evaluate. Fortunately, this is not necessary, as we discuss next.

### 3.1 Abstracting the pixels

The model above allows us to evaluate the likelihood of an object of class \( l \) being present in a region \( s \) seen from vantage point \( g \) in image \( I \), by comparing the latter with an image hallucinated from the object \( \hat{I} = h_\rho(s, l, g) \) via the (log likelihood) \( \ell^2 \) distance between pixels \( d_2(I, \hat{I}) = \|I - \hat{I}\|^2 \). But to evaluate the likelihood, we do not need to simulate the image-formation process all the way to the pixels. At the very least, we want the comparison to be unaffected by nuisance variability. This could be done by defining a (geodesic) distance between functions \( \phi_n(I) \) and \( \phi_n(\hat{I}) \), for instance designed to be maximal \( \kappa \)-invariant descriptors. But more in general, we can design a descriptor for \( I \) that approximates the likelihood function (11) directly, by choosing among a parametric class of functions \( \phi(I) \) that, given an image region \( b \), produce a positive score \( \phi(I[b])|k \) for each class \( k \). This can be used to test whether an object of class \( l \) and shape \( s \) is present in the scene, by projecting it as seen from vantage point \( g \), \( b = \pi(gs) \), and testing the resulting score, as described in Sect. 3.3. Then, to compute AI in (10), given samples from \( p(l, s|\hat{g}_l|t, \hat{x}_l|t, y^t) \), we simply simulate actions \( u \), and the resulting (hallucinated) image regions \( b = \pi(\hat{g}_l|t, us) \) from which we can compute the scores \( \phi(I[b]|k) \), and hence AI empirically.

### 3.2 Controlling nuisance variability

The effects of some nuisance variables can be eliminated by replacing the data with a (maximal) invariant. For instance, contrast transformations \( \kappa \) can be factored out at no loss by replacing the intensity \( I \) at \( x_{ij} \) with a maximal contrast invariant \( \phi_n(I) \), for instance the gradient orientation: \( I(x_{ij}) \rightarrow y(x_{ij}) = \phi_n(I(x_{ij})) = \frac{\nabla I(x_{ij})}{\|\nabla I(x_{ij})\|} \), modeling the linearized approximation of \( n_t \) as an angular Gaussian. Alternatively, in some cases nuisances can be controlled. For instance, a robot can control the vantage point \( g_t \), to a certain extent, or its velocity \( u_t \) is \( se(3) \equiv \mathbb{R}^6 \), defined by

\[ \hat{g}_t = g_t u_t \]  \hspace{1cm} (19)

where \( u_t \) represents (translational and rotational) velocity. Assuming \( g_0 \) to be known (the global frame is arbitrary), the current pose \( g_t \) is determined by the history of velocities \( u^t \), up to drift and initial condition, so \( g_t = g(u^t) \). These choices are just illustrative: In general what is controllable depends on the sensor platform, for instance one may be able to control contrast transformations \( \kappa_t \) by changing the sensor gain or actively changing the illuminant.

---

7The same can, in theory, be done for viewpoint \( g_t \) and for any other nuisance that acts as a group on the data (viewpoint changes induce diffeomorphic deformations of the domain of the image away from occlusions [54]). When nuisances are not a group, one can still construct invariants, but they will not in general be maximal. They can still be sufficient statistics for certain classes of tasks, which leads us to the notion of sufficient invariants [55], which can be computed by marginalization and/or profiling (max-out).
Remark 4 (Informative nuisances?). Nuisances can be controlled to yield maximum information. This may seem like a paradox since nuisances are, by definition, not informative of the scene. While the value of nuisances is independent of object attributes, it affects images of the object, which can be informative of such attributes. A chair is a chair no matter where we are when we look at it. However, an image may be more or less revealing of it being a chair depending on vantage point. So, except for biased datasets, knowledge of pose \( g_t \) does not reduce our uncertainty in the class \( l \) absent the image. However, nuisances \( g_t \) can be controlled so as to produce images \( I_t \) that are as informative as possible of the identity \( l \) of objects in the scene.

### 3.3 Measurement process

At each instant \( t \), the raw data \( I_t \) is processed by “probing functions” \( \phi \), which can be designed or trained to be invariant to nuisance variability that cannot be actively controlled. The SLAM system processes all past image measurements \( I^t \) and inertial measurements \( u_t \), which collectively we refer to as \( y_t = \{\phi, (I_t), u_t\} \), where \( \phi_k(I_t) = \{\phi_k(I_t|s_{ij})\} \) is a collection of sparse contrast-invariant feature descriptors for \( N_i \) visible regions of the scene, and produces a joint posterior distribution of poses \( g_t \) and a sparse geometric representation of the scene \( x = [x_1, \ldots, x_{N(t)}] \), assumed uni-modal and approximated by a Gaussian:

\[
p_{\text{SLAM}}(g_t, x|y^t) \sim \mathcal{N}(g_t, x; \hat{g}_t[t], \hat{x}_t[t], P_{\{g, x\}|t})
\]

where \( x \in \cup_j S_j \), i.e., the scene is assumed to be composed by the union of objects, including the default class “background” \( l_0 \). This localization pipeline is described in [56], and is agnostic of the organization of the scene into objects and their identity. It also restricts \( x \) to a subset of the scene that is rigid, co-visible for a sufficiently long interval of time, and located on surfaces that, locally, exhibit Lambertian reflection.

To compute the marginal likelihood for each class \( l_k \in \{l_0, \ldots, l_K\} \), we leverage on a CNN trained discriminatively to classify a given image region \( b_j \) into one of \( K + 1 \) classes, including the background class. The architecture has a soft-max layer preceded by \( K + 1 \) nodes, one per class, and is trained using the cross-entropy loss, providing a normalized score \( \phi_{\text{CNN}}(l|I_{t|s_{ij}})[k] \) for each class and image bounding box \( b_j \).

We discard the soft-max layer, forgo class-normalization and rebalance the classes based on their frequency in the training set. The responses at the \( K + 1 \) nodes in the penultimate layer of the resulting network provide a mechanism for, given an image \( I_t \), quantifying the likelihood of each object class \( l_k \) being present at each bounding box \( b_j \), which we interpret the (marginal) likelihoods for (at least an instance of) each class being present at the given bounding box:

\[
\phi_{\text{CNN}}(l|I_{t|s_{ij}})[k] \simeq p(I_t|l_k, b_j).
\]

Note that the resulting CNN is not an object detector, but a likelihood function [53]: It is not a discriminant for a random variable \( l \) that can take one of \( K + 1 \) possible values, but instead a function of a \( K + 1 \)-dimensional (binary) vector \( l \), which is not normalized with respect to \( l \). This process induces a likelihood on object classes being present in the visible portion of the scene regions \( s_j \) and corresponding vantage points \( g_t \), via \( b_j = \pi(g_t s_j) \):

\[
\phi_{\text{CNN}}(l|I_{t|s_{ij}})[k] \simeq p(I_t|s_j, l_k, g_t)
\]

where, given an image \( I_t \), for each possible region in the scene \( s_j \) and vantage point \( g_t \), we can test the presence of at least one instance of each class \( l_k \) within. This allows the possibility that more than one object be present in the same region of space, for instance a person and a chair, or multiple instances of chairs. Here the visibility function is implicit in the map \( \pi \). If an object is not visible, its likelihood given the image \( I_t \) is constant/uniform. Note that this depends on the global layout of the scene, since the map \( \pi \) must take into account occlusions, so objects cannot be considered independently.

---

8Photo-collections exhibit significant dataset bias, as images uploaded by humans tend to be iconic and captured from object-dependent biased vantage points.
3.4 Dependencies and co-visibility

Computing the likelihood of an object being present in the scene requires ascertaining whether it is visible in the image, which in turn depends on all other objects, so the scene has to be modeled holistically rather than as an independent collection of objects. In addition, the presence of certain objects, and their configuration, affects the probability that other objects that are not visible be present. Furthermore, relations among objects may be the object of interest themselves, as in events or action recognition.

To capture these dependencies, we note that the geometric representation \( p(g, x|y^t) \) can be used to provide a joint distribution on the position of all objects and cameras \( p(g^t, x|y^t) \), which yields co-visibility information, specifically the probability of each point in \( x \) being visible by any camera in \( g^t \). It is, however, of no use in determining visibility of objects, since it contains no topological information: We do not know if the space between two points is empty, or occupied by an object void of salient photometric features. To enable visibility computation, we can use the point cloud together with the images to compute the dense shape of objects in a maximum-likelihood sense: \( s_j = \arg \max p(s_j|y^t, x, y) \) using generic regularizers. This can be done (Fig. 2) but may be overkill. An alternative is to approximate the shape of objects with a parametric family, for instance bounding boxes or ellipsoids, and compute visibility accordingly, also leveraging the co-visibility graph computed as a corollary from the SLAM system, and priors on the size and aspect ratios of objects.

To this end, we approximate

\[
\hat{p}_{g, x}(z^j|y^t) = p(z^j|y^t, g_t, x) \approx \prod_j p(z_j|y^t, g_t, x, z^{-j})
\]

where \( z^{-j} \) indicates all objects but \( z_j \). Each factor is then expanded as

\[
p(s_j, l_j, x_j|y^t, g_t, x, z^{-j}) = p(s_j, l_j|y^t, g_t, x, s^{-j}, x^{-j}) P(l_j|y^t, g_t, x, l^{-j})
\]

where the first term is represented by swarms of Extended Kalman Filters (EKF) and the second by a point-mass filter (PMF). These provide samples from \( \hat{p}(z|y^t) \) in the RHS of \( (5) \), that are scored with the CNN to update the posterior. Note that visibility is accounted for in the likelihood model of the EKF that, based on the sparse representation of the scene \( x \) and the shape \( s^{-j} \) and pose \( x^{-j} \) of other objects within.

3.5 Scene context RNN

The last element needed to compute \( \hat{g} \) is a representation of the (abstract) scene \( \theta \), to be inferred using objects \( z^j \) up to \( j = j(t) \) seen at the current time \( t \). This is needed to produce sample objects, including those not yet seen. \( \hat{z}_{j+1} \sim \hat{p}(z_{j+1}|\hat{\theta}) \), for instance through a point estimate \( \hat{\theta} = \hat{\theta}(z^j) \) computed from all objects seen thus far. Rather than postulating a model for how the true scene generates objects, and then using it to infer the scene from observed ones, we postulate a model for the object predictor directly. Specifically, we assume that a point estimate of the scene predicts objects deterministically through a parametrized function \( \psi \)

\[
\hat{z}_{j+1} = \psi(\hat{\theta}_{j+1})
\]

and that the scene estimate \( \hat{\theta} \) is a parametrized function \( \phi \) of all past objects \( z^j \) but updated incrementally via

\[
\hat{\theta}_{j+1} = \phi(\hat{\theta}_j, z_j)
\]

as soon as each object \( z_j \) becomes manifest, starting from an initial condition \( \theta_0 \). We represent the scene \( \theta \) abstractly as a collection of \( L \) sets of hidden states (“layers”) \( \{\theta^l\}_{l=1}^L \) with constant dimension, and assume

\footnote{For instance, seeing a keyboard and a monitor on a desk affects the probability that there is a mouse in the scene, even if we cannot see it at present. Their relative pose also informs the vantage point that would most reduce the uncertainty on the presence of the mouse.}

\footnote{In an active perception setting, given structured illuminator, one could take a control action to reduce uncertainty and resolve the conundrum.}
that \( \hat{z} \) can be predicted as a linear function of the last layer through an unknown matrix \( W_L \) followed by a non-linearity, for instance SoftMax:

\[
\psi(\theta) = \text{SoftMax}(W_L \theta^L).
\]  

(27)

Similarly, we assume that the function \( \phi \) operates at each layer through linear combinations of the previous layer at the same index \( j \) and the same layer at the previous index, followed by a non-linearity \( \sigma \), for instance arctan:

\[
\phi(\theta, z_j^{l+1}) = \sigma(W_l \theta_j^l, W_{l+1} \theta_j^{l+1}, W_0 z_j) = \theta_j^{l+1}
\]  

(28)

where the matrices \( \{W_l\}_{l=0}^L \) do not depend on \( j \), and consist of unknown parameters that must be inferred from \( z^j \). This can be done in a number of ways, for instance minimizing the prediction error or cross-entropy loss, corresponding to an Elman RNN:

\[
\hat{W}(z^j) = \arg \min_W \sum_{i=1}^j \ell(z_i - \psi(W_L \hat{\theta}_i^L)) \quad \text{subject to} \quad (28)
\]  

(29)

where \( \ell \) is a loss function of choice, that has a semantic component \( \hat{l} - l \), which is hand-designed (the cost of mispredicting a couch as a chair is not as high as confusing a chair with a person) or obtained empirically from classification error of base detector on benchmark datasets, and a geometric component, including relative pose \( \hat{x}_j - x_j \) and self-intersection \( s_j \cap s_i \), that can be also learned empirically from filter performance, along with the filter’s probability of missed detection and false alarm. In the absence of ground truth objects \( z_i \) one can model directly the predictive density using a variational autoencoder [36].

4 Experiments

4.1 Implementation

SLAM is borrowed from [56, 16]. The CNN is implemented by modifying stock code [45] by removing the classification layer and normalization. The EKF and PMF are standard textbook material and the RNN is trained assuming manually labeled scene bounding boxes.

4.2 Empirical evaluation

Fig. 5 shows representative visualization of MAP estimate of object pose, encoded by a gravity-aligned ellipsoid (here visualized as a bounding box), and the section of the point-cloud within. In addition, topology is estimated (dense surface, Fig. 2), and the identity as inferred. Although objects are assumed static (or moving slowly) in the scene, camera motion induces a non-trivial dynamics in the images of objects, that would have to be tracked if object models were represented on the image. However, having represented them in the scene allows us to predict the (image) motion of objects, as shown in Fig. 7, even when the object is completely occluded. We note that some datasets include moving objects (people) that, however, move slowly relative to the data sampling rate, and therefore can be captured even assuming a static scene (Fig. 5).

Figure 7: Left-to-right: A chair is detected by the CNN (top) and initialized in the filter (bottom); as the camera moves behind a monitor, “TV screen” is detected; whereas the chair disappears from the instantaneous frame (top); yet, it persists in the representation (right-hand-side, green box), so the system can predict its projection, and re-establish visual correspondence after it re-appears (right).

For the purpose of validation, we also tested components of our system on the KITTI dataset [20]; representative results are reported in Fig. 8.
5 Discussion

Objects exist in the scene, not on the image. Therefore a CNN that returns bounding boxes on the image cannot find objects; it can only find clumps of pixels.

Objects are persistent. They do not flicker in-and-out of existence. When they become occluded, we are aware of their presence behind the occluder. CNNs have no (ontogenic, or short-term) memory of the present scene; the only (phylogenic) memory is the dataset on which they are trained, which pertains to different scenes, encoded in the weights. They learn nothing from the present scene.

Our confidence on object presence and attributes grows over time as we accumulate evidence. A CNN does not become more confident as we accumulate evidence; it does not improve even as we track objects that we have previously identified with high confidence.

The likelihood function computed by a CNN is (an approximation of) a minimal sufficient invariant statistic of a single datum $y_{t+1}$, for the inference of semantic properties of the scene [53]. The posterior maintained by $(2)$ is a minimal sufficient invariant Bayesian statistic of all past data, for the task of inferring semantic properties of the scene.

We have described what we believe to be the first real-time system to infer a representation of a scene that encompasses geometric and photometric attributes (Gravity-referenced Euclidean point cloud with local contrast invariant descriptors), topological attributes (dense reconstruction of portions of the scene relevant to interaction), and semantic attributes from visual-inertial sensing. We focus on semantics, where persistent identities are maintained by a Bayesian filter, with likelihood function computed by a CNN, geometric and visibility relations are maintained both by the geometric and topological component of the representation, and co-occurrence relations are maintained by a graphical model.

We have made stringent and admittedly restrictive assumptions in order to keep our model viable for real-time inference. One could certainly relax some of these assumptions and obtain more general models, but forgo the ability to close the loop, which would render the inference of a representation for interaction purposes moot.

We stress that our model represents uncertainty at all levels, from the posterior on object classes (context) all the way to the pixel level (local descriptors as likelihood given gradient orientation). We also interpret the CNN not as an object detector operating independently in each image, but as a mechanism to compute the likelihood of persistent objects in the scene, given imaging data, and integrate it in a Bayesian filtering framework.

The model can in principle be extended to class-dependent object motion models, but at the cost of significant computational cost.
Acknowledgments

Contributors to this work include Pratik Chaudhari, Alessandro Achille and Alessandro Chiuso for discussions leading to the development of the statistical framework, and Xiaohan Fei, Jingming Dong, Nikolas Karianakis Konstantine Tsotsos for its implementation and testing. Research supported by ONR, ARO and AFOSR.

References


