Construction of the Object Semantic Hierarchy

Changhai Xu[1] and Benjamin Kuipers[2] *

[1] Department of Computer Sciences
University of Texas at Austin
1 University Station, Austin, TX 78712, USA
changhai@cs.utexas.edu
[2] Computer Science and Engineering
University of Michigan
2260 Hayward Street, Ann Arbor, MI 48109, USA
kuipers@umich.edu

Abstract. An intelligent robot must be able to perceive and reason robustly about its world in terms of *objects*, among other foundational concepts. The robot can draw on rich data for object perception from continuous sensory input, in contrast to the usual formulation that focuses on objects in isolated still images. Additionally, the robot needs multiple object representations to deal with different tasks and/or different classes of objects [20]. We present the *Object Semantic Hierarchy (OSH)*, which consists of multiple representations with different ontologies. The OSH factors the problems of object perception so that intermediate states of knowledge about an object have natural representations, with relatively easy transitions from less detailed to more detailed representations. Each layer in the hierarchy builds an explanation of the sensory input stream, in terms of a stochastic model consisting of a deterministic model and an unexplained "noise" term. Each layer is constructed by identifying invariants to reduce the previous layer's noise term. In the final model, the scene is explained in terms of constant background and object models, and low-dimensional pose trajectories of the observer and the dynamic objects.
The object representations in the OSH range from 2D regions, to 2D planar components with 3D poses, to structured 3D models of objects. This paper presents the Object Semantic Hierarchy in detail, describes the current implementation, and presents evaluation results.

Keywords: Multiple Object Representations, Object Modeling, Object Tracking, 3D Pose Estimation

* This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by grants from the Texas Advanced Research Program (3658-0170-2007), and from the National Science Foundation (IIS-0413257, IIS-0713150, and IIS-0750011).

1 Introduction

In order to achieve high-level goals, an intelligent agent in the physical world requires knowledge of foundational domains such as Space and Objects. In these foundational domains, there are often several quite different ways to represent entities of interest, drawing on different *ontologies*, that is, classes of logical objects and relations. A *semantic hierarchy* is a collection of these different ontologies, arranged so that knowledge of the environment can be acquired in relatively small steps, and so that the knowledge exhibits "graceful degradation" when resources are limited.

The Spatial Semantic Hierarchy [17, 18] is one such semantic hierarchy, organized to represent knowledge of large-scale and small-scale space, as a mobile agent moves through it. In this paper, we present the *Object Semantic Hierarchy (OSH)*, which is a collection of representations for objects and their surrounding contexts. The OSH is sensor-independent: this paper focuses on the familiar case of visual sensing of objects, but similar methods can be used with laser range sensors [24–26].

The layers of the OSH are:

1. **Noisy world**: the high-dimensional sensory input;
2. **Static background**: a static model of the background, in which dynamic change is treated as noise;
3. **2D object in 2D space**: a blob with color statistics plus a collection of distinctive features, and the blob's time-variant shape;
4. **2D object in 3D space**: a small collection of 2D components, with their individual time-variant 3D poses;
5. **3D object in 3D space**: the same collection of components but with invariant relations among their 3D poses, and the time-variant 3D pose of the object as a whole.

Hierarchical object models are created by repeatedly constructing and refining stochastic models of the observation stream generated by the agent's sensors in the environment. Such a stochastic model has the form $z_t = M_t + \epsilon$, where M_t is a deterministic model explaining the contents of the observation stream, and $\epsilon = z_t - M_t$ is the residual between explanation and observation, interpreted as noise. At each level, new invariants are identified within the data described by ϵ, leading to a revised model M'_t and ideally a reduced level of noise $\epsilon' = z_t - M'_t$. In the end, the uncertainty in the sensor stream is factored into a collection of relatively simple models: the static background, the dynamic observer's pose, constant object models, dynamic object poses, and any remaining noise. The "blooming, buzzing confusion" of the initial pixel-level input is concisely explained in terms of a relatively small number of object-level concepts and relations.

The idea of the OSH is that early stages of analysis can robustly derive certain properties of the visual scene, that are then used as assumptions to make later processing layers simpler and more robust. When and if later layers fail, the

earlier layers still allow objects to be tracked in the image, until they are more accessible to the more sophisticated kinds of analysis.

This paper presents the Object Semantic Hierarchy in detail, describes the current implementation and evaluation of the layers of "Static background", "2D object in 2D space", and "2D object in 3D space" by assuming that the input image sequence is captured by a static camera and the object of interest is composed of a few (approximately) planar surfaces. The implementation of "3D object in 3D space" and extension to a dynamic camera and non-planar surfaces are ongoing work.

2 Related Work

Modayil and Kuipers [24, 25] developed a method whereby a learning agent can autonomously learn about object models, by detecting, tracking, and characterizing clusters of foreground "pixels" in the sensory stream. Their agent is a mobile robot that receives a stream of sensory information from a laser rangefinder. It is assumed that the agent has learned the structure of its sensory array using the methods of Pierce and Kuipers [28]. In our work we adopt the "model learning through tracking" strategy [24, 25] to build object models, but the input data is extended to camera images.

A lot of work has focused on learning object models from databases of static images under different viewpoints and different backgrounds [12, 37, 30, 1, 34, 27]. Our method differs from these in that we learn an object model from continuous sensory input of the object, which takes advantage of the fact that the object appearance does not change much between two consecutive frames and hence the feature correspondences are much easier to identify. In particular, unlike previous hierarchical object representations [1, 34, 27, 41, 9] which used only 2D object models, the OSH contains both 2D and 3D object models.

Object detection and tracking are two important steps in building the OSH. We adopt the method in [32, 33] to build the background model and detect moving objects. Various kinds of features can be used in object tracking such as color histogram, contour and affine invariant regional features [6, 15, 35]. In particular, distinctive point features have been widely used in object tracking, such as by the KLT method [31] or the SIFT matching method [21, 11]. While point features have many successful applications, maintaining feature tracks over many frames may be quite difficult [35, 42], especially when the input images are noisy. The KLT method is efficient, but it may suffer from the feature drift problem over a long sequence of images [42, 2, 10].

More robust tracking can be obtained by integrating edge/boundary features with point features [29, 36]. Our method similarly uses point and boundary features, but it differs in that only boundary features are used for permanent correspondence across the images and point features only maintain temporary correspondence. This allows us to achieve good tracking performance since the boundary features in general are more robust to image noise and tend to give

more accurate position estimation. In addition, our method does not assume known 3D object models.

Estimation of the poses of planar surfaces plays an important role in our work. Two images of the same 3D plane are related by a homography matrix [13, 22]. The plane pose, rotation and translation between the two camera spaces can be obtained by decomposing this matrix [22, 5, 19]. This method is fast, but it is sensitive to noise and may give more than one physically possible solution. Zelnik-Manor and Irani [40] derived constraints for multiple planes across multiple views to improve homography estimation. Nonlinear optimization method using multiple images [14, 22] can be used to improve the homography decomposition method, but it requires good initializations which are hard to guarantee in practice. We develop a new probabilistic method for plane pose estimation to overcome these problems.

3 The Object Semantic Hierarchy

The Object Semantic Hierarchy is a hierarchical computational model of the background world and the foreground objects, consisting of multi-layer representations.

Layer 0: Noisy world

The agent perceives its environment through a high dimensional pixel-level sensory stream. In this layer, everything is considered as noise.

$$z_t = \epsilon_0 \tag{1}$$

where z_t is the sensor input at time t, and ϵ_0 represents also the sensor input but treats it as noise.

Layer 1: Static background

In its learning process, the agent starts by constructing a constant model of the background world, treating any foreground objects as noise.

$$z_t = G_1(b, x_{rt}) + \epsilon_1 \tag{2}$$

where b is the static background model, x_{rt} is the agent's observing pose, G_1 is a function mapping the background model b to a 2D image given the observer's pose x_{rt}, and ϵ_1 represents the actual discrepancy between the prediction of the model $G_1(b, x_{rt})$ and what is actually observed z_t. The background b could be a pixel-level 2D model, or a more sophisticated 3D model. The problem of simultaneously identifying the background model b and the state trajectory x_{rt} given the sensor stream z_t is the problem of Visual SLAM. Good Visual SLAM algorithms can be found in [7, 16]. In this paper we only consider the case where the observing pose is static.

In the special case with a fixed observing pose, the above equation is reduced to

$$z_t = G_1(b) + \epsilon_1 \tag{3}$$

where a good candidate for b is a pixel-wise statistical background model [38, 32, 33].

In both cases, all changes due to dynamic objects are considered as noise.

Layer 2: 2D object in 2D space

In order to detect foreground objects, we identify dynamic pixels as portions of the sensor image that violate the static background assumption, found by clustering non-zero pixels in ϵ_1. Trackable clusters of dynamic pixels contribute new, larger-scale entities, that is, 2D objects.

$$z_t = G_2(b, y_t, x_{rt}) + \epsilon_2 \tag{4}$$

where $y_t = \{y_{1t}, ..., y_{n_o t}\}$ in which each $y_{it} (1 \leq i \leq n_o)$ represents a 2D object, G_2 is a function that maps the 2D object models y_t and the static background b to an image under the observing pose x_{rt}, and ϵ_2 is the remaining noise.

Now let's consider the single object case and denote its model as y_t. We represent the object as a constant 2D object model o, and a 2D time-variant shape s_t.

$$y_t = \{o, s_t\} \tag{5}$$

where o is the object's color statistics plus a list of distinctive features, and $s_t = \{e_1, ..., e_{n_{oe}}\}$ is an ordered list of basic shape elements which form the closed boundary of the object. Candidates for shape elements are line segments or basis B-splines.

The distinctive features in o can be local point features such as SIFT [21], edges/contours, or regions such as MSER features [23].

Together with (4), we get

$$z_t = G_2(b, o, x_{rt}, s_t) + \epsilon_2 \tag{6}$$

where the sensor stream is explained as the constant background b and 2D object model o, dynamic observing pose x_{rt} and object shape s_t, plus the remaining noise.

Layer 3: 2D object in 3D space

In the time-variant object image enclosed by its shape, invariants are identified as a collection of constant 2D components, which are planar or approximately planar surfaces embedded in 3D space.

$$y_t = \{c, q_t\} \tag{7}$$

where $c = \{c_1, ..., c_{n_c}\}$ is the new constant 2D object model consisting of a constellation of components, and $q_t = \{q_{1t}, ..., q_{n_c t}\}$ is the corresponding 3D poses for each component in c at time t. The components' models are constant, but their poses change over time.

From (4) and (7) we have

$$z_t = G_3(b, c, x_{rt}, q_t) + \epsilon_3 \tag{8}$$

where G_3 is a function mapping b and c to an image under x_{rt} and q_t, and ϵ_3 is the remaining noise.

Now the sensor stream is decomposed into the static background b, constant 2D object model c, dynamic observing pose x_{rt} and components' individual poses q_t, plus the remaining noise.

A component is represented by a shape/boundary and the 2D image enclosed by the shape/boundary.

$$c_k = \{s_k^c, I_k^c\} \tag{9}$$

where $s_k^c = \{e_1^c, ..., e_{n_{ce}}^c\}$ is an ordered list of basic shape elements which form a closed contour, which has the same form as s_t in (5).

Layer 4: 3D Object in 3D space

We now begin to relate individual components to each other, to create a fixed 3D structure with a number of different components. The relation between the 3D poses of two components is invariant under the assumption of rigid object.

$$q_t = G_p(p, x_{ot}) + \epsilon_p \tag{10}$$

where x_{ot} is the object's 3D pose at t, $p = \{p_1, ..., p_{n_c}\}$ are the poses of the 2D components with respect to the object pose x_{ot}, G_p is a function that maps p to q_t under x_{ot}, and ϵ_p is the remaining noise. All the changing component poses in (8) are explained in terms of the changing pose of the 3D object as a whole.

We define the 3D object model in 3D space as

$$m = \{c, p\} \tag{11}$$

where $c = \{c_1, ..., c_{n_c}\}$, and p represents their constant relative poses.

By combining (8), (10) and (11) we get

$$\begin{aligned} z_t &= G_3(b, c, x_{rt}, G_p(p, x_{ot}) + \epsilon_p) + \epsilon_3 \\ &= G_4(b, m, x_{rt}, x_{ot}) + \epsilon_4 \end{aligned} \tag{12}$$

where G_4 is a function mapping the 3D object model m and the static background b to an image under the observing pose x_{rt} and the object pose x_{ot}, and ϵ_4 is the remaining noise.

From (12), we have actually explained the sensory stream in terms of (i) the static background model b and the constant 3D object model m, (ii) dynamic observing pose x_{rt} and dynamic object pose x_{ot}, plus the remaining noise. The only parameters that are time-variant are the low-dimensional poses x_{rt} and x_{ot}.

The transformation functions G_1, G_2, G_3, G_4 and G_p are summarized in Table 1. Each of these functions is a well-understood transformation matrix [14, 22]. Table 2 is a summary of the information that is acquired at different layers.

In the rest of the paper, we will also use the abbreviations BG, 2D2D, 2D3D and 3D3D to denote the layers 1-4 respectively.

Table 1. Summary of transformation functions

Function	Description
G_1	Given static background model and observer pose, predict sensor input.
G_2	Given background, observer pose, object color model and shape, predict sensor input.
G_3	Given background, observer pose, object component models and poses, predict sensor input.
G_4	Given constant background and 3D object models, and observer and object poses, predict sensor input.
G_p	Given object pose in world frame, and component poses wrt object frame, predict component poses in 3D world frame.

Table 2. Acquired information in the OSH

Layer	Acquired information
BG	b - constant background model x_{rt} - dynamic 3D observer pose
2D2D	o - constant object color s_t - dynamic 2D object shape
2D3D	c - constant object components q_t - dynamic 3D poses of object components
3D3D	p - constant 3D poses of object components in object frame x_{ot} - dynamic 3D object pose in world frame

4 Implementation

In this paper we focus on constructing the BG, 2D2D, and 2D3D layers. Construction of the 3D3D layer is ongoing work. While various inference methods can be used, our implementation serves as an illustration example of the construction steps in the OSH.

4.1 Background Modeling

We consider the case where the agent's observing pose is fixed, and learn a pixel-level model of the static background by washing out noises due to dynamic changes. We adopt the Gaussian mixture model [32] to maintain the background image. Fig. 1(a) shows a typical frame in one of the test videos, and Fig. 1(b) shows the corresponding learned background image.

4.2 Foreground Extraction

Objects are initially individuated from the background based on motion. We use background subtraction to separate dynamic pixels from the static background.

(a) (b) (c) (d)

Fig. 1. (a) a video frame, (b) the learned static background image, (c) the learned 2D object model in 2D space which has a uniform color and a time-varying shape, (d) a reconstructed image through a projection of the static background, the 2D object color model and its current shape.

To label the connected dynamic pixels, we adopt the method proposed in [4]. This method uses a contour tracing technique to detect the external contours and possible internal contours. It runs in linear time, labeling the pixels and generating the boundaries at the same time.

For a labeled object, the boundary surrounding all the object pixels is defined as its shape. The constant model o consists of two parts: object color statistics and a set of distinctive features on the object.

$$o = \{clr, f^o\} \tag{13}$$

At each frame t, we calculate the object color clr_t^o as the average color for all pixels belonging to the object. Then from the history of clr_t^o we maintain a Gaussian distribution $clr = \{\mu_{clr}^o, \sigma_{clr}^o\}$. We also detect local point features at each frame, describe them with the SIFT descriptors [21], and store them in the feature set f^o. In the following frame, new features are detected and matched to the stored features in f^o. New features that do not have good matches are added to f^o, and the others are used to update existing features. Fig. 1(c) shows a frame of the detected 2D object with a uniform color and a temporal shape. Fig. 1(d) shows the corresponding reconstructed image.

4.3 Component Tracking

The boundary for a component is detected by searching for contours that are closed and composed of a sequence of line segments, within the moving region detected by background subtraction. This initialization may need user's interactions when the component is in a noisy background. Once the boundary is detected, a tracker is assigned to the component, and tracks the component automatically over time.

We use the KLT method [31] to track point features. The feature correspondence between two adjacent frames $t-1$ and t is used to predict the component boundary location at time t, based on the already-known boundary location at time $t-1$. The detected features at time $t-1$ and the tracked features at time t are related by a planar homography transformation H_{at} [14, 22].

Let the component boundary at time $t-1$ be s^c_{t-1}, then we have $\hat{s}^c_t = H_{at} s^c_{t-1}$, where $\hat{s}^c_t$ is the predicted boundary at time t. We then update the predicted boundary to fit the observed data by matching line segments in their neighborhood areas. Around each line segment on the predicted boundary $\hat{s}^c_t$, a local interest region is formed in the image at time t. Within this rectangle, candidate line segments are detected using the Hough transform [8] after the Canny edge detection process [3], and the best matched line segment is used to correct the predicted line segment. From at least four pairs of matched line features, another homography matrix H_{bt} is obtained [13], and the component boundary at time t is finally updated as $s^c_t = H_{bt} H_{at} s^c_{t-1}$.

Discussion. In our hybrid tracker, point features maintain only temporary correspondence between each two adjacent frames, while line features maintain the permanent correspondence across all the frames for the tracked component. The Hough transform is applied only within the local interest regions of the predicted boundary. In general the KLT tracking is fairly accurate between adjacent frames, so the interest regions are typically small such that the computational cost for boundary correction is low.

4.4 3D Component Pose Estimation

The frame sequence is numbered as $1, \ldots, t$. We also denote a certain frame as frame 0 which is called the reference frame. The world space is chosen to be aligned with the camera space at frame 0. We define a component space, where the x-axis is arbitrarily chosen on the component, the z-axis is along the direction of the component normal, and the origin can be any arbitrary point on the component.

At time t, let the translation and rotation from the component space to the camera space be T_t and $R_t = (R_{1t}\ R_{2t}\ R_{3t})$. A point $P^c = (P^c_x, P^c_y)^T$ on the component and its image coordinates $p_t = (p_{ut}, p_{vt})^T$ are related by

$$\lambda^P \begin{pmatrix} p_{ut} \\ p_{vt} \\ 1 \end{pmatrix} = (R_{1t}\ R_{2t}\ R_{3t}\ T_t) \begin{pmatrix} P^c_x \\ P^c_y \\ 0 \\ 1 \end{pmatrix} = (R_{1t}\ R_{2t}\ T_t) \begin{pmatrix} P^c_x \\ P^c_y \\ 1 \end{pmatrix} = H_t \begin{pmatrix} P^c_x \\ P^c_y \\ 1 \end{pmatrix} \quad (14)$$

where λ^P is the point depth in the camera space, and H_t is a homography matrix that maps points from the component plane to the image plane.

Since R_t is a rotation matrix, it satisfies $\|R_{1t}\| = \|R_{2t}\| = 1$ and $R_{1t} \perp R_{2t}$. Equivalently we have the following constraints,

$$\|H_{1t}\| - \|H_{2t}\| = 0, \qquad H^T_{1t} H_{2t} = 0 \quad (15)$$

To estimate the component pose in an image sequence, a key step is to estimate its normal N_0 in the reference frame. We represent N_0 in a spherical coordinates as

$$N_0 = (sin\theta^N_0 cos\phi^N_0, sin\theta^N_0 sin\phi^N_0, cos\theta^N_0)^T \quad (16)$$

where $\theta_0^N \in [0, \pi/2]$ and $\phi_0^N \in [0, 2\pi)$ are the normal parameters.

Our goal is to estimate the probability density function $Pr(\theta_0^N, \phi_0^N | z_{0:t})$. By applying Bayes' theorem and assuming independent observations, we have

$$\begin{aligned} Pr(\theta_0^N, \phi_0^N | z_{0:t}) &\propto Pr(\theta_0^N, \phi_0^N | z_0) Pr(z_{1:t} | \theta_0^N, \phi_0^N, z_0) \\ &\propto Pr(\theta_0^N, \phi_0^N | z_0) \prod_{k=1}^{t} Pr(z_k | \theta_0^N, \phi_0^N, z_0) \\ &\propto Pr(z_t | \theta_0^N, \phi_0^N, z_0) Pr(\theta_0^N, \phi_0^N | z_{0:(t-1)}) \end{aligned} \tag{17}$$

Based on the constraints in Eq. 15, we design $Pr(z_t | \theta_0^N, \phi_0^N, z_0)$ as

$$Pr(z_t | \theta_0^N, \phi_0^N, z_0) = \gamma(1 + e^{-\frac{2\alpha_1 | \|H_{1t}\| - \|H_{2t}\| |}{\|H_{1t}\| + \|H_{2t}\|} - \frac{\alpha_2 | H_{1t}^T H_{2t} |}{\|H_{1t}\| \|H_{2t}\|}}) \tag{18}$$

where α_1 and α_2 are user-determined positive constants, and γ is a constant normalizing term. Note that H_{1t} and H_{2t} are functions of θ_0^N, ϕ_0^N, z_0, and z_t. Intuitively, the better the constraints are satisfied, the higher the probability that is assigned to the likelihood function.

Once N_0 is estimated, the component poses in the image sequence can be obtained accordingly (see [39] for details).

Discussion. While the conventional homography decomposition method takes two input frames and provides two physically possible solutions, our method is based on all the observations up to the current frame, and guarantees a unique Bayesian optimal solution. Since the proposed estimation method is recursive such that at each time step only the current observation is used to update the estimation, the computational cost at each time step does not grow with the increasing number of past frames.

5 Experiments

We collected a set of videos, each containing a moving object. The moving objects include a checker board (Dataset 1), a letter board (Dataset 2-6), a tea box (Dataset 7), and a hard drive box (Dataset 8).

We first test our tracking algorithm. Some typical tracked frames from the videos are shown in Fig. 2. To demonstrate the importance of integration of boundary information, we also tested our tracking algorithm where the boundary correction step is disabled. This test was done for two cases, (i) the same features are tracked over time, and (ii) features are detected at each frame and tracked only in the next frame. In either case (i) or (ii), tracking only point features worked fine for the checker board, because it is highly-textured and the corner points are very salient. But for all the other videos, tracking only point features was not sufficient. Some failed tracking frames are shown in Fig. 3.

We then test our 3D component pose estimation method. We obtained the ground truth data of the component normals for datasets 1-6, from two laser

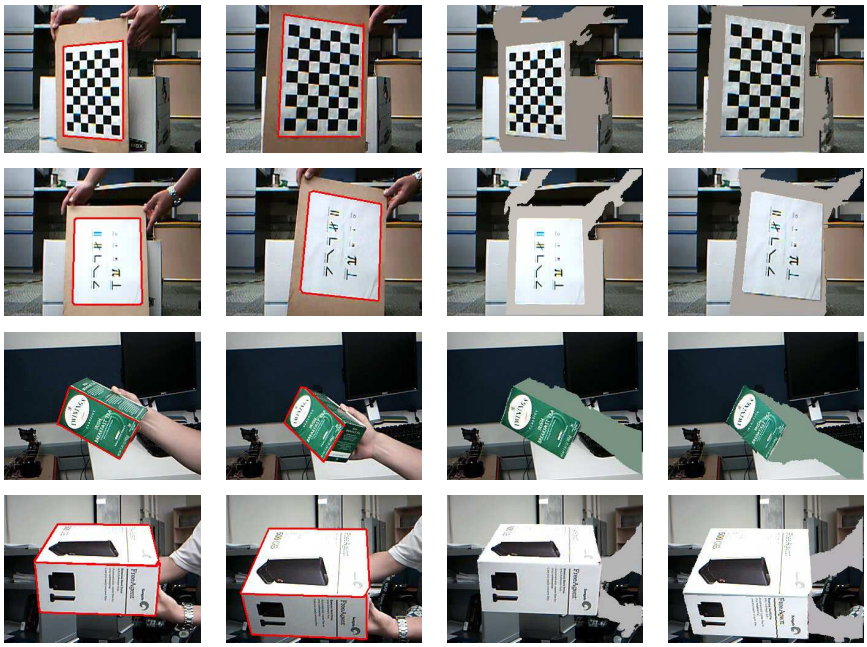

Fig. 2. Tracked components and reconstructed images (best viewed in color). The left two columns show the tracked components and the right two show the corresponding reconstructed images.

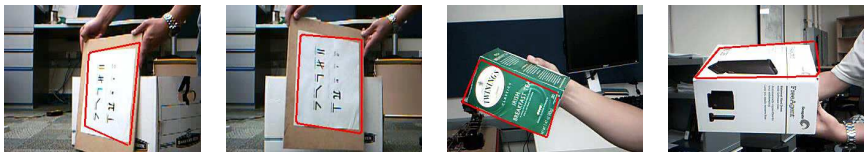

Fig. 3. Tracking failures when boundary correction (line features) is disabled. The failures are caused primarily by either accumulated feature position error or accumulated parameter estimation error.

rangefinders (horizontal and vertical). Because the camera was manually aligned with the laser sensors, we expect a small error in the ground truth data.

We compare the proposed method (PM) and the conventional homography decomposition (HD) method by measuring the estimation errors. The error is computed as the 2-norm of the difference between the estimated normal and the ground truth normal. While PM always gives a unique solution, the HD method in general provides up to two physically possible solutions. To show the robustness and accuracy of PM, we intentionally chose the solutions that are closer to the ground truth for the HD method. The quantitative errors are summarized in Table 3. Fig. 4 shows the comparisons for two individual datasets.

Table 3. Normal Estimation Errors for HD and PM

Error	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Average
HD	0.28	0.24	0.34	0.32	0.23	0.23	**0.27**
PM	0.13	0.18	0.27	0.36	0.16	0.26	**0.22**

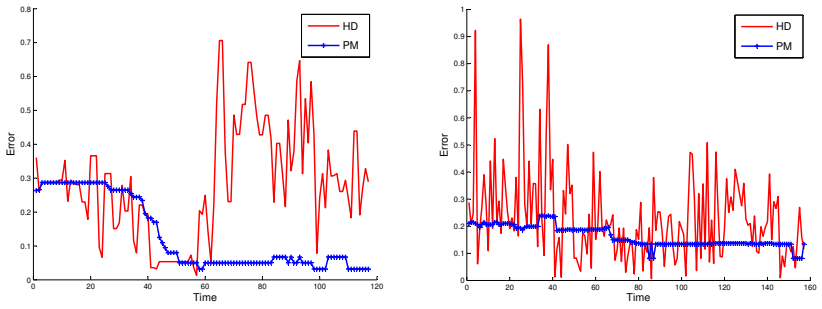

Fig. 4. Normal estimation errors for a checker board video and a letter board video. The error is computed as the 2-norm of the difference between the estimated normal and the ground truth normal.

After the models are constructed in the OSH, the agent is able to predict the sensor input by reconstructing an image through a projection of the constructed models (Fig. 2). We evaluate our work by investigating how the "noise" reduces from layer to layer, based on the difference between the input images and the reconstructed images. Fig. 5 shows comparisons between these errors for two test videos. The comparison results show the noise reduction trend from each layer to the following one.

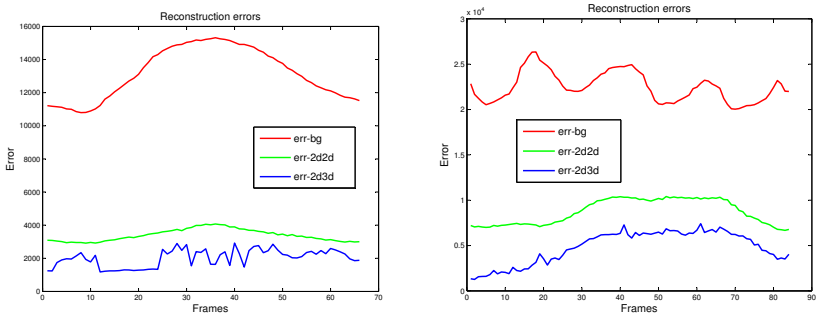

Fig. 5. Image reconstruction errors for the tea box video and the hard drive box video. Images are reconstructed at the BG, 2D2D and 2D3D layers. Noise is reduced at each layer compared with the previous layer.

6 Conclusion and Future Work

We presented the Object Semantic Hierarchy, which is a multi-layer representation for the background world and foreground objects.The input sensory stream is ultimately explained in a fairly simple representation which contains only constant models and a trajectory of low dimensional parameters. We presented our current implementation for the BG, 2D2D, and 2D3D layers.

We will complete construction of the 3D3D layer and investigate how multi-layer representations will help handle objects in tracking and recognition. Naturally, in the real world, not every object is composed of strictly planar surfaces. We will investigate the robustness of and extensions to our method when applied to curved surfaces. Current implementation of the OSH assumes a static observing pose, we will extend it to handle dynamic observing poses.

References

1. G. Bouchard and B. Triggs, "Hierarchical part-based visual object categorization," in *CVPR*, 2005.
2. F. Bourel, C. Chibelushi, and A. Low, "Robust facial feature tracking," *BMVC*, 2000.
3. J. Canny, "A computational approach to edge detection," *PAMI*, 1986.
4. F. Chang, C. Chen, and C. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Computer Vision and Image Understanding*, 2004.
5. D. Cobzas, M. Jagersand, and P. Sturm, "3D SSD tracking with estimated 3D planes," *Journal of Image and Vision Computing*, 2009.
6. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *PAMI*, 2003.
7. A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time single camera SLAM," *PAMI*, 2007.
8. R. Duda and P. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, 1972.
9. S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *CVPR*, 2007.
10. J. Gall, B. Rosenhahn, and H. Seidel, "Drift-free tracking of rigid and articulated objects," 2008.
11. I. Gordon and D. Lowe, "What and where: 3D object recognition with accurate pose," *Lecture Notes in Computer Science*, 2006.
12. K. Grauman and T. Darrell, "Unsupervised learning of categories from sets of partially matching image features," *CVPR*, 2006.
13. J. Guerrero and C. Sagüés, "Robust line matching and estimate of homographies simultaneously," *Pattern Recognition and Image Analysis: First Iberian Conference*, 2003.
14. R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge University Press, 2003.
15. M. Isard and A. Blake, "CONDENSATION - conditional density propagation for visual tracking," *IJCV*, 1998.
16. G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," *International Symposium on Mixed and Augmented Reality*, 2007.

17. B. Kuipers, "The Spatial Semantic Hierarchy," *Artificial Intelligence*, 2000.
18. B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," *ICRA, 2004*.
19. D. Kumar and C. Jawahar, "Robust homography-based control for camera positioning in piecewise planar environments," *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 2006.
20. N. Logothetis and D. Sheinberg, "Visual object recognition," *Annual Review of Neuroscience*, 1996.
21. D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.
22. Y. Ma, *An invitation to 3-D vision: From images to geometric models*. Springer Verlag, 2004.
23. J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *BMVC*, 2002.
24. J. Modayil and B. Kuipers, "Bootstrap learning for object discovery," *IROS, 2004*.
25. ——, "Autonomous shape model learning for object localization and recognition," *ICRA*, 2006.
26. ——, "Autonomous Development of a Grounded Object Ontology by a Learning Robot," *National Conference on Artificial Intelligence*, 2007.
27. D. Parikh and T. Chen, "Unsupervised learning of hierarchical semantics of objects (hSOs)," in *CVPR*, 2007.
28. D. Pierce and B. Kuipers, "Map learning with uninterpreted sensors and effectors," *Artificial Intelligence*, 1997.
29. M. Pressigout and E. Marchand, "Real-time 3d model-based tracking: Combining edge and texture information," *ICRA*, 2006.
30. S. Savarese and F. Li, "3d generic object categorization, localization and pose estimation," *ICCV*, 2007.
31. J. Shi and C. Tomasi, "Good features to track," *CVPR*, 1994.
32. C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *CVPR*, 1999.
33. ——, "Learning patterns of activity using real-time tracking," *PAMI*, 2000.
34. E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *ICCV*, 2005.
35. S. Tran and L. Davis, "Robust object trackinng with regional affine invariant features," *ICCV*, 2007.
36. L. Vacchetti, V. Lepetit, and P. Fua, "Combining edge and texture information for real-time accurate 3d camera tracking," *The 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2004.
37. M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," *ECCV*, 2000.
38. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *PAMI*, 1997.
39. C. Xu, B. Kuipers, and A. Murarka, "3D pose estimation for planes," *ICCV Workshop on 3D Representation for Recognition (3dRR-09)*, 2009.
40. L. Zeinik-Manor and M. Irani, "Multiview constraints on homographies," *PAMI*, 2002.
41. L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille, "Unsupervised structure learning: hierarchical recursive composition, suspicious coincidence and competitive exclusion," in *ECCV*, 2008.
42. T. Zinsser, C. Grassl, and H. Niemann, "Efficient feature tracking for long video sequences," *Pattern Recognition: 26th DAGM Symposium*, 2004.