Towards the Object Semantic Hierarchy

Changhai Xu
Department of Computer Science
University of Texas at Austin
1 University Station, Austin, TX 78712
changhai@cs.utexas.edu

Benjamin Kuipers
Computer Science and Engineering
University of Michigan
2260 Hayward Street, Ann Arbor, MI 48109
kuipers@umich.edu

***Abstract*—An intelligent agent, embedded in the physical world, will receive a high-dimensional ongoing stream of low-level sensory input. In order to understand and manipulate the world, the agent must be capable of learning high-level concepts. *Object* is one such concept. We are developing 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 structured to more structured 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 new invariants from the previous layer. 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 foreground objects.**

The object representations in the OSH range from 2D views, to 2D planar components with 3D poses, to structured 3D models of objects. This paper describes the framework of the Object Semantic Hierarchy, and presents the current implementation and experimental results.

I. Introduction

An intelligent agent 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 [15].

We are developing the Object Semantic Hierarchy (OSH) [31] to build a collection of object representations at different layers, motivated by the work of the Spatial Semantic Hierarchy (SSH) which consists of multi-level representations of large-scale space [13], [14], [1].

The framework of the OSH is shown in Fig. 1. The OSH has two types of layers: the object layers and the model layers. The object layers describe how the static background and each foreground object are individuated, and the model layers describe how the model for the static background and for each foreground object evolves from less structured to more structured representations.

In the object layers, the agent starts by constructing a constant model of the static background world, where foreground objects are treated as noise. Then the foreground objects are progressively individuated from the background and their models are constructed while they are tracked over time. Hierarchical 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 z_t, and $\epsilon = z_t - M_t$ is the residual between explanation and observation, interpreted as noise. At each layer, 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 sensory stream is factored into a collection of relatively compact representations: static background model, pose trajectory of the observer, constant foreground object models, pose trajectories of the foreground objects, 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

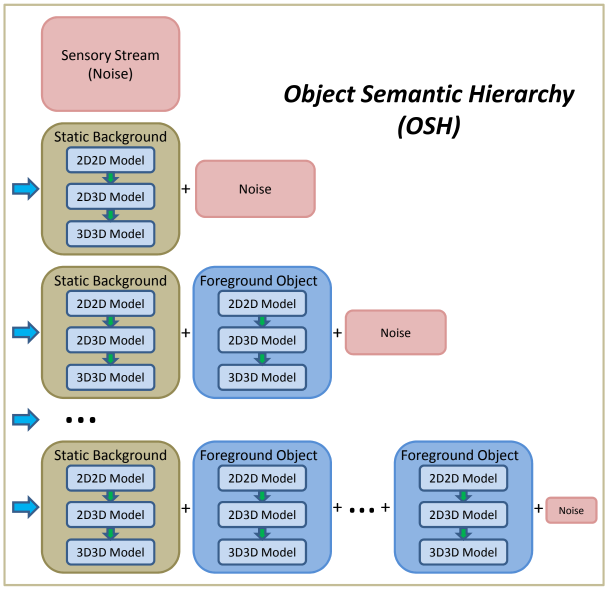

Fig. 1. The framework of the OSH (best viewed in color). The agent initially treats everything in the sensory stream as noise. By repeatedly identifying new invariants to reduce the noise, the agent progressively builds models for the background world and foreground objects. For the background world or each foreground object, the model evolves from 2D2D to 2D3D to 3D3D (see text for details).

concepts and relations.

In the model layers, the static background is treated as just another object. The construction of the static background model is taken in the same way as of any foreground object model. Each object model contains the following layers:

(a) **2D object in 2D space**: a sparse set of constant 2D object views, and the time-variant 2D object poses;
(b) **2D object in 3D space**: a small collection of constant 2D components, with their individual time-variant 3D poses;
(c) **3D object in 3D space**: the same collection of constant 2D components but with invariant relations among their 3D poses, and the time-variant 3D pose of the object as a whole.

The idea of the model layers 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 describes the framework of the Object Semantic Hierarchy, and presents the current implementation and experimental results.

II. Related Work

Modayil and Kuipers [19], [20] developed a method whereby a learning agent can autonomously learn 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 range-finder. It is assumed that the agent has learned the structure of its sensory array using the methods of Pierce and Kuipers [22]. In our work, we adopt the "model learning through tracking" strategy [19], [20] 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 [8], [30], [23], [4], [26], [21]. Our method differs from these in that object models are learned from continuous sensory input, where fine temporal granularity reduces the problem of correspondences between consecutive or nearby frames in the sensory stream and active control of perception allows the agent to obtain the right sort of information to build structured models from the sensory stream.

Marr and Nishihara [18] proposed a hierarchical representation for 3D object shape models and an approach to object recognition where the basic shape components are 3D cylinders. Biederman [3] also used a structural object description where basic components are 2D geons such as blocks, cylinders, spheres and wedges. In these representations, the discrimination among objects in the same category is difficult, since objects are modeled as a small set of components with simple shape descriptors. In the proposed OSH, detailed information is preserved as a normal view for each 2D planar surface, which makes our system more discriminative.

Various other 3D models have been proposed for object representation such as voxels, polygon meshes, or depth maps [24]. These representations describe objects at the pixel level of micro elements, whereas the goal of our work is to describe objects by identifying large-scale invariants. Both the constellation-of-planes in the OSH and the composition-of-geons in [3] are ways to address that goal.

Bouchard and Triggs [4] proposed a hierarchical model of object parts and subparts, with the object at the top level, and local image features at the bottom. While this method deals with a large number of local features, the number of parts at each level needs to be manually tuned and only three layers (object, part and feature) were tested in their experiments. Sudderth *et al.* [26] presented a hierarchical model for objects, the parts composing them, and the scenes surrounding them. Each object category has its own distribution over a set of parts which describes the expected appearance and location in the object centered coordinates, and parts are shared between objects. Parikh and Chen [21] presented hierarchical semantics of objects (hSOs) that capture relationships among multiple objects in a scene as observed by their relative positions in a collection of images. This hierarchy is a decomposition of the scene in terms of multiple objects. All these methods build a hierarchical representation for objects, but none of them includes 3D object models. Unlike these hierarchical object representations [4], [26], [21], [33], [7] which focused on building object models in the 2D image space, the OSH contains both 2D models in the image space and 3D models in the world frame.

III. 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.

As shown in Fig. 1, the OSH has two types of layers: the object layers and the model layers. In the object layers, the static background and foreground objects are progressively individuated by repeatedly identifying new invariants from the previous layer. In the model layers, the model for the static background and for each foreground object is refined from less structured to more structured representations.

The sensory stream is ultimately explained in a fairly simple representation which contains only constant background and object models, and low-dimensional pose trajectories of the observer and the foreground objects.

In the object layers, the agent starts by building a constant model of the static background world, where foreground objects are treated as noise. Once the background model is built as an explanation of its sensory stream, the agent continues to progressively individuate foreground objects by identifying new invariants within the discrepancy between the agent's explanation and observation.

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 is a random variable that represents 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.

$$z_t = G_1(M^b, x_t) + \epsilon_1 \tag{2}$$

where M^b is the static background model, x_t is the agent's observing pose, G_1 is a function mapping the background model M^b to a 2D image given the observer's pose x_t, and ϵ_1 represents the actual discrepancy between the prediction of the model $G_1(M^b, x_t)$ and what is actually observed z_t.

Layer 2: Foreground object 1

After the static background model is constructed, new invariants are identified within the data described by ϵ_1. The identified invariants contribute the first foreground object.

$$z_t = G_1(M^b, x_t) + G_2(M_1^o, y_{1t}) + \epsilon_2 \tag{3}$$

where M_1^o is the constant model for the first foreground object, y_{1t} is the object's pose, G_2 is a function mapping the object model M_1^o to a 2D image given y_{1t}, and ϵ_2 is the remaining noise. The plus sign is an operator that layers the foreground object image on top of the background image.

$$\vdots$$

Layer n: Foreground object n-1

At Layer n, the agent continues to identify new invariants within the noise term ϵ_{n-1} in the previous layer, and constructs a model for the $n-1^{th}$ foreground object.

$$\begin{aligned} z_t = G_1(M^b, x_t) + G_2(M_1^o, y_{1t}) + ... + \\ G_n(M_{n-1}^o, y_{(n-1)t}) + \epsilon_n \end{aligned} \tag{4}$$

where M_{n-1}^o is the constant model for the $n-1^{th}$ foreground object, $y_{(n-1)t}$ is the object's pose, G_n is a function mapping the object model M_{n-1}^o to a 2D image given $y_{(n-1)t}$, and ϵ_n is the remaining noise.

In the model layers, the static background model and each foreground object model have the following layers: 2D object in 2D space, 2D object in 3D space, and 3D object in 3D space. Here an *object* can be either the background world or any foreground object. In other words, the background world is treated as just another object. The object pose of the background world in the egocentric frame of reference is an implicit representation for the agent's observing pose in the allocentric frame of reference.

While the agent always tries to build all the model layers for an object, it can fall back to the already-constructed layers if at a certain layer the transition to the next is not feasible. Thus, the agent will still be able to work under lower-level models when higher-level models are not available.

Layer 2D2D: 2D object in 2D space

From the high-dimensional pixel-level object image stream, the agent identifies a sparse set of 2D object views as the object model v. The multi-view object representation has been shown very useful for object recognition [6], [29], [16].

The 2D object view model is described by

$$v = \{v_1, v_2, ..., v_{n^v}\} \tag{5}$$

where n^v is the number of the object views. The object views are connected by shared image features and/or the agent's motor signals.

The view model v should satisfy two constraints: (i) v is sparse compared to all the input object images, and (ii) v is complete such that any observed object image can be generated from v.

At each time step t, within the object model v, we locate the view that has closest observing pose with the new input image, by checking the overlapping ratio between each view and the input image. This located view is called the base view. The homography transformation between the input image and the base view, plus the pointer to the base view, is defined as the 2D object pose y_t^v. With this y_t^v, any observed object image can be reconstructed as an image transformed from the base view and the neighboring views of the base view.

Now we have

$$z_t = G_v(v, y_t^v) + \epsilon_v \tag{6}$$

where G_v is a function mapping v to an image under y_t^v, and ϵ_v is the remaining noise.

In Eq. 6, the object image stream is decomposed into the constant 2D object view model v, dynamic 2D object pose y_t^v, plus the remaining noise.

Layer 2D3D: 2D object in 3D space

Psychological experiments have shown that humans focus their study time on object views that are close to planar views (such as front, back, and side views) and ignore other views when actively interacting with objects [12].

Based on the 2D2D layer, we identify new invariants as a collection of constant 2D components, which are planar or approximately planar surfaces embedded in 3D space and are denoted by

$$c = \{c_1, ..., c_{n^c}\} \tag{7}$$

where n^c is the number of components, and each component in c is represented by its normal view. A normal view is defined as the component image that is observed when the optical axis is aligned with the normal of the component surface.

The object view model v in the 2D2D layer can then be described by

$$v = G_q(c, q) \tag{8}$$

where q are the 3D poses of the components appearing in the 2D views in v, and G_q is a function mapping c and q to v.

Let y_t^c denote the dynamic 3D component poses. The 2D object pose y_t^v in the 2D2D layer and the component poses q in Eq. 8 can both be represented as functions of y_t^c. Thus, by combining Eq. 6 and Eq. 8, we get

$$\begin{aligned} z_t &= G_v(G_q(c, q), y_t^v) + \epsilon_v \\ &= G_c(c, y_t^c) + \epsilon_c \end{aligned} \tag{9}$$

where G_c is a function mapping c to an image under y_t^c, and ϵ_c is the remaining noise. Note that y_t^c contains a history of the 3D poses for each individual component, where the 3D poses between different components are not related yet.

In Eq. 9, the object image stream is decomposed into the constant 2D object component model c, dynamic 3D component poses y_t^c, plus the remaining noise.

Layer 3D3D: 3D Object in 3D space

Compared to the multi-view representation in the 2D2D layer, a structured description of objects allows the agent to evaluate components and their relations independently [11]. In addition, a structured description tends to be more concise than the multi-view representation.

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.

$$y_t^c = G_p(p, y_t^o) \tag{10}$$

where y_t^o is the object's 3D pose at t, $p = \{p_1, ..., p_{n^c}\}$ are the 3D poses of the components with respect to the object pose y_t^o, and G_p is a function that maps p to y_t^c under y_t^o. All the changing component poses in Eq. 9 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

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

where both c and p are constant.

By combining Eq. 9, Eq. 10, and Eq. 11, we get

$$\begin{aligned} z_t &= G_c(c, G_p(p, y_t^o)) + \epsilon_c \\ &= G_o(o, y_t^o) + \epsilon_o \end{aligned} \tag{12}$$

where G_o is a function mapping the 3D object model o to an image under the 3D object pose y_t^o, and ϵ_o is the remaining noise.

In Eq. 12, the object image stream is decomposed into the constant 3D object model o, dynamic 3D object pose y_t^o, plus the remaining noise.

The transformation functions G_v, G_c, G_o, G_q and G_p are summarized in Table I. Each of these functions is a well-understood transformation matrix [9], [17]. Table II is a summary of the information that is acquired at different layers.

In the model layers, we have described object poses as y_t^v, y_t^c and y_t^o in the agent's egocentric frame of reference. For the static background as a special object, y_t^v, y_t^c and y_t^o are implicit representations for the agent's observing pose in the allocentric frame of reference. In later discussion, we will use x_t^v, x_t^c and x_t^o to denote the agent's observing pose in different model layers in the OSH (Fig. 2).

TABLE I
SUMMARY OF TRANSFORMATION FUNCTIONS

Function	Description
G_v	Given the constant 2D object view model and the dynamic 2D object pose, predict sensor input.
G_c	Given the constant 2D component models and the dynamic 3D component poses, predict sensor input.
G_o	Given the constant 3D object model and the dynamic 3D object pose, predict sensor input.
G_q	Given the models of a set of 2D components and their poses, predict 2D views in image space.
G_p	Given 3D object pose, and 3D component poses wrt object frame, predict 3D component poses in world frame.

TABLE II
ACQUIRED INFORMATION IN THE OSH

Layer	Acquired information
2D2D	v - constant 2D object view model y_t^v - dynamic 2D object pose
2D3D	c - constant 2D object component models y_t^c - dynamic 3D component poses
3D3D	o - constant 3D object model y_t^o - dynamic 3D object pose

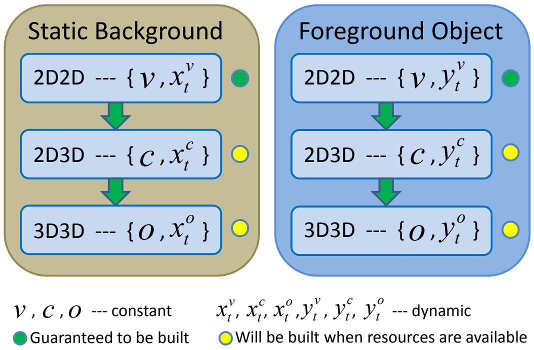

Fig. 2. The model layers in the OSH. The background model and foreground object models $\{v, c, o\}$ are constant. The observer's pose $\{x_t^v, x_t^c, x_t^o\}$ and the foreground object poses $\{y_t^v, y_t^c, y_t^o\}$ are time-variant. The 2D2D model is guaranteed to be built, and the 2D3D and 3D3D models will be built when the agent has necessary resources available.

IV. CONSTRUCTION OF THE OSH

In this section, we describe our current implementation of the OSH, and present experimental results.

A. Static Background Model

The static background environment usually has more complex structure than foreground objects. While our ultimate goal is to build a full 3D3D model for the background environment, we only seek to construct a 2D2D background model in this paper.

We previously used the Gaussian mixture model (GMM) [25] to build a pixel-level background image and wash out noises due to dynamic changes, where the observing pose is assumed fixed [31].

When the observing pose is dynamic, the variant GMM method by stitching images in a single view [5], [27], [2], [28], [10] suffers from accumulated image registration error. Instead, we identify a sparse set of views as the 2D2D background model v, where these views are connected by homography transformations. The homography transformations are obtained based on the robot's motor signals and the shared image features between overlapping views. When a new input image has small or no overlapping with all views in the 2D2D background model, the image is added to the model as a new view. If the input image has a large overlapping with a view in the model, the image is used to update the view.

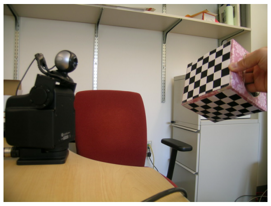

Fig. 3. A box is moving around in front of a webcam mounted on a pan tilt unit.

At each time step, within the 2D2D background view model, we find the base view that has the largest overlapping ratio with the new input image. Features are detected and matched between the input image and the base view. A homography matrix is calculated between these features. This homography matrix, together with the pointer to the base view, is the agent's observing pose x_t (here x_t is actually the observing pose x_t^v in the 2D2D layer in the OSH).

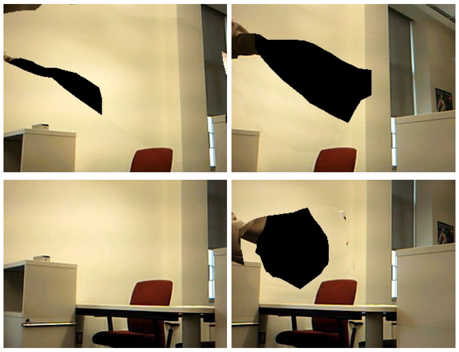

Fig. 4. 2D view examples in the 2D2D background model. The views correspond to different observing poses. The black holes in the images are due to permanent occlusion by foreground objects which are treated as noise.

Fig. 3 shows a simple robot which has a webcam mounted on a pan tilt unit. This setup is different from a general pan tilt camera in that the optical center of the webcam will have translations when the pan tilt unit moves beyond the horizontal plane. The robot has access to its motor signals, that is, pan and tilt positions of the camera.

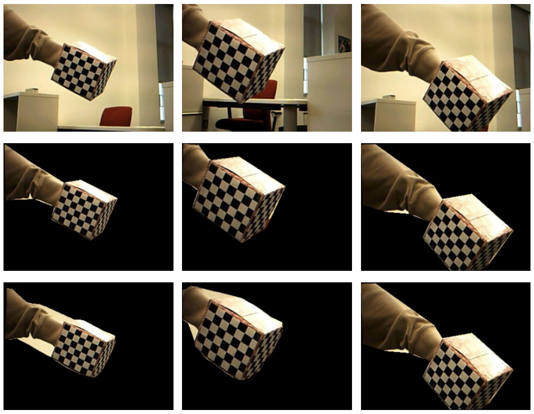

Fig. 5. Typical views in the 2D2D object model for a checker box. Rows 1-3 show the original images, the ground truth foreground objects, and the segmented 2D views for the foreground objects respectively.

To construct the 2D view model for the static background, the agent needs to be able to identify which part in each input image comes from the background (the remaining part is treated as noise). Motor signals allow the robot to predict the motion patterns of background features. In contrast, the motion patterns of non-background features will be different from the predictions because they have independent motions from the robot. This observation provides us a way to cluster image features based on their discrepancy with their predictions. Based on this observation, we first detect sparse features and label them as background features and non-background features. Then we propagate their labels to all pixels in the input image. This method segments an image into background pixels and non-background pixels based on only a few neighboring frames. It is robust to illumination changes, adapts fast to the environment, and does not suffer from accumulated image registration error.

Fig. 4 shows some view examples in the constructed 2D2D background model for the scenario in Fig. 3.

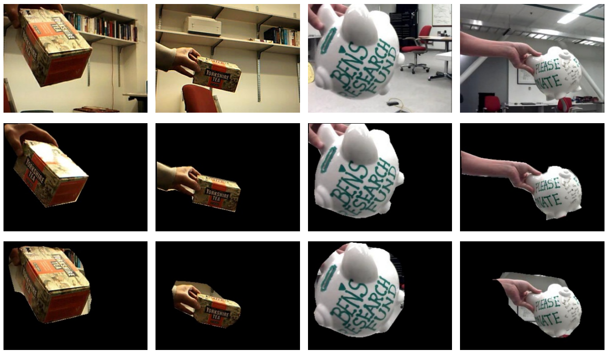

Fig. 6. 2D2D object models for a tea box and a toy pig (only a part of the 2D views are shown). Top row: original images, middle row: ground truth object views, bottom row: segmented 2D views for the foreground objects.

B. Foreground Object Model

a) 2D2D foreground object model: Once the background model is constructed, foreground object pixels can be individuated from the background. For each new input image,

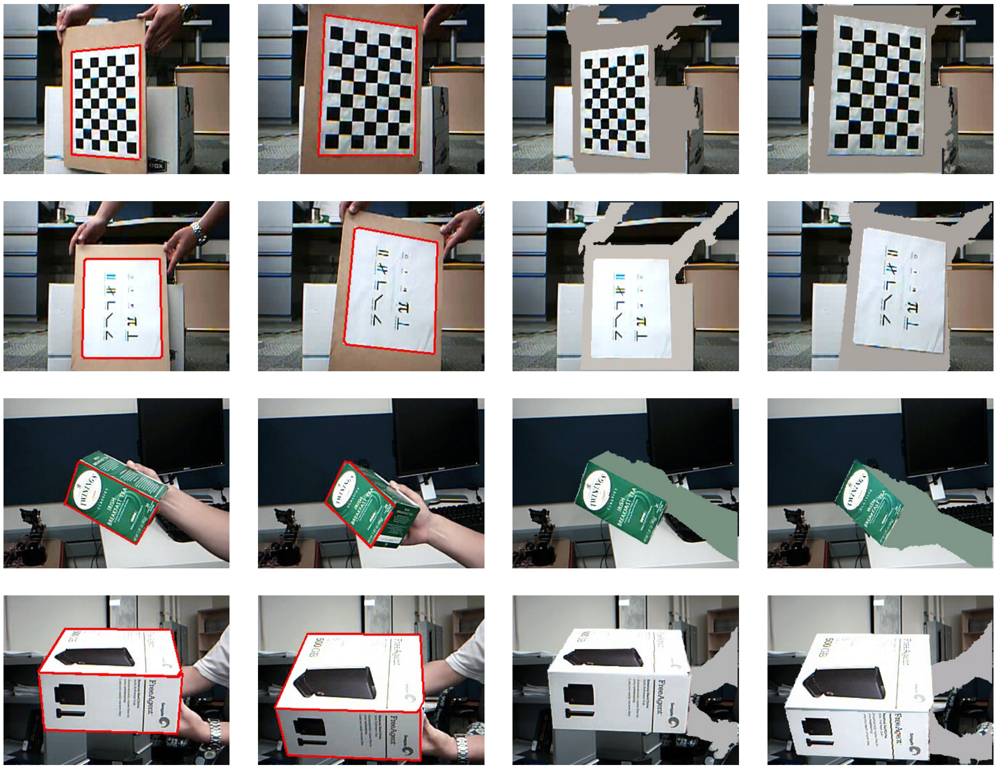

Fig. 7. Tracked components and reconstructed images. The left two columns show the tracked components and the right two show the reconstructed images at the 2D3D layer. The part that does not correspond to any component on the foreground object is shown in the average color of that part.

we detect sparse features and match them with the 2D2D background model. Those features that violate the background model are labeled as foreground features. Then the labels are propagated to all image pixels.

Within the 2D foreground object images where the background pixels have been filtered out, the 2D2D foreground object model v and the object pose y_t^v are obtained in a similar way as in the 2D2D background model construction. Fig. 5 shows some typical views in the 2D2D object model for a checker box which we have seen in Fig. 3. The 2D2D models for two other objects are shown in Fig. 6.

b) 2D3D foreground object model: In the foreground object image sequence, we track individual components c and estimate their 3D poses y_t^c. The tracking method uses both local point features and boundary features, and the pose estimation method (WINEP) provides an optimal solution based on all the observed frames (see [32] for details).

While the conventional homography decomposition method for plane pose estimation takes two input frames and provides two physically possible solutions, the WINEP method is based on all the observations up to the current frame, and guarantees a unique Bayesian optimal solution. Since this 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.

Fig. 7 shows some tracking examples and reconstructed images. Fig. 8 shows the estimation errors of the normals of a tracked component.

After the 3D poses y_t^c are estimated, the normal views for the tracked components c can be constructed accordingly. In Fig. 9 we show the normal views for a few components.

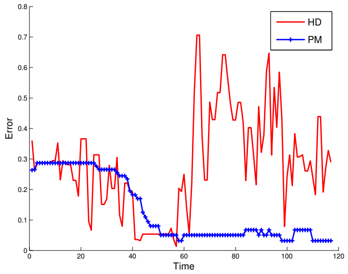

Fig. 8. Normal estimation errors for the conventional homography decomposition method (HD) and the WINEP method (PM). The error is computed as the 2-norm of the difference between the estimated normal and the ground truth normal. See [32] for details.

V. Conclusion and Future Work

We have 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 have described our current implementation of the background 2D2D model, foreground 2D2D model, and foreground

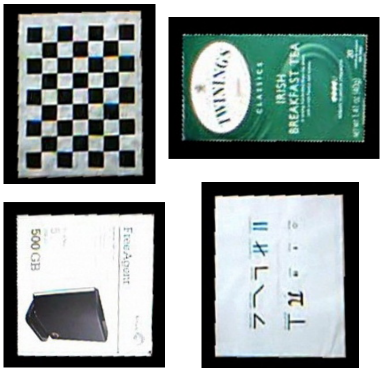

Fig. 9. The normal views for some tracked components. The components' 3D poses are estimated in the 2D3D layer, and their normal views are constructed accordingly.

2D3D model in the OSH. Our ultimate goal is to build full 3D3D models for both the background world and the foreground objects. Savarese *et al.* [23] proposed a method to represent 3D objects by linking together diagnostic parts from different viewing points, where parts are large discriminative regions and connected by their mutual homographic transformation. We will adopt this method in constructing 3D3D model construction.

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.

ACKNOWLEDGMENT

This work has taken place in the Intelligent Robotics Labs at the University of Texas at Austin and at the University of Michigan. Research in the Intelligent Robotics Labs is supported in part by grants from the National Science Foundation (IIS-0713150 to UT Austin and CPS-0931474 to UM) and from the TEMA-Toyota Technical Center to UM.

REFERENCES

[1] P. Beeson, J. Modayil, and B. Kuipers, "Factoring the mapping problem: Mobile robot map-building in the Hybrid Spatial Semantic Hierarachy," *International Journal of Robotics Research*, vol. 29, no. 4, pp. 428–459, 2010.

[2] A. Bevilacqua, L. Di Stefano, and P. Azzari, "An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a PTZ camera," in *IEEE Conference on Advanced Video and Signal Based Surveillance*, 2005, pp. 511–516.

[3] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychological review*, vol. 94, pp. 115–147, 1987.

[4] G. Bouchard and B. Triggs, "Hierarchical part-based visual object categorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 710–715.

[5] M. Brown and D. Lowe, "Recognising panoramas," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2003, p. 1218.

[6] H. Bulthoff and S. Edelman, "Psychophysical support for a two-dimensional view interpolation theory of object recognition," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, no. 1, p. 60, 1992.

[7] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[8] K. Grauman and T. Darrell, "Unsupervised learning of categories from sets of partially matching image features," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 19–25, 2006.

[9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[10] E. Hayman and J. Eklundh, "Statistical background subtraction for a mobile observer," in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, vol. 1, 2003, pp. 67–74.

[11] J. Hummel, "Where view-based theories break down: The role of structure in shape perception and object recognition," *Cognitive dynamics: Conceptual change in humans and machines*, pp. 157–185, 2000.

[12] K. James, G. Humphrey, T. Vilis, B. Corrie, R. Baddour, and M. Goodale, "Active and Passive Learning of Three-Dimensional Object Structure Within an Immersive Virtual Reality Environment," *Behavior Research Methods Instruments and Computers*, vol. 34, no. 3, pp. 383–390, 2002.

[13] B. Kuipers, "The Spatial Semantic Hierarchy," *Artificial Intelligence*, vol. 119, no. 1-2, pp. 191–233, 2000.

[14] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," *IEEE International Conference on Robotics and Automation*, vol. 5, pp. 4845–4851, 2004.

[15] N. Logothetis and D. Sheinberg, "Visual object recognition," *Annual Review of Neuroscience*, 1996.

[16] D. Lowe, "Local feature view clustering for 3D object recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001.

[17] Y. Ma, *An invitation to 3-D vision: From images to geometric models*. Springer Verlag, 2004.

[18] D. Marr and H. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc. of the Royal Society-London B*, 1978.

[19] J. Modayil and B. Kuipers, "Bootstrap learning for object discovery," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 742–747, 2004.

[20] ——, "The initial development of object knowledge by a learning robot," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 879–890, 2008.

[21] D. Parikh and T. Chen, "Unsupervised learning of hierarchical semantics of objects (hSOs)," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[22] D. Pierce and B. Kuipers, "Map learning with uninterpreted sensors and effectors," *Artificial Intelligence*, vol. 92, no. 1-2, pp. 169–227, 1997.

[23] S. Savarese and F. Li, "3d generic object categorization, localization and pose estimation," *International Conference on Computer Vision*, pp. 1–8, 2007.

[24] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 519–528.

[25] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, 1999.

[26] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *International Conference on Computer Vision*, vol. 2, 2005, pp. 1331–1338.

[27] R. Szeliski, "Image alignment and stitching: A tutorial," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, p. 104, 2006.

[28] P. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[29] S. Ullman, "Three-dimensional object recognition based on the combination of views," *Cognition*, vol. 67, no. 1-2, pp. 21–44, 1998.

[30] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," *European Conference on Computer Vision*, vol. 1842, pp. 18–32, 2000.

[31] C. Xu and B. Kuipers, "Construction of the Object Semantic Hierarchy," *Fifth International Cognitive Vision Workshop (ICVW-09)*, 2009.

[32] C. Xu, B. Kuipers, and A. Murarka, "3D pose estimation for planes," *ICCV Workshop on 3D Representation for Recognition (3dRR-09)*, 2009.

[33] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille, "Unsupervised structure learning: hierarchical recursive composition, suspicious coincidence and competitive exclusion," in *European Conference on Computer Vision*, 2008, p. 773.