Hierarchical Object Representation based on a Sparse Autoencoder Network

Anonymous ECCV submission

Paper ID ***

Abstract. We propose a method for constructing a pair of complementary hierarchical representations from 2D edge images and from view-based 3D data, respectively. As a learning algorithm, our model uses a sparse autoencoder network, which propagates information bottom-up, while offering an efficient inference mechanism across hierarchies, based on discretization of the responses at each layer. We introduce a probabilistic fusion model for inferring missing observations across modalities and for scene reconstruction by exploiting correlations between 2D and view-based 3D observations. The results obtained on the PaCMan dataset demonstrate the efficacy of our approach.

Keywords: Hierarchical Representation, Sparse Autoencoder Network, Depth Inference

1 Introduction

Visual object recognition and categorization has many potential applications, enabling scene interpretation, semantic visual retrieval, and cognitive robotics. Furthermore, efficient object representations can be useful in a number of robotic application domains, such as object manipulation by a robotic arm, including tasks such as loading and unloading a dishwasher, grasping objects or opening a door.

However, obtaining an efficient object representation is difficult due to high intra-class variation in shape, appearance and pose, but also due to illumination changes, occlusions or clutter. One approach useful for dealing with these limitations is to use a hierarchical object representation, which provides a compact representation on different levels of abstraction, provides object part shareability between different object categories and can cope with incompleteness produced by occlusions.

In recent years, many approaches have been proposed for object class detection using 2D features from a single-view, such as the deformable part model [1], but also multi-view based methods which are useful at predicting not only the category but also the pose of an object in a scene [2]. Furthermore, the availability of depth sensors such as Kinect had a great impact on the rapid development of algorithms for construction and acquisition of view-based 3D object models. Employing view-based 3D models for object recognition is useful, as they capture the intrinsic geometric structure of objects and provide a compact object representation [3]. Throughout this paper, by 2D data we mean intensity images, while an equivalent name of view-based 3D models in the literature is 2.5D models.

Provided that 2D and view-based 3D information are complementary, we aim to combine them in an efficient manner using a hierarchical representation, enabling improved recognition performance and inference of view-based 3D object parts given observed 2D parts.

There are different fusion methods available, such as early-level fusion [4], where different features are concatenated at an early stage, or decision-level fusion [5] in which recognition is performed individually for each input channel and the final decision is taken by combining the independent results, using different criteria such as taking the maximum or using a weighted average [6].

These methods are insufficient for our goal, as we aim to take advantage of the correlations between 2D and view-based 3D information not only at the beginning or at the end of the recognition process, but to model them in a hierarchical manner, by considering relationships between parts in the 2D and view-based 3D hierarchies. Even though there are works which consider the relationships between 2D and 3D visual information, to the best of our knowledge, there is no other work which investigates this problem using a hierarchical modeling approach.

Therefore, in this work, we propose using both 2D and view-based 3D primitives, constructed using hierarchical representations, in which the effect of low-level visual features on higher-level entities is learned, leading to detection of object models in a scene and cross-hierarchy inference.

The principles of hierarchy construction for both 2D and view-based 3D data are similar and are learned using sparse autoencoder networks. The relations between the two types of information are captured using an efficient probabilistic model. The model provides better inference, by reinforcing the evidence provided by one input channel with that from the other. Furthermore, it can be employed in the case of missing observations, inferring the most likely view-based 3D primitives from the observed 2D ones and vice-versa.

In the following section we present a review of related work, while in Section 3 we introduce our hierarchical model for object representation using the two available data types, 2D intensity-based shape information and 3D view-based depth data. Next, in Section 4 we present our method for capturing correspondences between 2D and view-based 3D primitives, while in Section 4.1 we introduce our proposed algorithm for inferring view-based 3D parts given observed 2D parts. The experimental results are introduced in Section 5 and conclusions are formulated in Section 6.

2 Related Work

One approach towards fusing 2D and 3D information consists of projecting a 3D shape into 2D and finding a set of meaningful multi-view 2D projections [7], an option supported by Fleming and Singh [8] who argue that human perception of 3D shapes is based on 2D observations. The relation between 2D and 3D is exploited for 3D shape segmentation. First, the 2D projections of a 3D shape are labeled, then they are transferred to the 3D shape via back-projection, while finally the integration of multiple labels is done for obtaining a unified segmentation. Projection of 3D information into 2D as a means to recognize 3D objects and their pose is not new in computer vision and has been investigated also by Cyr and Kimia [9]. This approach is justified by the simplification of the problem, obtained by working in a low-dimensional space and by making use of the bigger volume of 2D image data available in the online environment. The disadvantages of this method reside in the complexity of the process, which requires manual labelling of parts, an assumption regarding the upright orientation of objects, as well as a slightly worse performance on articulated objects in comparison with rigid objects.

Another approach for capturing 2D to view-based 3D correlations is described by Cheng et al. [10]. This work assumes that edge information in an image has a high probability of also being an edge in the depth map. Next, an image is divided into blocks and depth is assigned based on neighbourhood information and smoothed using bilateral filtering.

An adaptive fusion approach of 2D and 3D features is proposed by Arca et al. [4] with an application in face recognition. The integration of the two modalities is beneficial, as 2D features are more precise at capturing facial details, while 3D features are more robust under pose or lighting variations. The two types of features are fused at an early stage, while in the recognition process, only the most reliable features are used. Discarding of unreliable features is achieved using a matching criterion, which assigns weights based on a similarity measure. The same topic is investigated also by Soltana et al. [5], where the fusion strategy consists of an offline and an online weight learning process. In both cases the most relevant weights of all the scores for each sample in each modality is automatically selected.

3D object retrieval based on a 2D image represent another source of inspiration for linking 2D and 3D information. Eitz et al. [11] propose using a sketch representation based on optimized Gabor filters, followed by a bag-of-words approach. The feature extraction technique can be further refined into a hierarchical representation using concepts proposed by Dai et al. [12], which guide the learning of a dictionary of templates in an unsupervised manner. Each template is a composition of Gabor wavelets which are allowed to shift their location and orientation relative to each other. The invariance property is achieved using the active basis model [13].

Recovering 3D depth from a single image for scene understanding represents an important topic in computer vision and was investigated by Tian et al. [**?**] using a deep learning model, by Saxena et al. [**?**] by means of a Markov Random Field(MRF) model which captured the relationships between different parts of an image and by Konrad et al. [**?**] who proposed a depth sampling method, by fusing the corresponding depth images of the top k-nearest neighbour images of the query image.

The review of work dealing with 2D and view-based 3D information showed that fusing them is beneficial, while great care needs to be given to challenging issues such as conflicting data, outliers and noise, or data registration and correlation assessment. Without a proper pre-analysis of all these issues, the success rate of the fusion algorithm can be affected.

In the next section, we introduce our proposed hierarchical object representation model.

3 Hierarchical Object Representation

Our goal consists of matching 2D and view-based 3D parts in order to permit inference of 2D parts from observed view-based 3D parts and vice-versa and also to achieve better object recognition and/or object pose recognition accuracy.

In order to obtain 2D and 3D-view based object parts at different levels of granularity, we propose using a hierarchical model which enables binding of parts at different layers. A hierarchical representation has advantages such as capturing exponential variability in a compact way and allowing fast inference.

The reason for building a new hierarchical model, instead of using existing models, is that we need a consistent method for forming parts for both modalities (2D intensity and 3D view-based), parts which need to share the same receptive field, otherwise correlations might only partially reflect the real relationships between the two information channels.

The principles for forming the hierarchical object representations are consistent for both 2D and 3D view-based data, where the size of the receptive field is gradually increased at each layer, up to the top layer which is sufficiently large to represent the whole object. Therefore, we will only describe the learning process for one type of data, namely 2D information. To obtain an object representation robust to illumination changes we use 2D shape information, obtained by applying an edge detection algorithm to the 2D intensity images.

The input used for learning the 2D hierarchical representation is a set $\mathcal{Z}$ of training edge images $Z^i \in \mathcal{Z}, i = 1, \dots, |\mathcal{Z}|$, represented[1] as a set of vectorized patches Z_j, i.e., $Z^i = \{Z_1, \dots, Z_j, \dots, Z_{|Z^i|}\}$, where each image patch $Z_j = [z_{j1}, \dots, z_{jk}, \dots, z_{jn^2}]^T \in R^n$ is centered around an edge pixel z_{jk}. The size n^2 of an image patch is the size $F^1 = n^2$ of the corresponding receptive field at the first layer, while the number of patches of each edge image $|Z^i|$ is given by the number of edge pixels divided by the stride. The total number of patches used in the training phase is given by $|Z_*| = \sum_{i=1}^{|\mathcal{Z}|} |Z^i|$.

3.1 Learning algorithm

The first layer is learned using a sparse autoencoder algorithm, which is one approach to automatically learn features from unlabeled data and proved to be

[1] Here and in the following, superscripts always constitute upper indices; their use as powers will be indicated explicitly unless unambiguously clear from the context.

suitable not only for computer vision, but also for a range of problems including text, audio, etc. An autoencoder network is an unsupervised learning algorithm which applies backpropagation, setting the target values to be equal to the input values, as depicted in Figure 1 below. Therefore, one practical application is to use it for reconstruction.

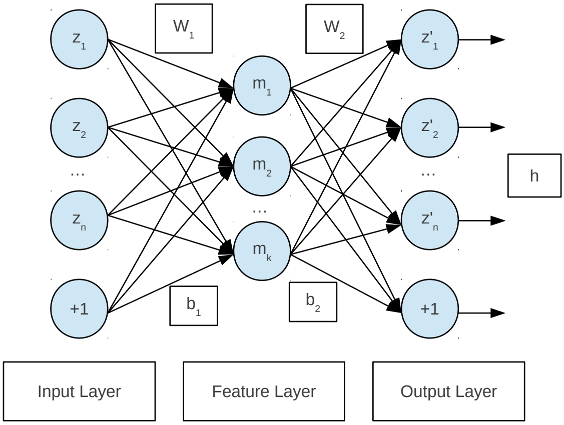


Fig. 1. Example of an autoencoder network, where the function h learns an output z' similar to the input z.

The function used to map the input data $Z_j \in R^n$ is the function

$$h(\theta, j) = g(W_2 g(W_1 Z_j + b_1) + b_2) \tag{1}$$

which uses a set of linear combinations of the model parameters $\rho = (W_1, W_2, b_1, b_2)$ and a non-linear activation function g.

The sizes of the matrices (W_1, W_2) and bias vectors (b_1, b_2) are defined according to the vision task, which in our case is a reconstruction task. Therefore, given a desired number of features k to be obtained, W_1 is of size $k \times n$, W_2 is of size $n \times k$, b_1 is of size k, and b_2 is of size n. The optimum number of features k can be found experimentally, e.g. via cross-validation, to minimize the reconstruction error.

For the activation function g there exist several options such as the sigmoid function

$$g(z) = \frac{1}{1 + \mathrm{e}^{-z}} \tag{2}$$

or the hyperbolic tangent

$$g(z) = \tanh z = \frac{\mathrm{e}^z - \mathrm{e}^{-z}}{\mathrm{e}^z + \mathrm{e}^{-z}}. \tag{3}$$

In the current implementation we chose to use the sigmoid function, as its range is $[0, 1]$ which is suitable for image data, while the tanh function has a range of $[-1, 1]$.

Learning the model parameters $\rho = (W_1, W_2, b_1, b_2)$ can be done using a batch gradient descent method, where the cost function J can be defined to minimize the reconstruction error and will be trained on the data itself:

$$J(\theta, j) = \|h(\theta, j) - Z_j\|_2^2 + \alpha \sum_{i=1}^{2} \|W_i\|_2^2 \tag{4}$$

where the second term of the function is a regularization term that tends to decrease the magnitude of the weights—which helps to prevent overfitting—and the parameter α controls the relative importance of the two terms.

Furthermore, by placing different constraints on the cost function, such as limiting the number of hidden units, we can obtain a compressed representation of the input data. A sparsity constraint will force most of the hidden units to be inactive most of the time. By inactive, we mean the neuronal activation value is close to 0, while active means their value is close to 1. The new cost function is defined as

$$J_{\text{sparse}}(\theta) = J(\theta) + \beta\, \mathrm{KL}(\rho, \rho') \tag{5}$$

where ρ is the mean activation of the hidden units, and ρ' is a sparsity parameter which is usually a small value close to zero. By minimizing the KL divergence, we thus force ρ to be close to zero. In other words, we force most of the hidden units to be inactive.

3.2 Discretization mechanism for obtaining parts

After learning the model parameters, we want to obtain a discrete set of parts. This goal is achieved by clustering in the space of feature vectors formed of hidden unit activations. For each input patch Z_j, we obtain a feature vector

$$a_j = W_1 Z_j + b_1,\ j = 1, \ldots, |Z_*|. \tag{6}$$

The output of the clustering method is a codebook

$$C^1 = \{C_1, \ldots, C_i, \ldots, C_{M^1}\} \tag{7}$$

which represents the set of parts at the first layer Γ^1.

The number of clusters $M^l, l = 1, \ldots, L_{2D}$ (where the number of layers for the 2D representation is L_{2D}) and the clustering method (k-means, agglomerative clustering, hierarchical clustering) represent two more design choices. These can be made based on experiments so as to find the optimum number of parts at each layer l.

The input to the second layer is a set of patches $Z_j^2 = \{a_1, \ldots, a_i, \ldots, a_{p^2}\}$, obtained by concatenating p^2 overlapping feature vectors a_i introduced in (6), extracted at different positions inside the specified receptive field at the second layer of twice the size, $F^2 = 2F^1$. The number p^l of overlapping sub-parts considered at each layer represents another design decision.

3.3 Parameter optimization based on reconstruction error

We chose to optimize the parameter values based on minimizing the reconstruction error at each layer l. For this purpose, we considered the relative root mean square reconstruction error [14], which for one input image/feature patch Z_j^l on layer l is

$$r_j^l = \frac{\left\| h^l(\theta, j) - Z_j^l \right\|_2}{\left\| Z_j^l \right\|_2}, \tag{8}$$

and the overall reconstruction error for a layer l is

$$r^l = \frac{1}{|Z_*^l|} \sum_{j=1}^{|Z_*^l|} r_j^l, \tag{9}$$

where $|Z_*^l|$ is the number of patches on layer l.

The same learning method as described above is used to obtain the model parameters $\rho^2 = (W_1^2, W_2^2, b_1^2, b_2^2)$ at the second layer, while the set of parts Γ^2 is obtained by clustering the feature vectors formed of activations at the second layer:

$$a_j^2 = W_1^2 Z_j^2 + b_1^2, \; j = 1, \ldots, |Z_*^2| \tag{10}$$

where $|Z_*^2|$ is the number of extracted patches on the second layer.

Next, all the other layers in our hierarchical representation, up to the top layer L_{2D}, are learned in an iterative manner as described above, where the input to each layer l is given by the activations of the hidden units learned on patches from the previous layer $l-1$. This hierarchical approach is useful for capturing non-linear dependencies in the input data and models them iteratively, from low-level image patches up to the object level.

For optimizing the construction of the hierarchical representation, we decided to use supervised information on the last two layers. This design choice leads to the formation of view-specific parts at layer $L_{2D}-1$ and category-specific parts at the last layer L_{2D}. The learning process is the same as described above, with the only difference that image patches are selected only from images belonging to one object category, instead of coming from the entire database.

Each layer will produce a set of weight vectors W_1^l which compress the input information, while producing a useful representation.

The set of model parameters at each layer l is given by

$$\theta^l = (rho^l, \Gamma^l, F^l), \rho^l = (W_1^l, W_2^l, b_1^l, b_2^l). \tag{11}$$

The set of parts obtained at each layer $l \in \{1, \ldots, n_{2D}\}$ for the 2D $\{\mathit{\Gamma}^l\}$ and view-based 3D $\{\mathit{\Omega}^l\}$ hierarchical representations are depicted in Section 5.

An approach similar to that described above was proposed by Masci et al. [15], where stacked autoencoders are trained in a similar fashion, one layer at a time, and each layer receives as input the latent representation of the layer below. The main difference from that work, addressed in our approach, is the cross-hierarchy prediction power which is achieved by discretizing the responses of each layer into a set of parts, using clustering techniques. This step enables efficient inference of parts from one hierarchy, given observed parts in the other hierarchy. In the next section we present our model for binding the two hierarchical representations for 2D and view-based 3D data.

4 Establishing correspondences between 2D and 3D view-based hierarchical representations

We build the 2D and the view-based 3D hierarchical representations on similar learning principles, and their parts encode at each layer the same spatial locality $F^l_{2D} = F^l_{3D}$, while the difference lies in the different types of information they encode, edges and depth. Furthermore, the two hierarchies are trained on the same object models, consisting of both 2D edge images and the corresponding depth images or point clouds. The hierarchical compositional 2D model is formed of L_{2D} layers, each layer being composed of a number of 2D parts. As our main goal was to be able to do efficient inference across the two representations, we chose to use the same number of layers in both hierarchies, i.e., $L_{2D} = L_{3D} = L$.

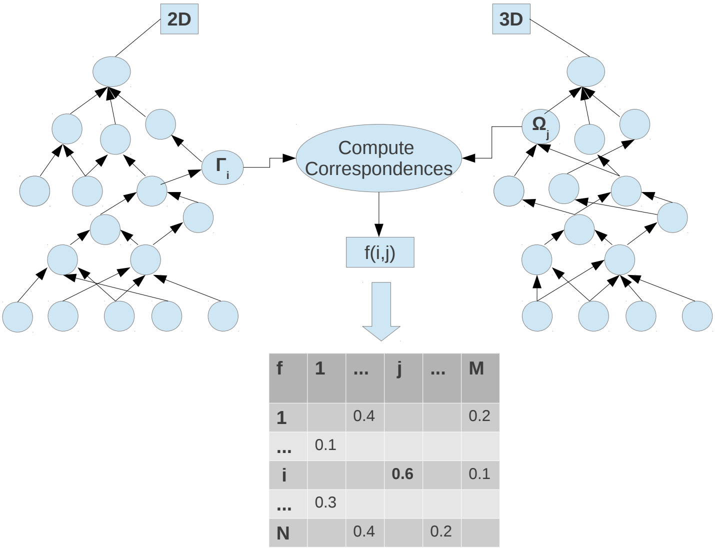
We denote part i constructed at layer l of a hierarchical vocabulary of 2D shapes by $\mathit{\Gamma}^l_i$, for $i = 1, \ldots, |\mathit{\Gamma}^l_*|$ and $l = 1, \ldots, L$. Part j constructed at layer l of the hierarchical vocabulary of view-based 3D shapes is denoted by $\mathit{\Omega}^l_j$, for $j = 1, \ldots, |\mathit{\Omega}^l_*|$ and $l = 1, \ldots, L$.

For establishing correspondences between the parts in the two hierarchies, we build an $N \times M$ *correspondence table* with $N = \sum_{l=1}^{L} |\mathit{\Gamma}^l_*|$ rows corresponding to the 2D parts and $M = \sum_{l=1}^{L} |\mathit{\Omega}^l_*|$ columns corresponding to the view-based 3D parts, as depicted in Figure 2 below. Our goal is to establish layer-by-layer correspondences $(\mathit{\Gamma}^l_i, \mathit{\Omega}^l_j)$ between 2D and view-based 3D parts (Figure 2).

The correspondences between the two types of parts are built using an observation function $c^l(i,j)$, which quantizes the co-occurrences of a 2D part $\mathit{\Gamma}^l_i$ and a view-based 3D part $\mathit{\Omega}^l_j$, and vice-versa, at each layer l. To every correspondence $c^l(i,j)$ we assign a real value. As to each 3D (2D) part might correspond several 2D (3D) parts, all associated correspondences will be activated. Therefore, the value $c^l(i,j)$ is computed as the joint probability of observing parts $\mathit{\Gamma}^l_i$ and $\mathit{\Omega}^l_j$ at the same spatial location:

$$c^l(i,j) = P(\mathit{\Gamma}^l_i, \mathit{\Omega}^l_j). \tag{12}$$

In general, there will not be a one-to-one relation between 2D and view-based 3D parts, as the 2D parts are represented using edges and they can be correlated



f	1	...	j	...	M
1		0.4			0.2
...	0.1				
i			**0.6**		0.1
...	0.3				
N		0.4		0.2	

Fig. 2. Correspondences between the 2D and view-based 3D hierarchies.

with depth parts which encode depth disparities on one side or the other one of the edge part and also can be correlated with internal depth edges. Still, the number of correlations will decrease for higher-level parts, as more complex edge structures such as L, Y or T junctions are better at predicting the possible associated depth structures.

The correspondence table represents the set of possible correspondences over all sets of 2D and view-based 3D parts. Moreover, the joint probability table of 2D and view-based 3D parts can be used to perform inference of view-based 3D parts/2D parts coherently and transparently across the 2D and view-based 3D hierarchies.

Still, the inference process can be regarded from two perspectives, given the type of parts we want to predict, either 2D or 3D. Therefore, for he task of inferring a 2D part Γ_i^l from an observed view-based 3D part Ω_j^l at layer l, we employ the conditional probabilities

$$P(\Gamma_i^l|\Omega_j^l),\ i = 1, \dots, |\Gamma_*^l| \tag{13}$$

to retrieve all 2D parts Γ_i^l which are likely to be observed with the view-based 3D part Ω_j^l. This is related to Eqn. 12 via Bayes' Rule by

$$P(\Gamma_i^l|\Omega_j^l) = \frac{P(\Gamma_i^l, \Omega_j^l)}{P(\Omega_j^l)}. \tag{14}$$

Conversely, inference of a view-based 3D part Ω_j^l can be performed using the conditional probability

$$P(\Omega_j^l|\Gamma_i^l),\ j=1,\dots,|\Omega_*^l|, \tag{15}$$

which permits retrieving all view-based 3D parts Ω_j^l which could be observed with the 2D part Γ_i^l. Again, this is related to Eqn. 12 via

$$P(\Omega_j^l|\Gamma_i^l)=\frac{P(\Gamma_i^l,\Omega_j^l)}{P(\Gamma_i^l)}. \tag{16}$$

The correspondence table introduced above can be considered a starting point for different tasks, such as inferring view-based 3D parts from observed 2D parts, and (conversely) inferring 2D parts from observed view-based 3D parts, object recognition, object reconstruction, or pose estimation.

Furthermore, the correspondence table can be used as a starting point for building better prediction models, such as training a classifier for predicting 3D view-based parts, given an observed 2D part, which takes into account not only the conditional probabilities across hierarchies, but also their associated feature vectors.

4.1 Inference of View-based 3D Parts using 2D and 3D Hierarchical Compositional Models

We now propose a probabilistic model for performing inference in the view-based 3D hierarchy, by reusing the inference mechanism existing in the 2D hierarchy in combination with the joint probability table introduced above. We use the assumption that parts in the view-based 3D hierarchy are formed by propagating evidence from lower to higher layers in a probabilistic manner.

Inference across hierarchies is performed by conditioning the probability of a view-based 3D part Ω_j^l at layer l not only on the sub-parts at the previous layer $l-1$ but also on the 2D parts Γ_k^l at layer l, that are probable to co-occur with it. Thus, we use in the inference process the conditional probability

$$P(\Omega_j^l|\Omega_i^{l-1}) \tag{17}$$

instead of

$$P(\Omega_j^l|\Omega_i^{l-1},\Gamma_k^l) \tag{18}$$

for $i=1,\dots,\Omega_*^{l-1}$.

2D parts Γ_k^l are retrieved using the joint probability table introduced in Section 4, and consist of the set of 2D parts which maximize the $P(\Gamma_k^l,\Omega_j^l)$. The threshold for part selection τ, where $P(\Gamma_k^l,\Omega_j^l)>\tau$, is learned according to the purpose of the inference process.

Furthermore, using Bayes' Rule we can estimate the conditional probability introduced in Eqn. 18 as

$$P(\Omega_j^l|\Omega_i^{l-1},\Gamma_k^l)=\frac{P(\Omega_i^{l-1},\Gamma_k^l|\Omega_j^l)P(\Omega_j^l)}{P(\Omega_i^{l-1},\Gamma_k^l)}. \tag{19}$$

To estimate the conditional probability

$$P(\Omega_i^{l-1}, \Gamma_k^l|\Omega_j^l) = P(\Gamma_k^l|\Omega_j^l, \Omega_i^{l-1})P(\Omega_i^{l-1}|\Omega_j^l) \tag{20}$$

we take into account the principle of forming correlations introduced in Section 4, which suggests that if the parts (Γ_k^l, Ω_j^l) can be correlated, then the parts $(\Gamma_k^l, \Omega_i^{l-1})$ are conditionally independent given part Ω_j^l, as the correspondences between parts are established at the same layers in the two hierarchies. Therefore,

$$P(\Omega_i^{l-1}, \Gamma_k^l|\Omega_j^l) = P(\Gamma_k^l|\Omega_j^l)P(\Omega_i^{l-1}|\Omega_j^l). \tag{21}$$

By substituting Eqn. 21 into Eqn. 19 we obtain

$$P(\Omega_j^l|\Omega_i^{l-1}, \Gamma_k^l) = \frac{P(\Gamma_k^l|\Omega_j^l)P(\Omega_i^{l-1}|\Omega_j^l)P(\Omega_j^l)}{P(\Omega_i^{l-1}, \Gamma_k^l)}, \tag{22}$$

where the conditional probability $P(\Gamma_k^l|\Omega_j^l)$ can be obtained from Eqn. 14, and $P(\Omega_i^{l-1}|\Omega_j^l)$ can be computed using the initial conditional probability introduced in Eqn. 17 as

$$P(\Omega_i^{l-1}|\Omega_j^l) = \frac{P(\Omega_j^l|\Omega_i^{l-1})P(\Omega_i^{l-1})}{P(\Omega_j^l)}. \tag{23}$$

Finally, by introducing Eqn. 23 into Eqn. 22 we obtain

$$P(\Omega_j^l|\Omega_i^{l-1}, \Gamma_k^l) = \frac{P(\Gamma_k^l|\Omega_j^l)P(\Omega_j^l|\Omega_i^{l-1})P(\Omega_i^{l-1})}{P(\Omega_i^{l-1}, \Gamma_k^l)}. \tag{24}$$

The view-based 3D part prior $P(\Omega_i^{l-1})$ can be considered uniform or computed using the inverse frequency operator on an input database, which takes into account the part frequency across the database, while the joint probability $P(\Omega_i^{l-1}, \Gamma_k^l)$ can be obtained from Eqn. 16 presented in Section 4.

The proposed model will be initialized with probabilities of 2D parts at the first layer of the 2D hierarchy. The observations in the 2D hierarchy will be used to propagate information to the view-based 3D hierarchy and finally to obtain hypotheses about view-based 3D parts.

Next, we introduce a method for optimizing the prediction of view-based 3D parts, by incorporating additional context information.

4.2 Classification

As there are many-to-many correspondences, choosing the best part according to the conditional probability is not always the best choice, due to ambiguity of edge structures which could correspond to many depth parts. Therefore, one solution for effective inference of view-based 3D parts, given observed 2D parts, consists of extracting complementary information from the region surrounding an edge part using HOG features. Additionally, we trained a SVM classifier to improve the prediction of view-based 3D parts. The comparison of results obtained for each layer are presented in Section 5, showing the detection accuracy of depth parts in both cases, without and with additional context information. The baseline model is the probabilistic inference model presented in Section 4.1.

We depict in Figure 3 two examples of reconstructed depth images using the inference algorithm introduced in subsection 4.1 based on only 2D edge information in Figure 3(A) and on additional complementary features in Figure 3(B). As it can be noticed adding context information helps in most cases at disambiguating the depth part corresponding to an edge part.

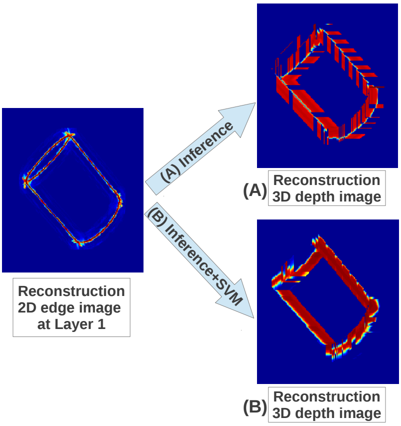

(A) Inference
(B) Inference+SVM
Reconstruction 2D edge image at Layer 1
(A) Reconstruction 3D depth image
(B) Reconstruction 3D depth image

Fig. 3. Reconstruction of depth image based on 2D edge parts using only the inference algorithm in (A) and combining the inference algorithm with the SVM classifier in (B).

In the following section we provide the results obtained using the proposed approach for learning the two hierarchical representations and the inference algorithm for prediction across hierarchies.

5 Experimental Results

We learned and tested the proposed hierarchical representations for 2D and view-based 3D data on a set of object categories from the PaCMan dataset which is publicly available[2]. We depict in Figure 4 a set of example images (edge and depth) from three object categories.

The dataset contains gray-scale and depth images generated from 3D models of 20 categories of various kitchen related objects, each category containing on

[2] http://www.pacman-project.eu/datasets/

average 20 instances. Each object is captured at dense, regular viewpoint intervals, and we use all the 48 different viewpoints of an object instance, both in the training and testing phase. We trained and tested our approach on three object categories, using three instances from each category in the training phase and two new, but similar instances in the testing phase. The testing set $\mathcal{Z}'$ of edge images $Z'^i \in \mathcal{Z}', i = 1, \ldots, |\mathcal{Z}'|$ is composed of vectorized image patches Z'_j, i.e., $Z'^i = \{Z'_1, \ldots, Z'_j, \ldots, Z'_{|Z'^i|}\}$ in the same manner as presented in Section 3.

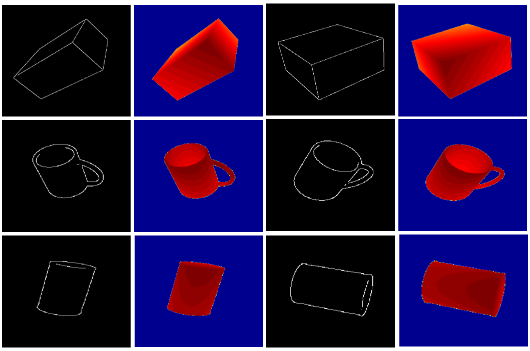


Fig. 4. Examples of object instances (box, mug, can) from the PaCMan database.

The number of constructed layers in a hierarchy depends on the size of the training objects. For the PaCMan dataset, the hierarchical representations are formed of five layers.

One important issue consisted of finding the optimum number of parts at each layer for each representation. Apart from minimizing the reconstruction error introduced in Eqn. 8, we also used the Davies-Bouldin (DB) index [16] as a method to assess the quality of the clustering process. The optimum number of parts k on a layer l is selected according to the minimum DB index. As the dimensionality of the feature vectors used in the clustering process increases at each layer, we applied the PCA dimensionality reduction method for a faster and more efficient selection of the optimum number of clusters.

Examples of the learned parts at each layer of the two hierarchical representations and correspondences between them are depicted in Figure 5.

We present in Table 1 for each layer in the 2D and view-based 3D hierarchies the optimum number of parts and the reconstruction error obtained for each hierarchy on the testing set. The reconstruction error is computed using an

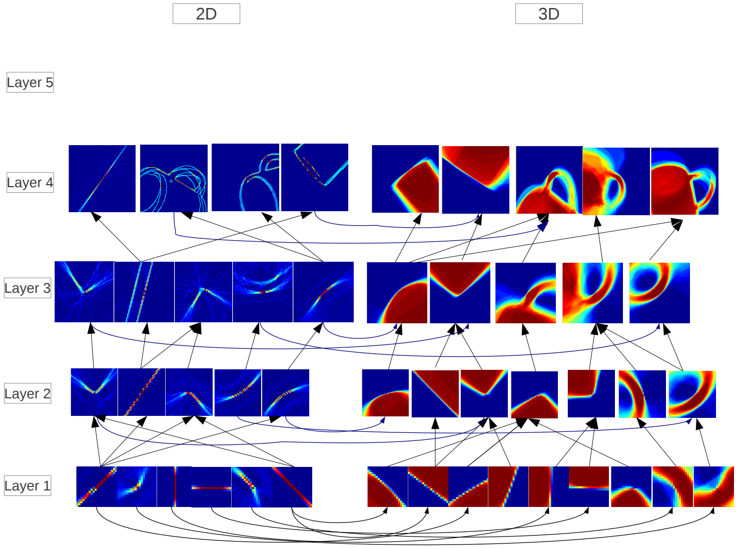


Fig. 5. Examples of parts at each layer in the 2D and view-based 3D hierarchies and correspondences between them.

adaptation of the formula introduced in Eqn. 8,

$$Err^l = \frac{1}{|Z'^l_*|} \sum_{j=1}^{|Z'^l_*|} r^l_j, \quad r^l_j = \frac{\left\| \Gamma^l_m - Z'^l_j \right\|_2}{\left\| Z'^l_j \right\|_2}, \tag{25}$$

where m is the index of best-fitting part for a patch from the testing set Z'^l_j at layer l that indicates the size of the receptive field F^l:

$$m = \min_{i=1}^{|\Gamma^l_*|} (\left\| \Gamma^l_i - Z'^l_j \right\|_2). \tag{26}$$

Layer l	$\lvert\Gamma^l_*\rvert$	Err_{2D}	$\lvert\Omega^l_*\rvert$	Err_{3D}
1	130	0.32	100	0.53
2	400	0.49	400	0.42
3	400	0.63	500	0.37
4	550	0.71	600	0.34
5	600	0.75	650	0.31

Table 1. Statistics of parts at Layers 1-5 in the 2D and view-based 3D hierarchies.

Analysis of the results presented in Table 1 highlights two phenomena: The reconstruction error in the 2D case increases with l, while in the 3D case it decreases. As the error is computed per pixel in a patch, a slight translation of an edge part will increase the error with quite a high margin. On the other hand in the depth case, a part translation impacts the error in a much smaller way, while the decrease of the overall error is due to the increasing of the number of parts with l, which improves the chances of finding a better fitting part for each patch.

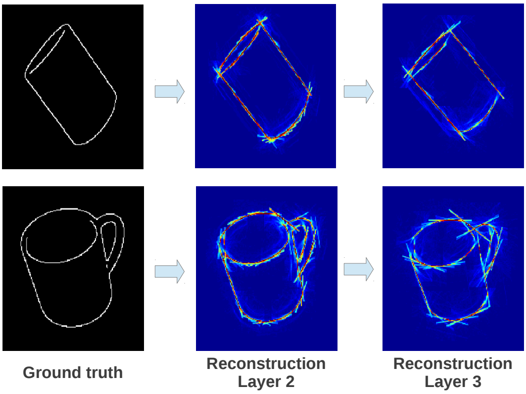


Fig. 6. 2D reconstruction examples of a mug instance from different viewpoints at Layers 2 and 3.

We present in Figures 6 and 7 several examples of reconstruction for both hierarchical representations at different layers.

One of our principal aims consists in quantifying the predictive power of the learned correspondences between 2D and view-based 3D information channels, at each layer of their compositional representations. This is assessed using multiple statistical measures. First, we assess the conditional probabilities of 2D parts given 3D parts and vice versa, and compute the predictive power in the case of choosing the part which maximizes the conditional probability. Some examples of correlations are depicted in Figure 8.

The results obtained for each layer are presented in Table 2 together with the reconstruction error of view-based 3D parts given 2D parts. The *Inference* column contains for each layer l the recall value for the correct prediction of depth parts relative to the ground truth depth patches using the inference algorithm, while the classifier column $Inference{+}SVM$ shows the recall when using additional context information in combination with the inference algorithm.

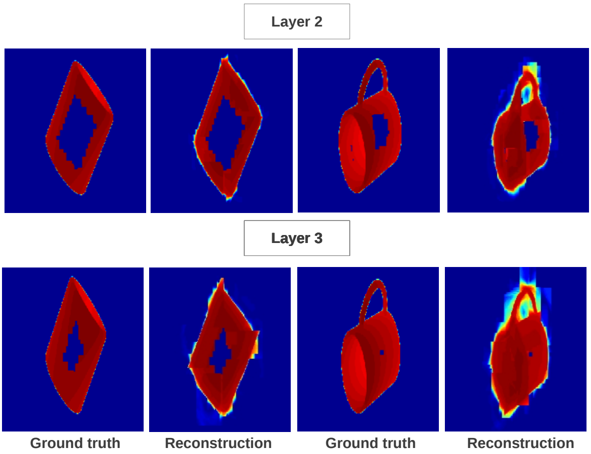


Fig. 7. 3D view-based reconstruction examples of a mug instance from different viewpoints at Layers 2 and 3.

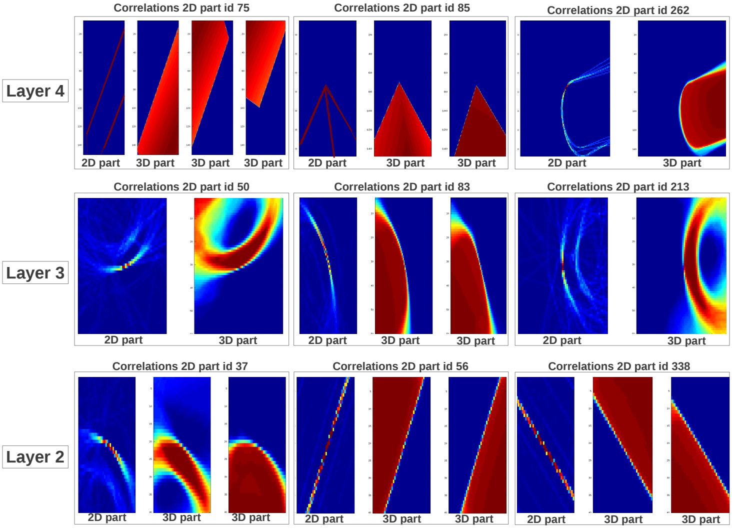


Fig. 8. Correlation examples between 2D and the best corresponding view-based 3D parts at Layers 2 and 3.

The cross-hierarchy reconstruction error is computed using an adapted version of Eqn. 8,

$$Err(3D|2D)^l = \frac{1}{|D'^l_*|}\sum_{j=1}^{|D'^l_*|} r^l_j(3D|2D), \quad r^l_j(3D|2D) = \frac{\left\|\varOmega^l_{m'} - D'^l_j\right\|_2}{\left\|D'^l_j\right\|_2}, \tag{27}$$

where m' is the index that maximizes the conditional probability of a view-based 3D part given the observed 2D part $\varGamma^l_m$, and D'^l_j is the ground-truth depth patch from the testing set corresponding to the observed 2D patch Z'^l_j at layer l, i.e.,

$$m' = \max_{i=1}^{|\varOmega^l_*|} P(\varOmega^l_i|\varGamma^l_m) \tag{28}$$

and m is the index of best fitting 2D part for the patch Z'^l_j,

$$m = \min_{i=1}^{|\varGamma^l_*|}(\left\|\varGamma^l_i - Z'^l_j\right\|_2). \tag{29}$$

Layer l	$Inference$		$Inference + SVM$	
	Recall	$Err(3D\|2D)$	Recall	$Err(3D\|2D)$
l_1	34.12%	1.39	50.4%	0.64
l_2	27.36%	0.99	53.6%	0.56
l_3	24.27%	0.87	56.4%	0.44
l_4	22.44%	0.69	45%	0.37
l_5	35.50%	0.52	40%	0.48

Table 2. Statistics of inference results of view-based 3D parts given 2D observed parts at Layers 1-5 in the 2D and view-based 3D hierarchies.

In Figure 9 we depict several examples of depth images formed of view-based parts inferred from observed 2D parts detected from a corresponding edge image using the probabilistic model introduced in Section 4.1. As we use overlapping receptive fields in both hierarchies, we enforce consistency of overlapping predictions using a correlation score for each pixel, weighted by the confidence of the 2D parts that predicted its value and the conditional probability of the view-based 3D part given the observed 2D part.

The set of images at each layer $l = 1, \ldots, 5$ in Figure 9 is composed of the ground truth edge image, the predicted edge image via the 2D hierarchy, the depth reconstructed image based on the inference of depth parts given 2D detected parts and the corresponding ground truth depth image, at each specific layer. The images only contain data around edge parts, while the size of the predicted region increases at each layer.

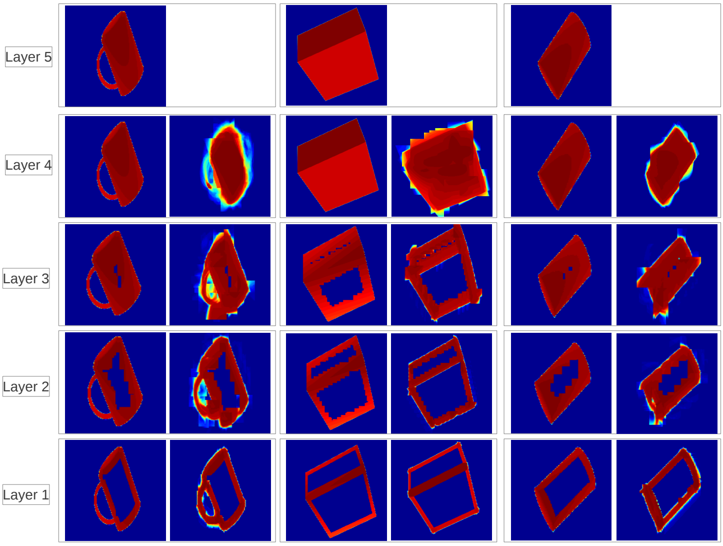


Fig. 9. Reconstruction examples of depth images based on inference results of view-based 3D parts given 2D observed parts at Layers 1–5 in the 2D and view-based 3D hierarchies.

6 Conclusion

In this work we proposed a method for constructing two hierarchical representations, for two information channels, 2D intensity data and view-based 3D depth data. We employed as a learning algorithm a sparse autonencoder network, where each layer receives as input the latent representation of the layer below. For obtaining predictions across hierarchies, we discretized the responses of each layer into a set of parts, using clustering techniques. Furthermore, we introduced a probabilistic fusion model for linking the two information channels across hierarchies. The fused model enables an improved and refined representation of the world, while being able to infer 3D object shape from 2D intensity edge information.

Next, we evaluated the statistical relationship between representations obtained from 2D and view-based 3D models, analysis which facilitates also the understanding of the geometric properties of the two compositional representations. The evaluation of the learned correspondences between parts in the two hierarchies showed that meaningful correlations can be formed for a set of parts, while specific 2D parts have low prediction power, due to ambiguity of edge structures which could correspond to many depth parts. We handled this problem, by extracting complementary information from the region surrounding an edge part using HOG features, which were fed to a SVM classifier. This method proved its usefulness in the inference process, leading to an improved prediction of the correct corresponding depth part.

As future work, we plan to test the efficacy of our approach on scenes composed of several objects exposing different levels of complexity, such as clutter and occlusions.

References

1. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. Volume 32. (2010) 1627–1645
2. Teney, D., Piater, J.: Multiview feature distributions for object detection and continuous pose estimation. In: Computer Vision and Image Understanding Journal. Volume 125. (2014) 265–282
3. Liebelt, J., Schmid, C.: Multi-view object class detection with a 3d geometric model. In: IEEE Conference on Computer Vision & Pattern Recognition. (jun 2010) 1688–1695
4. Arca, S., Lanzarotti, R., Lipori, G.: Face recognition based on 2D and 3D features. In: International Conference on Knowledge-Based Intelligent Information and Engineering Systems. (2007) 455–462
5. Soltana, W.B., Huang, D., Ardabilian, M., Chen, L.: Comparison of 2D/3D features and their adaptive score level fusion for 3D face recognition. In: 3D data processing, visualization and transmission (3DPVT). (2010) 455–462
6. Mahmoudi, F., Samadzadegan, F., Reinartz, P.: A decision level fusion method for object recognition using multi-angular imagery. In: International Conference on Sensors and Models in Photogrammetry and Remote Sensing. (2013) 409–414
7. Wang, Y., Gong, M., Wang, T., Cohen-Or, D., Zhang, H., Chen, B.: Projective Analysis for 3D Shape Segmentation. ACM Transactions on Graphics **32**(6) (2013)
8. Fleming, R.W., Singh, M.: Visual perception of 3d shape. In: ACM SIGGRAPH. (2009) 1–9
9. Cyr, C.M., Kimia, B.B.: 3D object recognition using shape similarity-based aspect graph. In: ICCV. (2001) 254–261
10. Cheng, C.C., Li, C.T., Chen, L.G.: A novel 2D-to-3D conversion system using edge information. In: IEEE Transactions on consumer electronics. Volume 56. (2010) 1739–1745
11. Eitz, M., Richter, R., Boubekeur, T., Hildebrand, K., Alexa, M.: Sketch-based shape retrieval. Volume 31. (2012) 31:1–31:10
12. Dai, J., Hong, Y., Hu, W., Zhu, S.C., Nian Wu, Y.: Unsupervised learning of dictionaries of hierarchical compositional models. (June 2014)
13. Wu, Y.N., Si, Z., Gong, H., Zhu, S.C.: Learning active basis model for object detection and recognition. In: International Journal of Computer Vision. Volume 90. (2010) 198–235
14. Chai, T., Draxler, R.R.: Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature. In: Geoscientific Model Development. Volume 7. (2014) 1247–1250
15. Masci, J., Meier, U., Ciresan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: 21th International Conference on Artificial Neural Networks (ICAN'11). Volume 1. (2011) 52–59
16. Davies, D.L., Bouldin, D.W.: A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence **1**(2) (1997) 224–227