Probabilistic Scene Modeling for Situated Computer Vision

Sven Wachsmuth, Agnes Swadzba

Applied Computer Science, Faculty of Technology, Bielefeld University
33615, Bielefeld, Universitätsstraße 25, Germany
{swachsmu, aswadzba}@techfak.uni-bielefeld.de

Abstract. Verbal statements and vision are a rich source of information in a human-machine interaction scenario. For this reason Situated Computer Vision aims to include knowledge about the communicative situation in which it takes place. This paper presents three approaches how to achieve scene models of such scenarios combining different modalities. Seeing (planar) scenes as configurations of parts leads to a probabilistic modeling with Bayes' nets relating spoken utterances with results of an object recognition step. In the second approach parallel datasets form the basis for analyzing the statistical dependencies between them through learning a statistical translation model which maps between these datasets (here: words in a text and boundary fragments extracted in 2D images). The third approach deals with complex indoor scenes from which 3D data is acquired. Planar structures in the 3D points and statistics extracted on these planar patches describe the coarse spatial layouts of different indoor room types in such a way that a holistic classification scheme can be provided.

Keywords. Scene Modeling, Human Robot Interaction

1 Introduction

We are currently witnessing that computer vision is becoming a more and more important cue in human-machine interaction. Verbal statements in this interaction relate to the external scene, gestures provide means of non-verbal communication, actions indicate human intentions, and gaze provides hints on human attention. From the standpoint of communication, vision is a rich source of contextual information. But we can also turn the perspective around. Vision takes place in a communicative situation, that provides expectations for visual processing and dictates which aspects are relevant in a scene. *Situated computer vision* aims at taking this information into account.

Situatedness refers to an inherent ambiguity occurring in a selective perception process. The perceiver needs to be aware of the current situation in order to infer an intended result because interpretations are uncertain and different interpretations might be possible. Such situational constraints can be formulated as contextual knowledge or as a prior of a probability distribution. Perception

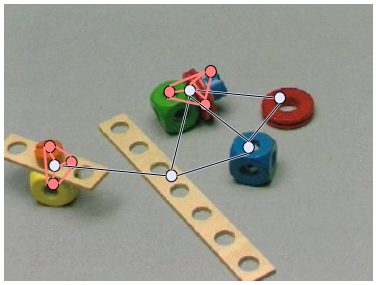
Dagstuhl Seminar Proceedings 08091
Logic and Probability for Scene Interpretation
http://drops.dagstuhl.de/opus/volltexte/2008/1609

is modeled as a selective process, i.e. it does not aim at a complete nor generic object or scene understanding. Instead, perception is embedded in a purposive capturing process or a related task. It could also be constraint by the system's own embodiment as suggested by recent work in cognitive systems [1]. Thus there is a large spectrum of possible situational constraints ranging from physical embodiment to mental models. The latter are the focus of this article. They have been introduced as situation models already a long time ago [2,3]. Situation models have been first described as amodal representations, i.e. independent of any perceptual process. However, newer experimental studies show that they are tightly linked to visual perception [4].

In the following sections, computer vision will mostly be treated as a stochastic process. This pays tribute to the inherent uncertainty of relating situational expectations to vision results. Modeling of context plays an important role that should lead to more stable scene interpretations. The article focuses on three different approaches for situated computer vision. The first example represents scenes as a configuration of objects and relates verbal descriptions to them (Section 2). The second example explores cross-situational inferences and exploits re-occurring patterns for model acquisition (Section 3). The third example is directed to complex 3D scenes and demonstrates a holistic scene classification (Section 4).

2 Scenes as configurations of parts

Because of the central role of objects and relations in the categorization of scenes, many approaches choose this level for recognition purposes [5,6]. The strength as well as the weakness of this approach is that it is built on an object recognition step. On the positive side, it enables a generic and compositional approach for modeling contexts that directly relates to language. On the negative side, it is an error-prone strategy that especially suffers from segmentation difficulties.



Verbal scene specifications:

- "... bar with a bolt ..."
- "... motor in front of the bar ..."
- "... blue cube in front of the ring ..."
- "... long thing in the middle ..."
- etc.

Fig. 1. Graph-based scene representations: a graph can be defined on different granularity and different relational semantics.

In order to deal with recognition errors on the level of primitives, Wachsmuth and Sagerer [7] combine spatial relationships with an uncertainty model of the

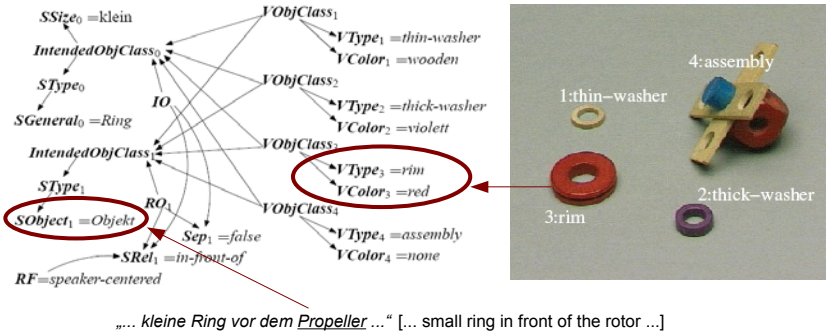


Fig. 2. Example of a German dataset: The Bayesian network is dynamically generated from the visual and verbal information given. Note that the semantics of the word “Propeller” is previously not known. Therefore, a general term *object* is instantiated in the corresponding observable variable.

object recognition process as well as of the object and relation naming process using graphs. The nodes of the graph are defined by object detection results and are attributed with the object region, object type, and object color information from visual processing. Edges are defined by topological relations (e.g. connected *with*) or projective relations (e.g. *in front of*). Here, a partial scene model is specified by a verbal description. This needs to be matched to the visual representation. For this purpose a Bayesian network is dynamically constructed from the verbal and visual information given.

In Fig. 1 an example scene and possible verbal descriptions are given. The scenario is limited by a fixed number of elementary object types, a fixed number of color classes, and a uniform table background. However, more complex objects can be constructed by aggregation of elementary objects using bolts and nut-parts. These parts can dynamically be assigned names in the course of a dialog that refer to functional parts of the construction goal, e.g. “motor unit” of a toy-airplane. Thus, relations are defined on two different granularity levels: (i) relations between parts connected in an aggregated structure, (ii) relations between spatially separated scene entities. In a specific matching case, the level of granularity is selected by the wording used in the verbal description.

In Fig. 2 an example of the dynamically generated Bayesian network is shown. It is constructed from pre-modeled sub-networks that are combined with regard to the number of visual scene objects and the number of verbally described objects. The hidden variables IO and RO realize the mapping between verbal items and visual items. The mapping is constrained by the spatial relation that further depends on the reference frame selected by the speaker. The model parameters are partially learned from labeled training data[1] and are partially hand-crafted[2].

[1] On the vision side, the statistics of the object recognizer and pixel-based color classifier are measured by respective confusion matrices. The statistics on the class-specific wording is based on an online questionnaire.

[2] A hierarchy of nouns denoting object classes and super-classes was modeled by hand.

Further details of the Bayesian network are discussed in [7]. The spatial model of the system is further described in [8].

The matching is computed by the maximum *a posteriori* hypotheses of the IO and RO variables,

$$(io^*, ro^*) = \underset{io,ro}{\operatorname{argmax}} Pr(IO = io, RO_1 = ro|\mathcal{E}) \tag{1}$$

where $\mathcal{E}$ is the set of observations given by the verbal description and the visual scene representation. Evaluation experiments show that a correct scene identification can be inferred despite recognition errors on the visual as well as on the verbal side. The system achieves a correct object mapping (IO) (with two additionally selected objects allowed) of 76% for high input error rates of 21% lost or wrongly recognized verbal features as well as 15% false type classifications and 9% false color classifications. For objects, that are sufficiently specified by verbal descriptions, it is even possible to correct erroneous recognition results.

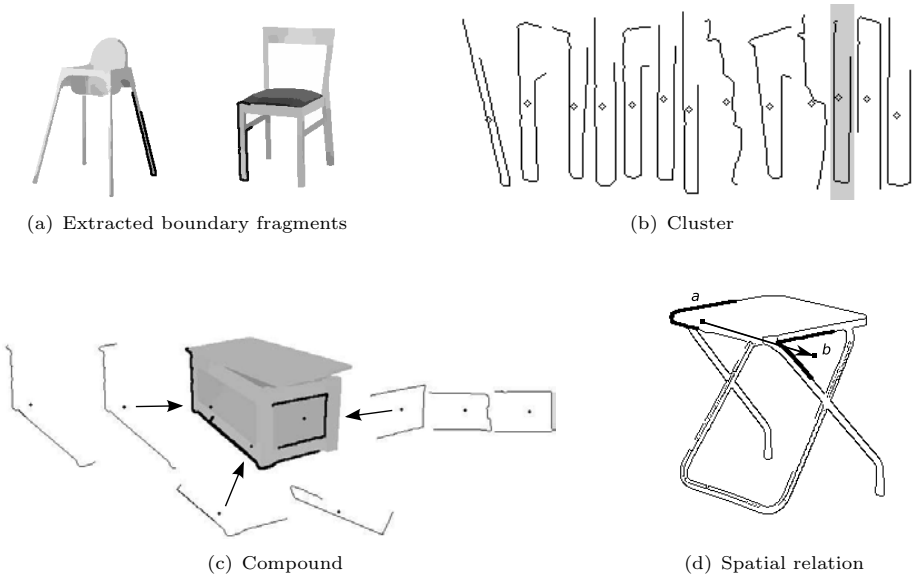
3 Cross-situational learning

Another rich source of information are parallel datasets. They provide a coarse grouping of paired collections from different modalities that is the basis for analyzing the statistical dependencies between them. The coarse groupings are given by *situations*. Here, a situation is defined by a simple pairing of an image and its caption, but it could also be given by an observed scene or action performed and a spoken utterance. It is difficult to learn something from a single isolated situation because correspondences between different modalities are not given explicitly. The system has to infer them despite noise, distracting data, and the inherent combinatorics of n-to-m relations. Related approaches have been put forward by [9,10]. However, these concentrate either on blob-based representations using color and texture features or on local descriptors like SIFT [11].

The model acquisition task is formulated as the learning of a statistical translation model. In statistical language models, we generally seek to find the translation string $\mathbf{e} = (e_1, \ldots, e_L)$ that maximizes the probability $Pr(\mathbf{e}|\mathbf{f})$, given the source string $\mathbf{f} = (f_1, \ldots, f_M)$ (where $\mathbf{f}$ refers to French and $\mathbf{e}$ refers to English in the original work by [12]). Using Bayes' rule and maximizing the numerator, the following equation is obtained:

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}}\, Pr(\mathbf{e}|\mathbf{f}) = \underset{\mathbf{e}}{\operatorname{argmax}}\, Pr(\mathbf{f}|\mathbf{e})Pr(\mathbf{e}). \tag{2}$$

$Pr(\mathbf{e})$ is incorporated into the formula, which is the probability distribution over all valid strings $\mathbf{e}$ provided by the grammar of the model. $Pr(\mathbf{f}|\mathbf{e})$ is known as the *translation model* (prediction of $\mathbf{f}$ from $\mathbf{e}$), and $Pr(\mathbf{e})$ as the *language model* (probabilities over $\mathbf{e}$ independent of $\mathbf{f}$). Most of the work that transfers this concept to image annotation tasks [13,14,15] concentrates on the translation model; taking $\mathbf{f}$ as the words in the text and $\mathbf{e}$ as the visual words in the images, they thus predict words from image items. However, the omission of



(a) Extracted boundary fragments (b) Cluster

(c) Compound (d) Spatial relation

Fig. 3. Building compounds from boundary fragments. First, fragments are clustered using a symmetrically defined edge distance. Secondly, compounds are learned that encode spatial relations between fragment classes.

the language model component, $Pr(\mathbf{e})$ (in this case, probabilities over the "language" of images—i.e., over "good" image representations), can be seen as a shortcoming. The structural information in images is mostly neglected.

Local descriptors exploit the textural characteristics of object surfaces, but they do not capture the overall shape of an object. Moringen et al. [16] focus on an alternative image representation that uses boundary fragments. These can be directly extracted from an image by a connected component analysis on edge pixels [17] or generated from an image abstraction provided by a region segmentation. In the second case, region boundaries define the edge pixels. A boundary fragment $\mathbf{f}$ can simply be defined as a connected sequence of edge pixels $f_k = (x, y)$:

$$\mathbf{f} = (f_1, \ldots, f_K), \text{ where } |f_k - f_{k+1}| \leq \sqrt{2}, \quad 1 \leq k < K. \tag{3}$$

Similar to [17], Moringen et al. extract fragments by chaining from randomly chosen seed points. These provide templates for fault-tolerant shape recognition by using chamfer matching as described by [18]. Chamfer matching utilizes a distance transform in order to implement an efficient way of computing the edge distance d_{edge} between a boundary fragment $\mathbf{f}$ possibly transformed by T and an edge image $\mathbf{I}$,

$$d_{edge}(\mathbf{f}, T, \mathbf{I}) \equiv \frac{1}{|\mathbf{f}|} \sum_{k=1}^{|\mathbf{f}|} \mathbf{I}^d[(Tf)_k]^2 \text{ where } |\mathbf{f}| = |(f_1, \ldots, f_K)| = K. \tag{4}$$

where $\mathbf{I}^d$ is the distance transformed image (pixels are coding the distance to the next edge pixel rather than the present of edges).

In the following, the edge distance serves two different purposes in the translation framework: (i) it provides the basis for a distance metric on boundary fragments that is used for clustering purposes and (ii) it defines a detector for fragment classes on images.

Similar to other related approaches, a basic visual vocabulary is learned by clustering a set of singleton features. An agglomerative clustering is used because the fragments and edge distance do not form a vector space[3]. [16] use a symmetric variant of d_{edge} for clustering:

$$\begin{aligned} d_{symm}(\mathbf{f}_1, \mathbf{f}_2) &= d'_{edge}(\mathbf{f}_1, \mathbf{f}_2) + d'_{edge}(\mathbf{f}_2, \mathbf{f}_1), \\ \text{where } d'_{edge}(\mathbf{f}_1, \mathbf{f}_2) &= \min_{T \in \mathcal{T}} d_{edge}(\mathbf{f}_1, T, \mathbf{I}_{\mathbf{f}_2}) \end{aligned} \tag{5}$$

Here, $\mathcal{T}$ is a discrete set of transformations that is applied to $\mathbf{f}_1$ when overlaying it over an image during chamfer matching, $\mathbf{I}_{\mathbf{f}_2}$ is a bitmap representation of the boundary fragment $\mathbf{f}_2$. Then, a cluster v_l is defined as a triple $(\mathcal{F}_l, \hat{f}_l, \eta_l)$ consisting of a set $\mathcal{F}_l$ of fragments, a representative fragment $\hat{f}_l$ and a detection threshold η_l that is later optimized on the training set by applying the translation model.

In agglomerative clustering, there are different possibilities to transfer the distance function defined on elements to a distance function between clusters. [19] reports a good performance for a maximum operation:

$$d_{max}(v_1, v_2) = \max_{\mathbf{f}_j \in \mathcal{F}_1, \mathbf{f}_k \in \mathcal{F}_2} d_{symm}(\mathbf{f}_j, \mathbf{f}_k), \text{ with } v_l = (\mathcal{F}_l, \hat{f}_l, \eta_l), l = 1, 2. \tag{6}$$

The clusters define fragment classes $\mathcal{V}$ that provide the basic visual vocabulary for the translation model. Fig. 3(b) shows an exemplary cluster that may be associated with the semantic concept of *legs of chairs, tables, or stools*. More specific visual descriptions can be defined by visual compounds:

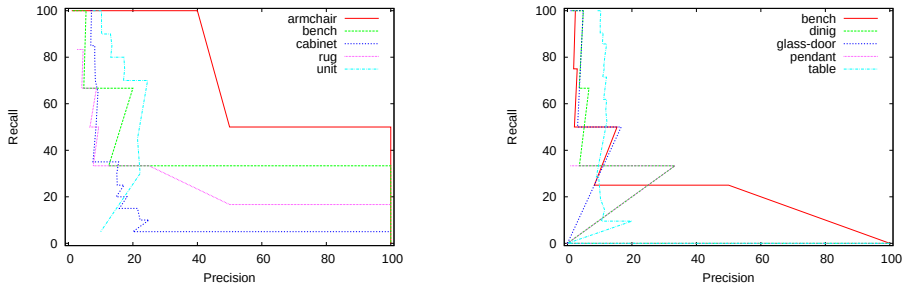
$$\begin{aligned} c_m = (\mathcal{V}_m, \mathcal{R}_m), \text{ where } \mathcal{V}_m &= \{v_{mj} | v_{mj} \in \mathcal{V}, j = 1 \dots J_m\}, \\ \mathcal{R}_m &= \{r_m^{jk} : \mathcal{P} \times \mathcal{P} \to I\!R | j, k = 1 \dots J_m\} \\ &\text{where } \mathcal{P} \text{ is set of pixel positions } \{0, 1, \dots, 255\}^2. \end{aligned} \tag{7}$$

Here, $\mathcal{V}_m$ is a collection of fragment classes with spatial relations $\mathcal{R}_m$ between them. Let $v_{m\,1}$ and $v_{m\,2}$ be fragment classes detected in the image at positions $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$. Then the spatial relation between them can be judged by

$$r_m^{1\,2}(p_1, p_2) = \frac{1}{n_m} \sum_{i=1}^{n_m} \mathcal{N}_{\mu_{mi}^{1\,2}, \sigma}(p_2 - p_1). \tag{8}$$

Here, n_m is the number of occurrences of the compound c_m in the training set. Each offset between the detected fragment classes v_{mj} and v_{mk} is stored in μ_{mi}^{jk} defining a Gaussian kernel with standard deviation σ.

[3] As a consequence, the mean-fragment cannot be computed directly.



(a) Training set (b) Test set

Fig. 4. Boundary fragment compounds: results for a furniture dataset with 300 training images and 225 test images.

During the training of the translation model, Moringen et al. search for compounds by using Melamed's method for finding non-compositional-compounds (NCCs) in parallel text [20]. Fig. 3(c) shows an exemplary compound generated in an experiment on a captionized furniture dataset. The dataset consisted of 525 images (300 training, 225 test) with single pieces of furniture or groups of furniture. The captions have been processed by a tagger [21] and partial parser [22,23] leaving between 1 and 4 head nouns. Precision-recall curves are given in Fig. 4 for some of the vocabulary words learned. Relatively low precision values indicate that there is a large variance of shapes in the dataset. The training set included only a few exemplars per word category so that generalizing models are difficult to learn. However, for some words like 'bench' reasonable compound models have been extracted.

4 Dealing with Complex 3D Scenes

This section is going to contribute to the question, how to capture scene structures and how to extract context for interpreting tasks in complex scenes. The importance of context has already been recognized a long time ago. Systems like CONDOR [24] or SPAM [25] coded explicit contextual rules performing a complex knowledge engineering task. More recently, graphical models have been applied in order to provide a more concise model relating objects and aspects of the considered scene [26,27]. Murphy, Torralba, and Freeman estimate global contexts, like persons, vehicles, furniture and vegetation from low-level image features [26]. Hoiem, Efros, and Herbert first extract a 3D surface geometry from 2D images and relate the estimated local geometries to object classes predicted by a window-based object detector [27].

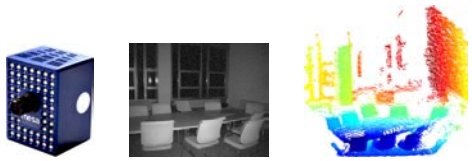
The work discussed so far mainly deals with 2D image information. Murphy et al. demonstrate that many different scene categories can be distinguished by purely considering 2D image statistics on texture and edges. Other approaches also distinguish successfully indoor/outdoor [28], sky/no-sky, vegeta-

tion/no-vegetation [29]. However, this does not necessarily extrapolate to finerly graded scene categories like different types of rooms, e.g. “office” or “meeting room”. Here, typical furniture like tables, chairs, and shelves reoccur, but in different layouts. Furthermore, furniture in the same type of room may have changing colors and textures or be viewed from different directions.

In these cases, a 3D description of the scene is much more invariant with regard to in-class variations. However, strategies that provide a complete semantic interpretation of the 3D scene suffer from very constraint settings and the necessity of extensive modeling. Therefore, we aim at a more holistic 3D approach to scene classification in the spirit of the gist approach used by Torralba [30]. In the following, we describe the scene by a collection of planar structures and analyze whether it is possible to compute proper feature vectors for the classification of different room types (here: office, hall, and meeting room). The challenge faced is to categorize rooms only based on the information of one frame. Details of our approach can be looked up in [31]. Section 4.1 presents our 3D data acquisition and necessary steps for determining sets of planar structures in this data. In Section 4.2 and 4.3 features and classifiers are chosen and examined with regard to their performance in categorizing room percepts to room types.

4.1 3D Data Acquisition and Meaningful Structure Extraction

3D Data Acquisition. Besides the known techniques like laser scanners and stereo rigs for acquiring 3D information recently new hardware was developed by Swiss Center for Electronics and Microtechnology (CSEM) [32]. This camera, Swissranger SR3000 (Fig. 5(a)), provide 3D data in real-time independent of texture and lighting conditions. 176×144 CMOS active pixel sensors measure distances (Fig. 5(c)) between the optical center of the camera and the real 3D world points via the time-of-flight of a near-infrared signal. Additionally each sensor delivers an amplitude value indicating the amount of light reflected by a world point.



(a) (b) (c)

Fig. 5. (a) Swissranger SR3000, (b) example amplitude image, and (c) example 3D point cloud preprocessed.

To deal with noise arising from the different reflection properties, several preprocessing techniques proposed in [33] are applied. A distance-adaptive median filter smooths the distance values with mask sizes depending on this value. Points with small amplitude values, which indicate low quality of the measurement, are removed and edge points arising in the case when light from the foreand the background hits the same pixel simultaneously are rejected. Finally, the distances are backprojected to compute 3D coordinates with regard to a camera coordinate system. The computed 3D points are arranged regularly in a 2D lattice enabling us to nicely apply 2D preprocessing and search methods to 3D data

saving computation time and complexity. Nevertheless, all methods proposed in the following are applicable to any type of 3D data.

Meaningful Structures. For many applications it is necessary to extract meaningful structures which enable a semantic description of complex scenes. If using 3D data, we decided to focus on geometric aspects. Human-made environments – like walls, floors, and furniture – consist of large planar structures. Therefore, it is a reasonable step to find planar surfaces within a given 3D point set. It is assumed that preceptions of halls, offices, and meeting rooms can be categorized in a proper way using planar structures, because they provide more stable features compared to colors, textures, and materials occurring in different indoor scenarios.

In principal there are three possibilities to extract planes from a 3D point set. First, the Random Sample Consensus (RANSAC) algorithm [34] can be used to fit robustly plane models in 3D data, possibly in combination with the Iterative Closest Points (ICP) algorithm [35] or SIFT features for refining the planes [36,37]. Second, the Expectation Maximization (EM) algorithm can be used to adjust the number of planes and estimates the locations and orientations by maximizing the expectation of a log-likelihood function [38,39]. Finally, region growing based approaches start from an initial triangle mesh and merge adjacent planar triangles iteratively [40].

In the following, a combination of seeded region growing [41] and RANSAC based on special values holding the correlating arrangement of points is introduced. The main idea is to decompose the point cloud into planarly connected regions and to extract planes in these regions for refinement.

First, oriented particles similar to Fua's approach [42] are defined for each point: A point's normal is computed using a point set $\{\boldsymbol{p}_i \mid \boldsymbol{p}_i \in \mathcal{N}_{3\times3}\}$ defined on the 8-neighborhood of the Swissranger image plane. The normal $\boldsymbol{n}_c$ of the current point $\boldsymbol{p}_c$ is determind by the principal component analysis of the points $\{\boldsymbol{p}_i\}$. The deviation σ_c of the point $\boldsymbol{p}_c$ to the fitted plane classifies whether a point is *locally planar* ($\sigma_c < \theta_\sigma$) or *nonplanar* [43].

Second, this set of 3D points annotated with their normals is decomposed into connected regions using region growing. Iteratively, points are selected randomly as seeds of regions and extended with points of the 8-neighborhood $\mathcal{N}_{3\times3}$ if four criteria are fulfilled. Two criteria are defined on the particles themselves, which are the validation of the points generated by the preprocessing and the local planarity as defined above. The other two criteria are computed on pairs of particles – the conormality and coplanarity measurement defined by Stamos and Allen [43]. Two points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are conormal, when their normals $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ hold:

$$\alpha = \cos^{-1}(\boldsymbol{n}_1 \cdot \boldsymbol{n}_2) < \theta_\alpha \tag{9}$$

Two points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are coplanar if the distance d

$$\begin{aligned} d &= \max(\mid \boldsymbol{r}_{12} \cdot \boldsymbol{n}_1 \mid, \mid \boldsymbol{r}_{12} \cdot \boldsymbol{n}_2 \mid), \\ \boldsymbol{r}_{12} &= \boldsymbol{p}_1 - \boldsymbol{p}_2 \end{aligned} \tag{10}$$

is smaller than a threshold θ_d. The distance d is computed with respect to the orientation and the distance of the oriented particles.

As a result, a set of mainly planar connected patches is constructed. On each of these regions several runs of the RANSAC algorithm extract the largest and smoothest planes. This step can be seen as a postprocessing step in which basically the parameters $\boldsymbol{n}_c, d_c$ of the planes $\{\mathcal{P}_c\}$ are refined. Due to oversegmentation neighboring planar patches which are close to each other (so-called *close* patches) and belong to the same affine plane have to be merged. A plane is chosen randomly and merged with planes within a region of interest (ROI) fulfilling the angle condition (Eq. 9).

Figure 6 presents exemplary photos of the six room scenarios – two offices, two halls, and two meeting room – and planar patches produced by the algorithm introduced above using 3D point clouds provided by the Swissranger SR3000. The thresholds mentioned here were set (for the current point $\boldsymbol{p}_c$) to $\theta_\alpha = 10°$, $\theta_d = 0.2 \cdot z_c$, and $\theta_\sigma = \bar{\sigma} + \sqrt{\frac{1}{n}\sum_{c=1}^{n}(\sigma_c - \bar{\sigma})^2}$ where n is the number of valid points per frame and $\bar{\sigma} = \frac{1}{n}\sum_{c=1}^{n}\sigma_c$ is the mean deviation.

4.2 Feature Extraction

For classification an extraction of meaningful features from the given planes is required. The aim is to classify a perception of a room (here: one frame of the 3D ToF sensor) while e.g. a robot enters the room. The result of the classification should be a hypothesis which room type was entered, even if the robot has not seen this particular room before.

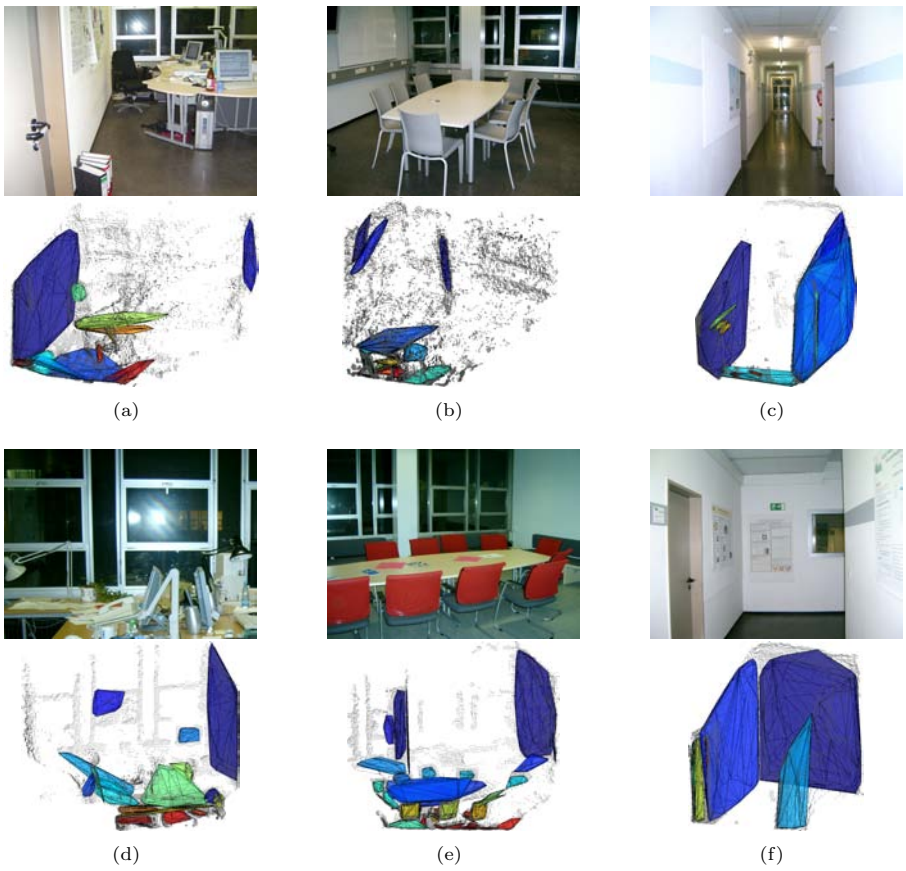
As well defined feature vectors have to fulfill several conditions it is not suitable to use all plane parameters merged into one vector as features for classification as proposed by Lourenco [44]. The features should not only be independent from colors and textures in the scene, which is implemented by the extracted planar structures, but they should also be invariant with respect to changes in the absolute number of planes, changes in view angle and view direction of the camera, and invariant to in-class variation of the furniture configuration. In the following, different aspects of the planar patches $\{\mathcal{P}_i\}$ in a frame are examined as first simple features for classification concerning the conditions listed above:

(i) Number of Points per Patch: $\forall i : n_i = \frac{|\mathcal{P}_i|}{\sum_j |\mathcal{P}_j|}$.

(ii) Angles between Patch Normals: $\forall i \neq j : \alpha_{ij} = \cos^{-1}(\boldsymbol{n}_i \cdot \boldsymbol{n}_j)$. Alternatively, to introduce structural information, angles α'_{ij} only between close patches can be computed.

(iii) Ratios between Sizes of Patches: $\forall i \neq j : r_{ij} = \frac{\min(|\mathcal{P}_i|,|\mathcal{P}_j|)}{\max(|\mathcal{P}_i|,|\mathcal{P}_j|)}$.

Finally, the feature vectors (FV1, FV2, FV3, FV4) are computed as histograms over these terms $n_i, \alpha_{ij}, \alpha'_{ij}, r_{ij}$ where the values in the bins are normalized to the range $[0, 1]$.



(a) (b) (c)

(d) (e) (f)

Fig. 6. First, exemplary photos and 3D point clouds of the training set are shown: the trained (a) office, (b) meeting room, and (c) hall. Second, the rooms for testing are displayed: the tested (d) office, (e) meeting room, and (f) hall.

4.3 Experiments and Discussion

For the following experiments 300 frames of two different offices, two halls, and two meeting rooms were acquired. The camera was positioned at a height of 145 cm (robot's camera head) and rotated horizontally 30° left/right and vertically 10° up/down in order to simulate a more or less random view on the room while entering. Also, the rooms chosen had significant differences in the layout within a room type as shown in Fig. 6. One office, one meeting room, and one hall (Fig. 6(a), 6(b), 6(c)) form the training set where 270 frames per room are used to train the classifiers and the remaining 30 frames to test the performance in recognizing (recog) an already seen room. 300 frames per room of the three other rooms (Fig. 6(d), 6(e), 6(f)) form the *main* test set for examining the performance of our system in categorizing (catego) percepts of rooms which

	NN		SVM		GMM	
	recog	catego	recog	catego	recog	catego
FV1	0.91	0.62	0.92	0.64	0.90	0.65
FV2a	0.71	0.51	0.71	0.50	0.73	0.68
FV3	0.89	0.65	0.84	0.67	0.88	0.71
FV4	0.83	0.54	0.77	0.53	0.78	0.52
FV1, FV3	0.90	0.79	0.93	0.77	0.89	0.77
FV1, FV2a	0.92	0.69	0.97	0.68	0.91	0.74
FV1, FV2b	0.91	0.68	0.88	0.65	0.86	0.66
FV1, FV3, FV4	0.89	0.78	0.90	0.77	0.94	0.81
FV1, FV2b, FV3, FV4	**0.97**	**0.79**	**0.99**	**0.79**	**0.97**	**0.81**

Table 1. This table presents results of the recognition (recog) and categorization (catego) using different feature vectors. Three classifiers are tested: a neuronal network (NN), a support vector machine (SVM), and a gaussian mixture model (GMM). FV1 describes the histogram of the relative sizes of the patches, FV2a the histogram of angles between all patches, FV2b the median of these angles, FV3 the histogram of the angles between close patches, and FV4 the histogram of the ratios between sizes of patches.

have not be seen before. We intentionally started with a very small training set containing a single room per category in order to show the generalizability of the learned model.

Three different classifiers are used to examine the proposed features in Section 4.2. The examined feature vectors are the number of points (FV1), the angles between patches (FV2a) and the median over these angles (FV2b), the angles between close patches (FV3), and the ratio of number of points between pairs of patches (FV4). The features are tested separately and in combination. A neural network (NN) [45], the support vector machine SVM^{light} (SVM) [46,47] and a gaussian mixture model (GMM) [48] are used for the classification task.

Table 1 presents all classification results for different feature vectors and combinations of them. The first four rows show results using the feature vectors FV1, FV2a, FV3, and FV4 in isolation. FV1 and FV3 turn out as features which contribute most to a good feature vector with 0.90 correct recognition of known rooms and 0.65 right categorization of new rooms. Combining these two features (FV1 and FV3) improves the rates up to 0.93 and 0.79, respectively. The categorization can be further improved up to 0.81 if the feature vector FV4 is added while the recognition rate is increased up to 0.99 using FV2b. As an assumption it can be stated that GMMs provide the most stable and proper classifiers using [FV1 FV2b FV3 FV4] as a feature vector. Round about 75% of the false classified vectors contains a mix up between meeting room and office. Since both room categories have commonalities like a large table area in the middle of the room, this is an expected result.

For additional experiments extra offices were recorded. Four of the now six different offices have a similar layout with two opposing work places while the other two rooms contain only a single work place. At least 0.69 of the four double-

place offices are categorized properly while only 0.34 to 0.51 of the single-place office percepts are classified correctly. If the training data is extended with frames of a single-place room the categorization rate of all offices can be increased to 0.88 on average.

Eighty percent of successful room categorization indicates that these planar structures on the 3D point clouds provide meaningful information about categories of rooms whereon feature vectors suitable for classification can be defined.

5 Conclusion

In this paper we present three possible approaches mainly using structural information and context for scene modeling. In simpler planar scenarios where scenes can be seen as configuration of parts spatial relationships as well as results from the object and relation naming process are used to deal with uncertainties from the object recognition process. For more general cases statistical dependencies between different modalities can be analyzed through cross-situational learning. Here, translation models from one data source to an other source have to be learned. We implemented exemplary a system inferring from word sets to visual compounds consisting of boundary fragments and spatial relationships between them. For modeling more complex indoor scenarios like different room types in a more view point independent manner using 3D data seems to be reasonable. Extracted planar patches and statistics on these patches describe the coarse layouts of room types independent from colors, textures, and design of the furniture and objects typical for these room types.

In a more general learning system of a robot these three approaches can be combined to enable a learning on different granularity levels. Such a process could start from a distinction between different room types, go to learning the important functional regions, and finish in a detailed object learning within the constraint setup of this functional region.

References

1. Vernon, D.: Cognitive vision: The case for embodied perception. Image and Vision Computing, Special Issue on Cognitive Vision **26** (2008) 127–141
2. Kintsch, W., van Dijk, T.A.: Toward a model of text comprehension and production. Psychological Review **85** (1978) 363–394
3. Johnson-Laird, P.N.: Mental models: Towards a cognitive science of language, inference, and consciousness. Harvard University Press (1983)
4. Zwaan, R.: The immersed experiencer: Toward an embodied theory of language comprehension. The Psychology of Learning and Motivation **44** (2004)
5. Lipson, P.R.: Context and Configuration Based Scene Classification. PhD thesis, MIT (1996)
6. Neumann, B., Mller, R.: On scene interpretation with description logics. In: Cognitive Vision Systems. Volume 3948. (2006) 247–275

7. Wachsmuth, S., Sagerer, G.: Bayesian networks for speech and image integration. In: National Conf. on AI. (2002) 300–306
8. Wachsmuth, S.: Multi-modal Scene Understanding Using Probabilistic Models. Ibidem Verlag (2001)
9. Barnard, K., Duygulu, P., de Freitas, N., Forsyth, D., Blei, D., Jordan, M.I.: Matching words and pictures. Journal of Machine Learning Research **3** (2003) 1107–1135
10. Jamieson, M., Fazly, A., Dickinson, S., Stevenson, S., Wachsmuth, S.: Learning structured appearance models from captioned images of cluttered scenes. In: Intl. Conf. on Computer Vision. (2007) 1–8
11. Lowe, D.G.: Object recognition from local scale-invariant features. In: Intl. Conf. on Computer Vision, Corfu, Greece (1999) 1150–1157
12. Brown, P.F., Della Pietra, V.J., Della Pietra, S.A., Mercer, R.L.: The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. **19** (1993) 263–311
13. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Europ. Conf. on Computer Vision. (2002) 97–112
14. Wachsmuth, S., Stevenson, S., Dickinson, S.: Towards a framework for learning structured shape models from text-annotated images. In: HLT-NAACL Workshop on Learning word meaning from non-linguistic data. (2003) 22–29
15. Jamieson, M., Dickinson, S., Stevenson, S., Wachsmuth, S.: Using language to drive the perceptual grouping of local image features. In: Intl. Conf. on Compter Vision and Pattern Recognition. (2006) 2102–2109
16. Moringen, J., Wachsmuth, S., Dickinson, S., Stevenson, S.: Learning visual compound models from parallel image-text datasets. In: DAGM Symposium on Pattern Recognition. (2008)
17. Opelt, A., Fussenegger, M., Pinz, A., Auer, P.: Weak hypotheses and boosting for generic object detection and recognition. In: Europ. Conf. on Computer Vision. Volume 3022. (2004) 71–84
18. Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm. Trans. on Pattern Analysis and Machine Intelligence **10** (1988) 849–865
19. Moringen, J.: Lernen von wort-form-korrespondenzen aus bildern und bildunterschriften. Technical report, Bielefeld University (2007)
20. Melamed, D.: Automatic discovery of non-compositional compounds in parallel data. In: Conf. on Empirical Methods in Natural Language Processing. (1997)
21. Brill, E.: Some advances in transformation-based part of speech tagging. In: National Conf. on AI. Volume 1. (1994) 722–727
22. Abney, S.: Parsing by Chunks. In: Principle-Based Parsing. (1991)
23. Abney, S.: Partial Parsing via Finite-state Cascades. In: ESSLLI Robust Parsing Workshop. (1996)
24. Strat, T.M., Fischler, M.A.: Context-based vision: Recognizing objects using both 2D and 3D imaging. In: Trans. on Pattern Analysis and Machine Intelligence. Volume 13. (1991) 1050–1065
25. McKeown, D.M., Harvey, W.A., McDermott, J.: Rule-based interpretation of aerial imagery. In: Readings in Computer Vision: Issues, Problems, Principles, and Paradigms. (1987) 415–430
26. Murphy, K., Torralba, A., Freeman, W.T.: Using the forest to see the trees: A graphical model relating features, objects, and scenes. In: Advances in Neural Information Processing Systems. Volume 16. (2003)
27. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: Intl. Conf. on Computer Vision and Pattern Recognition. Volume 2. (2006) 2137–2144

28. Szummer, M., Picard, R.W.: Indoor-outdoor image classification. In: Intl. Workshop on Content-Based Access of Image and Video Databases. (1998) 42–51
29. Paek, S., Chang, S.F.: A knowledge engineering approach for image classification based on probabilistic reasoning systems. In: Intl. Conf. on Multimedia and Expo. Volume 2. (2000) 1133–1136
30. Torralba, A.: Contextual priming for object detection. Intl. Journal of Computer Vision **53** (2003) 153–167
31. Swadzba, A., Wachsmuth, S.: Categorizing perceptions of rooms using 3d features. In: Intl. Workshops on Statistical Techniques in Pattern Recognition, Orlando, Florida, USA (2008) submitted.
32. Weingarten, J., Gruener, G., Siegwart, R.: A state-of-the-art 3D sensor for robot navigation. In: Intl. Conf. on Intelligent Robots and Systems. (2004)
33. Swadzba, A., Liu, B., Penne, J., Jesorsky, O., Kompe, R.: A comprehensive system for 3D modeling from range images acquired from a 3D ToF sensor. In: Intl. Conf. on Computer Vision Systems. (2007)
34. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In: Commun. ACM. Volume 24. (1981) 381–395
35. Besl, P.J., McKay, N.D.: A method for registration of 3D shapes. Trans. on Pattern Analysis and Machine Intelligence **14** (1992) 239–256
36. Nüchter, A., Surmann, H., Hertzberg, J.: Automatic model refinement for 3D reconstruction with mobile robots. In: Intl. Conf. on Recent Advances in 3D Digital Imaging and Modeling. (2003) 394–401
37. Lee, S., Jang, D., Kim, E., Hong, S., Han, J.: A real-time 3D workspace modeling with stereo camera. In: Intl. Conf. on Intelligent Robots and Systems. (2005) 2140–2147
38. Liu, Y., Emery, R., Chakrabarti, D., Burgard, W., Thurn, S.: Using EM to learn 3D models of indoor environments with mobile robots. In: Intl. Conf. on Machine Learning. (2001)
39. Lakaemper, R., Latecki, L.J.: Using extended EM to segment planar structures in 3D. In: Intl. Conf. on Pattern Recognition. (2006) 1077–1082
40. Hähnel, D., Burgard, W., Thrun, S.: Learning compact 3D models of indoor and outdoor environments with a mobile robot. Robotics and Autonomous Systems **44** (2003) 15–27
41. Adams, R., Bischof, L.: Seeded region growing. Trans. on Pattern Analysis and Machine Intelligence **16** (1994) 641–647
42. Fua, P.: From multiple stereo views to multiple 3D surfaces. Intl. Journal of Computer Vision **24** (1997) 19–35
43. Stamos, I., Allen, P.K.: Geometry and texture recovery of scenes of large scale. Computer Vision and Image Understanding **88** (2002) 94–118
44. Lourenco, A., Freitas, P., Ribeiro, M.I., Marques, J.S.: Detection and classification of 3D moving objects. In: Mediterranean Conf. on Control and Automation. (2002)
45. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Parallel Data Processing. Volume 1. (1986) 318–362
46. Vapnik, V.N.: The Nature of Statistical Learning Theory. (1995)
47. Joachims, T.: Learning to Classify Text Using Support Vector Machines. PhD thesis, Cornell University (2002)
48. Fink, G.A.: Developing HMM-based recognizers with ESMERALDA. In: Lecture Notes in AI. Volume 1692. (1999) 229–234