A Detailed Analysis of a New 3D Spatial Feature Vector for Indoor Scene Classification

Agnes Swadzba[a,*], Sven Wachsmuth[a]

[a] *Applied Informatics, Faculty of Technology, Bielefeld University*

Abstract

Enhancing perception of the local environment with semantic information like the room type is an important ability for agents acting in their environment. Such high-level knowledge can reduce the effort needed for, e.g., object detection. This paper shows how to extract the room label from a small amount of room percepts taken from a certain view point (like the door frame when entering the room). Such functionality is similar to the human ability to get a scene impression from a quick glance. We propose a new 3D spatial feature vector that captures the layout of a scene from extracted planar surfaces. The trained models emulate the human brain sensitivity to the 3D geometry of a room. Further, we show that our descriptor complements the information encoded by the Gist feature vector – a first attempt to model the mentioned brain area. The global scene properties are extracted from edge information in 2D depictions of the scene. Both features can be fused resulting in a system that follows our goal to combine psychological insights on human scene perception with physical properties of environments. This paper provides detailed insights into the nature of our spatial descriptor.

Keywords: indoor scene classification, 3d vision, planar surfaces, spatial cognition, robot vision

[*] corresponding author

Email addresses: aswadzba@techfak.uni-bielefeld.de (Agnes Swadzba), swachsmu@techfak.uni-bielefeld.de (Sven Wachsmuth)

URL: http://aiweb.techfak.uni-bielefeld.de/user/aswadzba (Agnes Swadzba)

1. Introduction

Recognizing the type of an indoor room, e. g., "living room", is a basic spatial ability of agents. For example, such high-level concepts of the surrounding can be used to activate top-down knowledge that guides the visual analysis in further tasks, e. g., enhancing object detection by context [1, 2, 3]. Humans have special capabilities for recognizing quickly the gist of a scene. It has been hypothesized that object detection plays a minor role while the scene geometry seems to encode the relevant scene information. This hypothesis is supported by the discovery of a brain area called Parahippocampal Place Area (PPA) which gives strong response to stimuli with a spatial layout [4] especially from full-view indoor scenes [5], but less response to close-up views of scene-relevant objects (e. g., a kitchen oven) and no response to arrays of objects without three-dimensional spatial context. Recent findings propose that in addition to PPA also the area V1, the Retrosplenial Cortex (RSC), and the Lateral Occipital Complex (LOC) contain information to distinguish among natural scene categories [6]. The V1 area seems to be sensitive to local orientations and spatial frequencies while the activity patterns in PPA, RSC, and LOC go beyond low-level features and play a more direct role in humans' ability to categorize scenes. Especially, it is proposed that PPA and RSC are responsible for extracting differences in spatial layout among different categories.

Torralba and colleagues [7] have designed a V1-like feature, the so-called *Gist* feature, which is a scene representation based on global distribution of spatial frequencies. It classifies 2D images of outdoors scenes based on their spatial properties assembling the so-called *spatial envelope*. Basically, characteristics like naturalness, openness, roughness, expansion, and ruggedness are encoded. A problem of the Gist feature is that it fails with indoor scenes. This has been tackled recently by using local object information [8]. In contrast to this approach, – along the line of the above psychological findings – we propose to extend the Gist feature with a feature vector [9] that captures like PPA the 3D spatial layout of a room. It is derived from planar surfaces that usually implement most man-made environments [10]. As 3D cameras like the Kinect camera are now available at a reasonable price we show in this paper results for dense point clouds acquired with a SwissRanger camera. Concretely, we show a system that determines the room type of a short sequence of point clouds taken from a certain view point while panning and tilting the camera. But in prin-

ciple it is also thinkable to use planar patches extracted from 2D images. Approaches such as Yu's depth-ordered plane extraction from line segments [11] show that computer vision research has made progress towards this direction but results are still not detailed enough to be used reliably for scene classification. The contribution of our paper is a holistic scene model that captures the spatial layout of rooms. The advantage of such models is their independence from specific object detections and knowledge about interdependencies between objects and room types.

The basic idea has already been presented in our ACCV paper [9]. In this paper, we concentrate on an in-depth analysis of the 3D spatial feature vector. We will analyze on our 3D database the reasons for success and failure of our categorization approach. Further, we will explore empirically and systematically how specific spatial information, namely angles, sizes of areas, shapes, and area ratios, contribute to the classification of a specific room type. Last, we will apply our approach to a newly assembled database of Kinect recordings giving us the opportunity to test the 3D spatial feature vector quantitatively on real-world data. This paper is structured as follows. Section 2 discusses related work on scene classification, especially, focused on the indoor scene classification problem. Section 3 gives details on plane surface extraction in dense 3D point clouds. Section 4 presents our new 3D feature vector that captures the 3D spatial layout of rooms, gives examples for different room types, shows it distinctiveness, and compares it to the Gist feature vector. Section 5 shows how to derive and fuse scene models and how to adjust the importance of a patch characteristic for a specific room type. Section 6 evaluates the classification performance of the final system and analyzes in-depth the 3D feature vector. Especially, the contribution of the different patch characteristics is further explored providing details on how to choose the appropriate spatial information necessary to learn a specific room model. Section 7 discusses the categorization results achieved for real-world data.

2. Related Work

Related work on scene classification comes from robotics and computer vision. Approaches from robotics mostly concentrate on recognition and categorization of indoor scenes based on data acquired with robot platforms driving around. Approaches from computer vision categorize 2D images using databases of images collected from the web.

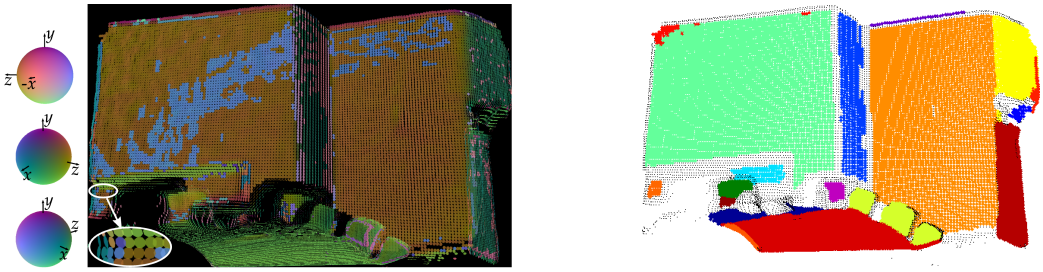
Early spatial abilities of robots have been developed in the context of determining drivable areas followed by the decomposition of maps into places. Places are mostly defined as some continuous areas, which are found through detection of transitions like doorways [12, 13] or segmentation of the open space into connected room-like places [14]. The re-detection of known rooms is often realized by comparing features in the current camera image or laser scan to those in saved views [15, 16, 17]. First attempts into the direction of recognizing the type of a room have been done for the basic place types "corridor", "hallway", "room" and "doorway" using simple geometrical features of 360° laser scans [12]. Further refining of the concept "room" into sub-concepts like "kitchen", "lab", or "office" has been realized by detecting objects and using interdependencies between objects and room types. These interdependencies can come from an indoor ontology [18] or are learned from training data [19, 20, 21, 22, 2].

A well-known approach used for real world scene recognition that bypasses detection of individual objects is the so-called Gist descriptor [23]. It captures the *Spatial Envelope* of a scene by encoding its naturalness, openness, roughness, expansion, and ruggedness. Impressive performance can be achieved for outdoor scenes while indoor scenes remain a problem. Recently, Torralba and colleagues have tackled the break down of the Gist feature for indoor rooms by using object information that comes from histograms of candidate regions computed from prototype images [8]. The usage of local scene information for scene classification has been explored by a variety of computer vision approaches. Mostly, visual words assembling a codebook are computed by clustering local features such as textures or intermediate themes and associating them with a room type by assigning probabilities or encoding their occurrences [24, 25, 26, 27, 28, 29].

3. Planar Surface Extraction

As typical man-made indoor environments mostly consist of planar surfaces it is a reasonable step to represent the surrounding 3D scene by a collection of planar patches extracted from the 3D point cloud of the current environment.

In principle, there are three main algorithms for extracting planar surfaces from 3D data: Expectation Maximization (EM), Region Growing (RG), and RANdom SAmple Consensus (RANSAC). The advantage of the RANSAC algorithm [30] is that planes are fitted robustly to data while



(a) oriented particles (b) extracted planar patches

Figure 1: (a) shows a set of 3D points transformed to oriented particles. The patches are colored according to their orientations to the axes of the coordinate system. Parallel planes like the wall in the back and the cupboard doors are colored equally (e. g., orange) or with a complementary color (e. g., blue) depending on the sign of the normal. (b) planar patches extracted via region growing using a conormality and coplanarity constraint and a RANSAC refinement.

outliers are omitted. In general, three points are chosen randomly determining a plane and the remaining points are added if they fulfill the plane equation. This basic algorithm can be enhanced by color information [10] or refined by some ICP [31] iterations. The mentioned methods perform well on convex scenes such as halls but may encounter problems in cluttered and noisy scenes like living rooms. EM techniques provide a solution to this problem as for a given number of planes their parameters are estimated such that the likelihood of the (noisy) data is maximized. As the number of planes must be known in advance a second optimization step is needed to determine the correct number of planes. This is often done by minimizing the Bayesian Information Criterion [32]. Planes are dropped if they encode redundant information in the model [33, 34]. In contrast to EM, Region Growing techniques do not need to know the number of planes in advance. In general, the algorithm starts with a seed point and extends the current set with points from the neighborhood if a homogeneity criterion (mostly, planarity criterion) is fulfilled. After the growing has stopped a new plane is initialized by selecting randomly a new seed point [35]. Resulting patches are often noisier compared to those produced by EM and requires therefore subsequent smoothing.

Our scenario requires a methodology that allows segmenting a dense 3D point cloud of a cluttered scene into an arbitrary number of bounded planar patches. We have implemented a region growing approach over 3D

points that are enhanced with normal vectors encoding the local surface properties. In recent approaches it has been shown that this local surface information has a positive effect on the surface extraction [36, 37]. Further, neighboring points can be computed efficiently using the 2D image plane for 3D cameras like the SwissRanger or the Kinect. Our algorithm starts with transforming a set of 3D points (sorted row-wise according to their position on the camera's image plane) to a set of oriented particles [38]. The normal vector $\vec{n}_i$ of a point $\vec{f}_i$ is computed by applying Principal Component Analysis (PCA) to a set of neighboring points selected through determining the 8-neighborhood $\mathcal{N}_{3\times3}$ of the point $\vec{f}_i$ on the image plane of the camera. Figure 1(a) shows the estimated oriented particles of an example point set. In a second step, initial planar patches are extracted by by iteratively adding points from the current 8-neighborhood $\mathcal{N}_{3\times3}$ if the conormality and coplanarity constraints are fulfilled [39]. Two points $\vec{f}_1$ and $\vec{f}_2$ are conormal if the acute angle α between their normals $\vec{n}_1$ and $\vec{n}_2$ is smaller than a threshold θ_α (here, $\theta_\alpha = 10°$):

$$\text{conormal}\left(\vec{f}_1,\ \vec{f}_2\right) \ \Leftrightarrow\ \alpha < \theta_\alpha, \quad \alpha = \begin{cases} \arccos(\vec{n}_1 \cdot \vec{n}_2) & :<= \frac{\pi}{2} \\ \pi - \arccos(\vec{n}_1 \cdot \vec{n}_2) & : \text{else} \end{cases} \tag{1}$$

Two points $\vec{f}_1$ and $\vec{f}_2$ are coplanar if their distance d computed with respect to the orientation and distance of the oriented particles is smaller than a threshold θ_d:

$$\text{coplanar}\left(\vec{f}_1,\ \vec{f}_2\right) \ \Leftrightarrow\ d < \theta_d, \quad \begin{aligned} d &= \max\left(|\vec{r}_{12} \cdot \vec{n}_1|,\ |\vec{r}_{12} \cdot \vec{n}_2|\right), \\ \vec{r}_{12} &= \vec{f}_1 - \vec{f}_2 \end{aligned} \tag{2}$$

Finally, the extracted regions are refined by some RANSAC iterations. Figure 1(b) shows some extracted planar patches. The interested reader is referred for further details to [40].

4. Features for Indoor Scene Classification

This section presents the computation of our proposed 3D spatial feature which is based on histograms of plane characteristics. Further, the well-known Gist feature vector is presented which has been developed for scene classification from 2D image utilizing the overall scene information encoded in scene edges [7].

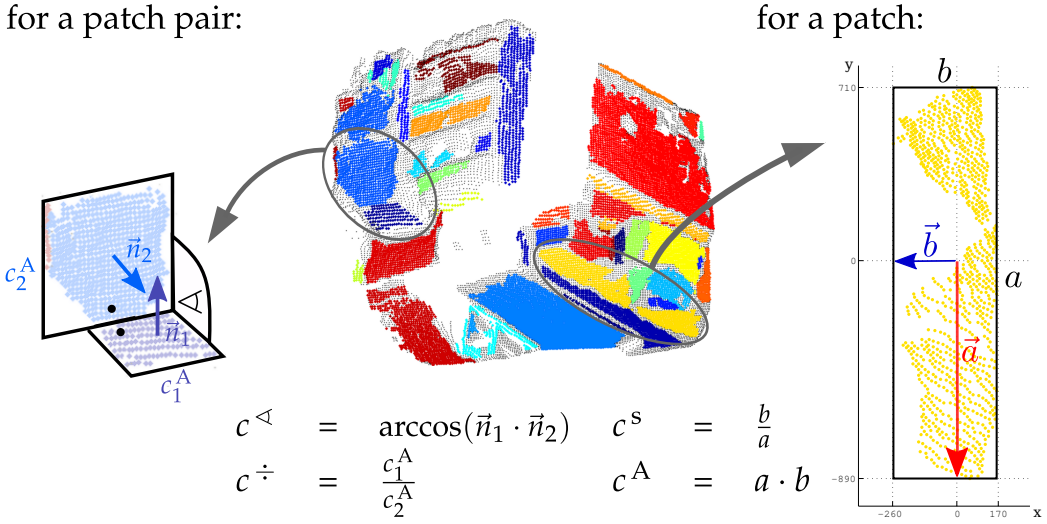


Figure 2: This figure shows exemplarily the 3D point cloud of livingroom `liv.5` (frame 162) with points highlighted by color according to their patch membership; the left close-up shows the acute angle between two patches; the right close-up shows a plane transformed so that the vectors indicating the two largest variance directions in the data are parallel to the coordinate axis [9]; below the four patch characteristics extracted from an example patch or patch pair are given.

4.1. The 3D Spatial Feature Vector

This section is going to present the computation of our new 3*D spatial feature vector* $\vec{x}^{3D}$. This feature vector aims at capturing the spatial layout of a room in such a way that it is robust to different furnitures and layouts of rooms and different views on them.

A robot can easily capture a dense 3D point cloud using cameras like the SwissRanger or Kinect. A set of planar patches – denoted as $\left\{\mathcal{P}_j\right\}_{j=1}^{m}$ – is extracted from such a point cloud using the method described in the previous section. These patches encode the geometry of a scene independent from scene colors and textures. As shown in Figure 2 each patch can be approximated by a minimum bounding box enclosing the patch points (see right part of Figure 2). The Principal Component Analysis (PCA) of these points deliver the three vectors $\vec{a}$, $\vec{b}$, and $\vec{n}$. The vector $\vec{a}$ indicates the direction of the largest variance in the data, $\vec{b}$ the orthogonal direction with the second largest variance, and $\vec{n}$ the normal vector. The

bounding box is aligned parallel to the axis given by $\vec{a}$ and $\vec{b}$. The box extension is indicated by the scalar values a and b. The definition of the 3D spatial feature vector is based on four patch characteristics: the *shape* of a patch (c^{s}), its *size* (c^{A}), the *angle* ($c^{\sphericalangle}$) between a pair of patches, and their *size ratio* ($c^{\div}$). These four characteristics are visualized in Figure 2.

Definition. ***Patch characteristics.***

Each planar patch element in $\left\{\mathcal{P}_j\right\}_{j=1}^{m}$ *is characterized by:*
c_j^{s}, *the shape characteristic and* c_j^{A}, *the size characteristic.*

Each pair of planar patches in $\left\{(\mathcal{P}_k, \mathcal{P}_l)\right\}_{(k,l)=(1,2)}^{(m-1,m)}$ *is characterized by:*
$c_{kl}^{\sphericalangle}$, *the angle characteristic and* $c_{kl}^{\div}$, *the size ratio characteristic.*

Characteristics for a patch. The computation of the shape characteristic c^s and the size characteristic c^{A} is inspired by classical 2D shape analysis. The shape of a patch $\mathcal{P}_j$ is encoded by term c^{s} which is computed by dividing the smaller box edge by the larger box edge ($\rightarrow$ s(.)):

$$c^{\mathrm{s}} \;=\; \mathrm{s}\left(\mathcal{P}\right) = \frac{\min(a,b)}{\max(a,b)}. \tag{3}$$

Elongate patches have c^{s}-values near to zero while squarish patches are indicated by values near one. The second patch characteristic is the area covered by the patch. For simplicity we approximate ($\rightarrow$ A(.)) the patch area by the area of the corresponding bounding box:

$$c^{\mathrm{A}} \;=\; \mathrm{A}\left(\mathcal{P}\right) = a \cdot b. \tag{4}$$

For a set of patches $\left\{\mathcal{P}_j\right\}$ a set of m shape characteristics $C^{\mathrm{s}} = \left\{c_j^{\mathrm{s}}\right\}$ and a set of m size characteristics $C^{\mathrm{A}} = \left\{c_j^{\mathrm{A}}\right\}$ is computed:

$$\begin{aligned} \left\{\mathcal{P}_j\right\} \;&\xrightarrow{\mathrm{s}(.)}\; C^{\mathrm{s}} = \left\{c_j^{\mathrm{s}}\right\}, \\ &\xrightarrow{\mathrm{A}(.)}\; C^{\mathrm{A}} = \left\{c_j^{\mathrm{A}}\right\}, \quad j = 1, \ldots, m. \end{aligned} \tag{5}$$

Characteristics for a patch pair. The angle $c^{\measuredangle}$ and the size ratio $c^{\div}$ characteristics are computed for each patch pair with the goal to provide an encoding of a room that is independent to changes in the view or concrete patch arrangement. The angle $c^{\measuredangle}_{12}$ between two patches $\mathcal{P}_1$ and $\mathcal{P}_2$ is estimated as acute angle between their normals $\vec{n}_1$ and $\vec{n}_2$:

$$c^{\measuredangle}_{12} \;=\; \measuredangle\Big(\mathcal{P}_1, \mathcal{P}_2\Big) = \begin{cases} \arccos(\vec{n}_1 \cdot \vec{n}_2) & :<= \frac{\pi}{2} \\ \pi - \arccos(\vec{n}_1 \cdot \vec{n}_2) & : \text{else} \end{cases}. \tag{6}$$

The size ratio characteristic $c^{\div}_{12}$ is a quotient built from the size of the smaller patch divided by the size of the bigger patch:

$$c^{\div}_{12} \;=\; \mathrm{R}\Big(\mathcal{P}_1, \mathcal{P}_2\Big) = \frac{\min(c^{\mathrm{A}}_1, c^{\mathrm{A}}_2)}{\max(c^{\mathrm{A}}_1, c^{\mathrm{A}}_2)} = c^{\div}_{21}. \tag{7}$$

A value near zero indicates that patches have quite different sizes while a value near one means that they have roughly the same size. For a set of patch pairs $\left\{\left(\mathcal{P}_k, \mathcal{P}_l\right)\right\}$ $(m\cdot(m-1))/2$ angle values $C^{\measuredangle} = \left\{c^{\measuredangle}_{kl}\right\}$ and $(m\cdot(m-1))/2$ size ratio values $C^{\div} = \left\{c^{\div}_{kl}\right\}$ are computed:

$$\begin{aligned} \left\{\left(\mathcal{P}_k, \mathcal{P}_l\right)\right\} \;&\xrightarrow{\measuredangle(.)}\; C^{\measuredangle} = \left\{c^{\measuredangle}_{kl}\right\}, \\ &\xrightarrow{\mathrm{R}(.)}\; C^{\div} = \left\{c^{\div}_{kl}\right\}, \quad (k,l) = (1,2),\ \ldots,\ (m-1,m). \end{aligned} \tag{8}$$

The feature vector. For each of the four value sets C^{s}, C^{A}, $C^{\measuredangle}$, and $C^{\div}$ extracted for a given set of planar patches a histogram H(.) is computed and normalized to 1. The normalization factor for $\vec{x}^{\mathrm{s}}$ and $\vec{x}^{\mathrm{A}}$ is m as we have m planar patches and for $\vec{x}^{\measuredangle}$ and $\vec{x}^{\div}$ $m(m-1)$ because of $m\cdot(m-1)$ patch pairs. The four histogram vectors are concatenated to one feature vector $\vec{x}^{\mathrm{3D}}$ defining the 3*D spatial feature vector*:

$$\begin{aligned} \vec{x}^{\mathrm{3D}} &= \left(\vec{x}^{\measuredangle}, \vec{x}^{\mathrm{s}}, \vec{x}^{\div}, \vec{x}^{\mathrm{A}}\right)^{\mathrm{T}} && \text{with} \\ \vec{x}^{\mathrm{s}} &= \frac{1}{m}\cdot \mathrm{H}_{\mathrm{s}}\Big(C^{\mathrm{s}}\Big), && \vec{x}^{\mathrm{A}} = \frac{1}{m}\cdot \mathrm{H}_{\mathrm{A}}\Big(C^{\mathrm{A}}\Big), \\ \vec{x}^{\measuredangle} &= \frac{2}{m(m-1)}\cdot \mathrm{H}_{\measuredangle}\Big(C^{\measuredangle}\Big), && \vec{x}^{\div} = \frac{2}{m(m-1)}\cdot \mathrm{H}_{\div}\Big(C^{\div}\Big), \end{aligned} \tag{9}$$

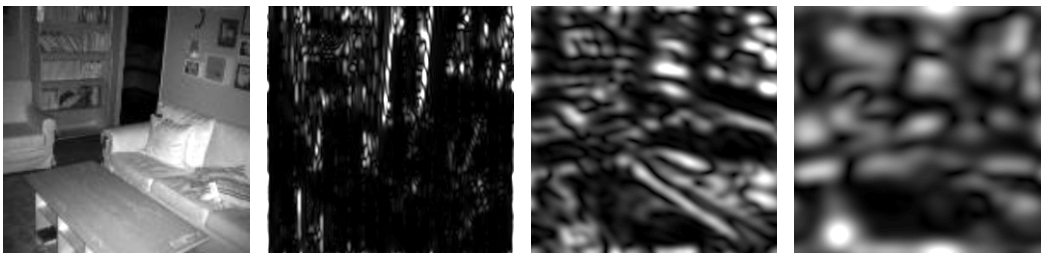


Figure 3: This figure shows some result responses of different Gabor filters applied to the 2D low-resolution image on the left.

$H_{\measuredangle}(.)$ divides the range [0°90°] into 9 bins, $H_s(.)$ and $H_{\div}(.)$ divide the range [01] into 5 bins, and $H_A(.)$ has 6 bins with the bin boundaries [0, $(25\text{cm})^2$, $(50\text{cm})^2$, $(100\text{cm})^2$, $(200\text{cm})^2$ $(300\text{cm})^2$, ∞]. The result of the concatenation is a 25-dimensional feature vector $\vec{x}^{3D}$.

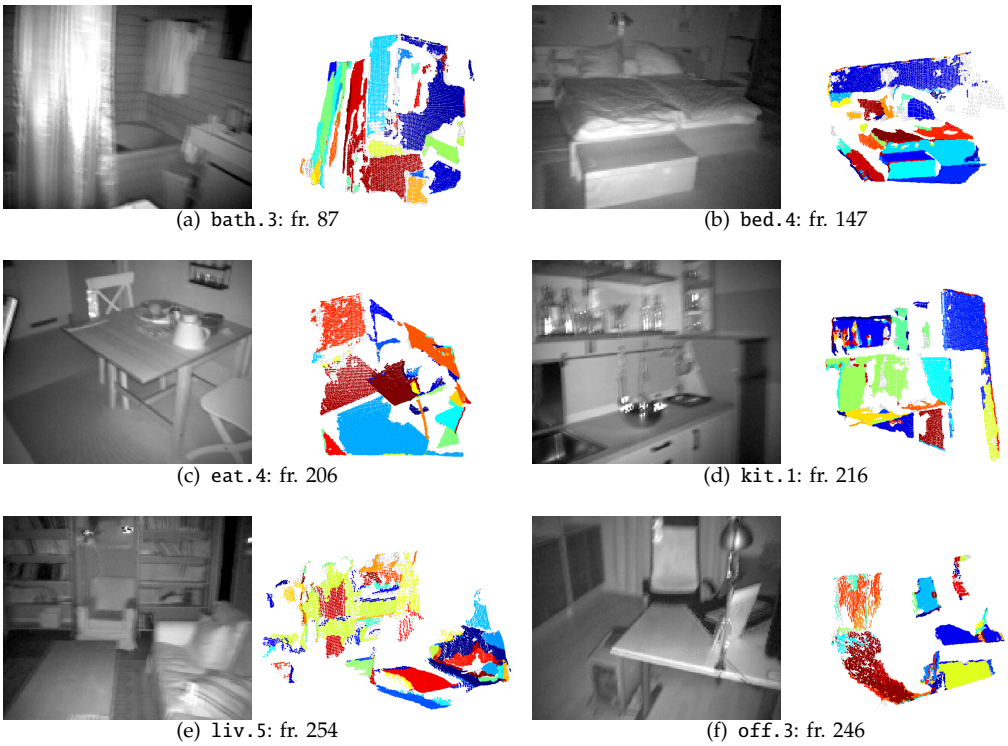
4.2. The Gist Feature Vector

A well-known global feature vector for determining the overall scene of a 2D image has been proposed by Torralba and colleagues [7]. The computation of this so-called Gist feature relies on a wavelet image decomposition. Each image location is represented by the output of filters tuned to different orientations and scales. We have applied Torralba's implementation [1] which consists of a set of Gabor filters of 8 orientations and 4 scales. When a SwissRanger camera is used to capture 3D data the corresponding amplitude image (176×144 resolution) can be utilized as gray-scale image for extracting Gist information. As Torralba's implementation requires an image with an edge size of power 2, we have clipped the SwissRanger image to 144×144 and have interpolated it bilinearly to 256×256. The resulting feature vector $\vec{x}^{\text{Gist}}$ is 512-dimensional. As shown Figure 3 the Gist feature vector encodes the spatial layout of a room based on the edge information introduced by the layout.

4.3. Detailed Insights on the 3D and Gist Feature in the Indoor Context

This section will give some insights into the nature of the 3D and Gist feature vector when applied to the indoor classification problem. The analysis is conducted on our IKEA database (→ Section 6.1, a database that

[1] http://people.csail.mit.edu/torralba/code/spatialenvelope/



(a) bath.3: fr. 87 (b) bed.4: fr. 147

(c) eat.4: fr. 206 (d) kit.1: fr. 216

(e) liv.5: fr. 254 (f) off.3: fr. 246

Figure 4: The pictures show views on rooms from the IKEA database which area nearest to the centroids of the room type clusters.

contains frames from 28 rooms spread over 6 room types). We compute for each room type in the database its class centroid using the 3D spatial features extracted from all frames of a specific room type. Figure 4 shows for each room type the frame which is closest to the corresponding centroid. The amplitude image and the extracted planar surfaces are displayed. The selected frames are remarkably intuitive views on the rooms, especially, if compared to the frames which are farthest away from the class centroid (see Figure 5. Such frames consist at least partially of close-up views on doors or show room type unrelated parts like the sideboard in the case of the eating place. Figure 6 presents as example the 3D spatial feature vectors of centroid-closest frames. The discriminative power of the 3D spatial feature is illustrated by the confusion matrix in the left part of the figure. The gray values in the matrix represent the distances between the mean histograms of different room types obtained using the Histogram

(a) bath.1(205) (b) bed.3(140) (c) eat.1(191) (d) kit.2(191) (e) liv.2(25) (f) off.2(25)

Figure 5: These pictures show frames which 3D features are farthest away from the corresponding class centroids.

Intersection Kernel [41]:

$$d(\vec{x}_1, \vec{x}_2) = \sum_{i=1}^{\sharp\text{bins}} \min(x_1^i, x_2^i) \tag{10}$$

We further visualize the separation quality of the different room categories by plotting per category the 50 feature vectors that are closest to the mean histogram of the category. This is done in 2D via classical multidimensional scaling [42] to preserve the original inter-point distances and can be looked up in Figure 7. It can be seen that four clusters which correspond to the room categories bathroom, bedroom, kitchen, and living room are clearly visible. Only office and eating place are not clearly distinguishable in the 2D plot. Recalling the spatial layouts of the two room types this is reasonable since both room types share some similarities like being assembled of at least a table and a chair. Figure 7 shows also the plots of the sub-features $\vec{x}^{<}$, $\vec{x}^{\text{s}}$, $\vec{x}^{\text{A}}$, and $\vec{x}^{\div}$. They illustrate which plane property describes a room class best. For example, angle, shape, and size characteristic separate "bathroom" percepts from other percepts. "Bedroom" percepts only differ clearly from other rooms when the size characteristic is observed and "eating place" percepts when the size ratio characteristic is observed. In both characteristics also a definite "kitchen" cluster is visible. A clear "office" cluster is not noticeable in the defined patch characteristics. A further analysis of the correlations between the sub-vectors is given in Section 6.3.

Figure 8 shows a 2D plot of some Gist features computed on amplitude images in the 3D IKEA database. Per category again the mean feature vector and the 50 closest vectors are displayed. The clusters for bathroom, bedroom, and a subpart of the office category are already in 2D nicely separated. But as plotting 512-dimensional feature vectors in 2D while

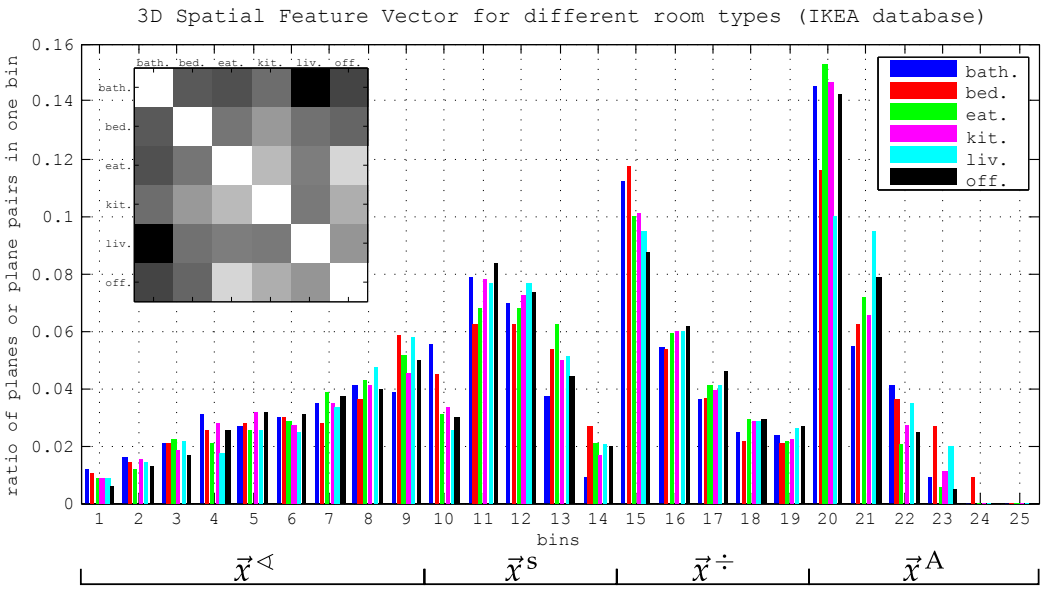


Figure 6: Example 3D spatial feature vectors for the six room types bathroom, bedroom, eating place, kitchen, living room, and office. The confusion matrix illustrates the discriminative power of the 3D feature. (best viewed in color)

preserving their inter-point distances is a hard problem we have also examined the 3D feature plot. It turns out that the "eating place" forms a compact cluster while in 3D "kitchen", a subpart of the "office" cluster, and the "living room" cluster are still intersecting and are not linearly separable. Fortunately, the 3D features form a compact cluster for "living room" so that the spatial features can contribute substantially to the categorization of "living room" percepts (see Section 6.2).

5. Training Room Models and Combining Single Classifications

After suitable features have been computed classifiers are trained to estimate the boundaries between the different room types. For each room type a discriminant model is learned which can be used to compute the probability that a feature belongs to a class. For example, this probability could dependent on the distance of a feature vector to the class boundary. We refer to this set of classifiers as the *holistic scene model*.

Given a set of 6 classes

$$\Omega \;=\; \{\omega_i\} = \{\texttt{bath.}, \texttt{bed.}, \texttt{eat.}, \texttt{kit.}, \texttt{liv.}, \texttt{off.}\} \tag{11}$$

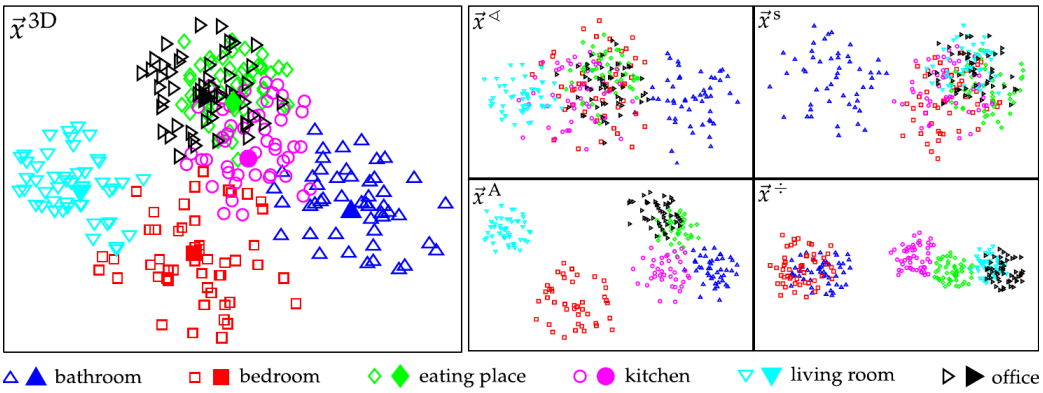


Figure 7: (left) Plotting in 2D 25-dimensional 3D features $\vec{x}^{3D}$ using the multidimensional scaling method. Per category the mean feature vector and the 50 closest feature vectors are plotted. ▲, ■, ♦, ●, ▼, and ► label the mean vectors. The Euclidean distance is used as inter-point distance. Four of six categories (bathroom, bedroom, kitchen, living room) are already in 2D nicely clustered. (right) shows plots of the sub-feature vectors $\vec{x}^{\sphericalangle}$, $\vec{x}^{S}$, $\vec{x}^{A}$, and $\vec{x}^{\div}$. They give an impression by which plane property a class separation may be caused.

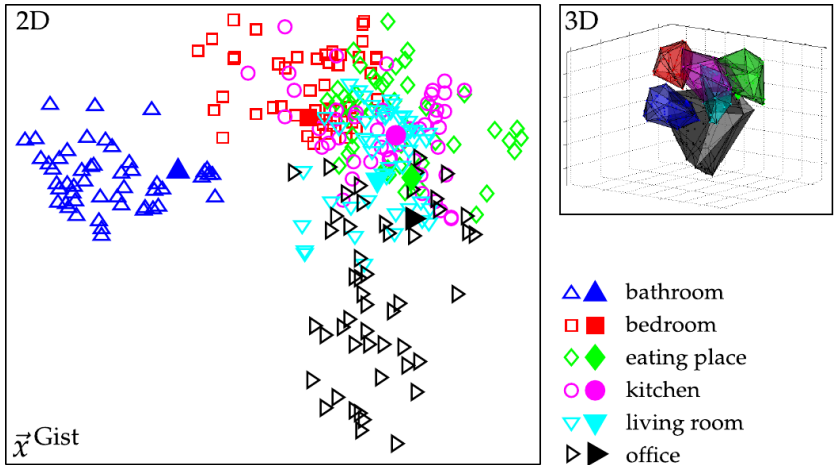


Figure 8: Using multidimensional scaling to plot the 512-dimensional Gist feature vectors in 2D and 3D. The means and the 50 closest vectors are displayed. The clusters of bathroom, bedroom, and a sub-cluster of the office category are already nicely separated in 2D. In 3D the features of the eating place category occurred to be linear separable from the other features while living room and kitchen features are still mixed.

a vector of discriminant functions $G(\vec{x})$ is learned where each function $g_i(\vec{x})$ maps the n-dimensional feature vector $\vec{x}$ on a scalar value d_i encoding how likely $\vec{x}$ lies in class ω_i [43]:

$$\begin{aligned} G &: \mathbb{R}^n \rightarrow \mathbb{R}^{|\Omega|} \\ g_i &: \mathbb{R}^n \rightarrow \mathbb{R}, \quad i = 1, \ldots, |\Omega| = 6 \\ G\left(\vec{x}\right) &= \left(d_1 = g_1(\vec{x}), \ldots, d_6 = g_6(\vec{x})\right)^{\mathrm{T}} \quad \text{with } d_i = \begin{cases} > 0 & : \vec{x} \in \omega_i \\ < 0 & : \vec{x} \notin \omega_i \end{cases}. \end{aligned} \tag{12}$$

The final decision on the best class model is based on determining that class boundary to which the feature vector $\vec{x}$ has the largest distance. Mathematically, this can be formulated by a decision function $E(.)$ which maps the vector $\vec{d}$ on a binary vector $\vec{e}$ with the i_1-th component equal to 1 if $i_1 = \arg\max_i d_i$ is the maximum value in $\vec{d}$. All other components are set to 0:

$$\begin{aligned} E\left(\vec{d}\right) &= \left(e_1, \ldots, e_6\right), \text{ where} \qquad e_i = \begin{cases} 1 & : i = i_1 \wedge \frac{d_{i_1} - d_{i_2}}{d_{i_1}} > \theta_{\text{rej}} \\ 0 & : \text{else.} \end{cases} \\ i_1 &= \arg\max_i d_i, \qquad i_2 = \arg\max_{i \setminus i_1} d_i \end{aligned} \tag{13}$$

This equation also incorporates the rejection of uninformative scene views by setting the i_1-th component to 0 if the corresponding distance value d_{i_1} does not differ significantly from the second largest distance value d_{i_2}. During evaluation $\theta_{\text{rej}} = 0.05$ has turned out to be best suited as not more than 20% of the test frames have been rejected.

A classification decision based on a single feature vector might be quite noisy because of the limited view of the SwissRanger camera. Fortunately, the envisioned scenario where a standing robot scans an unknown room by panning and tilting its camera allows a stabilization of the room categorization by fusing classifier responses of several consecutive frames $\{ \mathcal{F}_i \}_{i=0}^n$. This set of frames is transformed to a set of feature vectors $\{ \vec{x}_i \}_{i=0}^n$. The corresponding set of classifier responses $\{ \vec{d}_i \}_{i=0}^n$ can be seen as results of independent classifiers which means that simple classifier combination schemes are an obvious choice [44]. According to Kittler's theoretical framework [45] the fusion can then be done through a product, sum, max, min, median, or majority vote rule. In an experimental comparison, Kittler

and colleagues have shown that the sum rule outperforms the other classifier combination schemes. Therefore, we fuse the normalized classifier responses of consecutive frames using the following sum rule:

$$\begin{aligned} \vec{d} &= \sum_{i=0}^{n} L\left(\vec{d_i}\right), \quad \vec{d_i} = G\left(\vec{x_i}\right), \\ L\left(\vec{d}\right) &= \left(l(d_1), \ldots, l(d_6)\right)^{\mathrm{T}}, \quad l(x) = \frac{1}{1+e^{-x}}. \end{aligned} \tag{14}$$

The logistic function $l(.)$ normalizes the distances to the class boundaries to values in the range of $[\,0, 1\,]$. The decision function $E(.)$ can also be applied to the summed classifier responses. This combination scheme can be easily extended by other feature vectors simply through adding the classifier responses of all feature vectors:

$$\vec{e} = E\left(\sum_{i=0}^{n}\left(L\left(G^{\mathrm{3D}}(\vec{x}_i^{\mathrm{3D}})\right) + L\left(G^{\mathrm{Gist}}(\vec{x}_i^{\mathrm{Gist}})\right)\right)\right). \tag{15}$$

So far, we have concatenated the four sub-vectors $\vec{x}^{\prec}$, $\vec{x}^{\div}$, $\vec{x}^{\mathrm{s}}$, and $\vec{x}^{\mathrm{A}}$ encoding the spatial layout as one feature vector. Instead of concatenation, one could also think of other combination strategies like the classifier fusion through summing up sub-responses as proposed in Equation 14. However, the unweighted summing of classifier responses can only form a baseline. Therefore, we have applied an AdaBoost algorithm [46] for a binary classification task to choose for each room type the best weak classifiers and their weights. As formulated above, we see our classification task as a binary decision per room type (e.g., "bath" or "no bath", see Equation 12). Therefore, the boosting algorithm is applied to each room type separately to find per room type the best combination of weak classifiers. Concretely, we define the family of weak classifiers $\mathcal{G}$ as set of four weak classifiers based on the four sub-features $\vec{x}^{\prec}$, $\vec{x}^{\div}$, $\vec{x}^{\mathrm{A}}$, and $\vec{x}^{\mathrm{s}}$:

$$\mathcal{G} = \{g_{\prec}, g_{\div}, g_{\mathrm{A}}, g_{\mathrm{s}}\}. \tag{16}$$

Given the training set $(\vec{x}_1, y_1)$, …, $(\vec{x}_m, y_m)$ where $y_i \in Y = \{-1, +1\}$ and the number of iterations T the distribution w_i^1 is initialized as $w_i^1 = \frac{1}{m}$ with $i = 1, \ldots, m$. In iteration t $(= 1, \ldots, T)$ the classifier g^t is chosen that

maximizes the absolute difference of the weighted error ϵ_t from 0.5 with respect to the distribution w^t:

$$g^t = \arg\max_{g^t \in \mathcal{G}} |0.5 - \epsilon^t|, \qquad \text{where} \quad \epsilon^t = \sum_{i=1}^{m} w_i^t \, I(y_i \neq g^t(\vec{x}_i)) \tag{17}$$

The distribution is updated by

$$w_i^{t+1} = \frac{1}{\bar{w}^{t+1}} \cdot w_i^t \exp -\alpha^t y_i g^t(\vec{x}_i), \quad \bar{w}^{t+1} = \sum_i w_i^t e^{-\alpha^t y_i g^t(\vec{x}_i)}, \tag{18}$$

and the final classifier for a room type is then defined as:

$$G^{\text{room}}(\vec{x}) = \sum_{t=1}^{T} \alpha^t g^t(\vec{x}), \qquad \text{where} \quad \alpha^t = \frac{1}{2} \ln \frac{1 - \epsilon^t}{\epsilon^t}. \tag{19}$$

It provides votes for the room types so that the overall classification is a maximum decision over the votes given by:

$$G^{\text{3D Boost}}(\vec{x}) = (G^{\text{bath.}}(\vec{x}), \ldots, G^{\text{off.}}(\vec{x})), \qquad \text{(compare to Equation 12).} \tag{20}$$

This allows to fuse the responses of this new strong spatial classifier with the Gist classifier and over time in the same way as proposed in Equation 15.

6. Evaluation

In order to evaluate the classification performance of our new 3D spatial feature vector we have collected a new 3D indoor database. Details on this database are give in Section 6.1. Using this database Support Vector Machines (SVMs) are trained [47]. Per class one SVM model $g_i(\vec{x})$ is trained using the one-vs-all training scheme where all frames in the 3D database (without the frames of the test sequences) belonging to one room type form the positive samples and all remaining frames are used to sample the same amount of negative samples. The SVM^{light} library [2] [48] is utilized using the built-in Radial Basis Function (RBF) kernel. The RBF parameter γ and the regularization parameter c are optimized in a 10-fold-cross-validation

[2] `http://svmlight.joachims.org`

scheme choosing a model with a small number of support vectors while reducing the classification error (here, $c = 900$ and $\gamma = 2$). In Section 6.2 we compare the classification rates of the 3D spatial feature vector with other feature extraction approaches and analyze in-depth for which classes the spatial feature vectors perform well and the reasons for failures. In Section 6.3 we analyze the influence of the four patch characteristics on the classification of indoor scenes.

6.1. The 3D IKEA database

Existing databases of indoor percepts [3] [4] [5] are either not taken from a robots perspective or are assembled from 2D color images of indoor rooms [6] or contain only outdoor scenes for which 3D data is estimated [7] [50]. The 3D spatial layout of an indoor scene can be inferred from a 2D image using line segments [51, 52, 53, 11, 54], but the resulting 3D layout is currently to errorneous and coarse for capturing all relevant spatial structures about scene-typical furniture that is necessary for our 3D spatial feature vector. Exemplarily, we have applied the geometric reasoning approach [51] to a 2D picture of a bathroom. Figure 9 shows the achieved result. The general frame of the room is estimated partially, but important mid-level spatial structures like the toilet or the washbasin are missed at all. Inferring the 3D structure using the LabelMe3DToolbox [50] also fails for indoor images.

Instead, we have compiled an own indoor database [8] to explore the applicability of the 3D spatial layout of indoor scenes for categorizing data of a room a mobile robot will be typically confronted with. This database contains dense 3D point clouds from a sufficient large number of different rooms. Sensors like the SwissRanger or the Kinect camera are best suited for capturing in real-time dense 3D point clouds from indoor rooms, especially, from homogeneous furniture areas. As the data acquisition has been done in 2009, we have taken the SwissRanger SR3100 camera to a regular IKEA home-center [9]. The exhibition is ideally organized for our

[3] http://web.mit.edu/torralba/www/indoor.html [8]
[4] http://www.emt.tugraz.at/~pinz/data/tinygraz03 [49]
[5] http://vision.stanford.edu/Datasets/SceneClass13.rar [27]
[6] http://cogvis.nada.kth.se/COLD [16]
[7] http://people.csail.mit.edu/brussell/research/LabelMe3D/LabelMe3dDownload.html
[8] http://www.techfak.uni-bielefeld.de/~aswadzba/3D-IKEA-database.tar.gz
[9] http://www.ikea.com/us/en

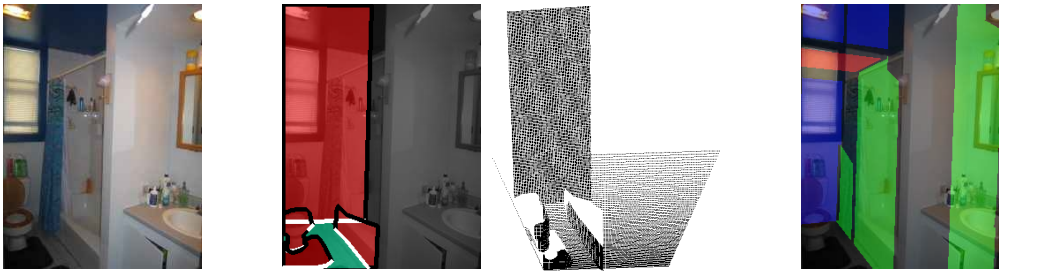


Figure 9: Two results of extracting 3D information from a single 2D picture of a bathroom. The tested approaches are the LabelMe3DToolbox [50] extracting 3D information from user annotations and Geometric Reasoning [51] infering the spatial layout from detected line segments.

purpose as it is assembled by opened 3D boxes showing example rooms. 28 rooms of 6 room types have been scanned for round about 30 seconds with a SwissRanger camera that was panned and tilted continuously by ca. 40° left/right and ca. 10° up/down. As the average frame rate of the camera is ca. 10 fps this results in an acquisition of 300 to 400 frames per room. The camera was positioned at a height of ca. 140cm simulating the robot's view. Figure 10 shows digital photos of the scanned rooms taken at the positions of the 3D camera. As IKEA has stores all over the world, a database on IKEA data can be easily extended and holds furniture arrangements available in real rooms all over the world.

6.2. Detailed Analysis of Categorization Performances

This section is going to compare the classification performances that can be achieved by different features on the 3D IKEA database. The evaluated features are our proposed 3D spatial feature vector $\vec{x}^{3D}$ (→ Section 4.1), the Gist feature vector $\vec{x}^{Gist}$, the Depth-Gist feature vector $\vec{x}^{DGist}$ where the Gist computation is applied to depth images, and Lazebnik's feature vector $\vec{x}^{SP}$ which is based on a spatially ordered visual vocabulary. Their 2D local features are edge points at two scales and eight orientations and SIFT descriptors of 16×16 pixel patches.

6.2.1. Setup of the Experiments

The envisioned analysis require a separated training and test set. Both sets are generated by choosing randomly from the 3D database one room per room type as test sequence. The remaining rooms of one type form

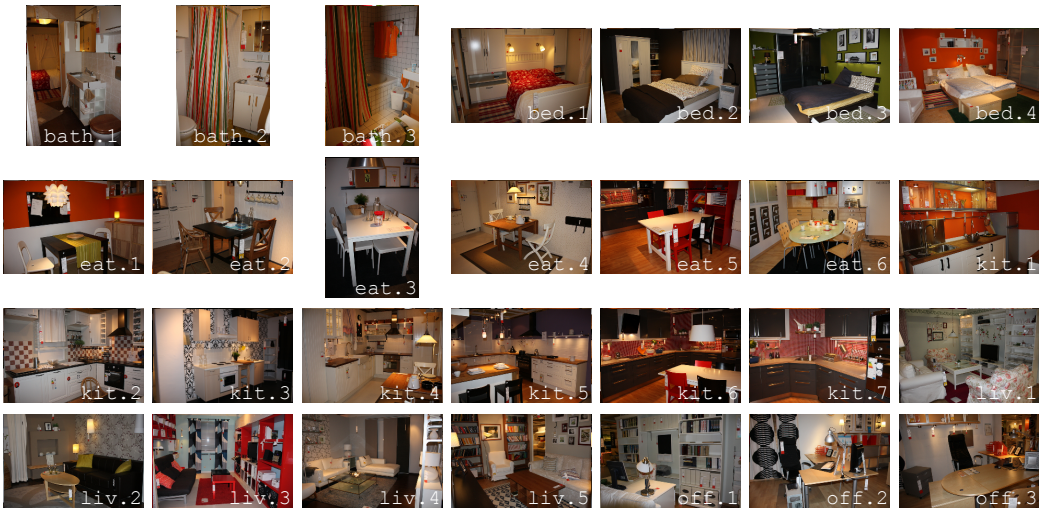


Figure 10: This figure shows photos of 28 rooms of 6 room types that have been scanned with the SwissRanger SR3100 in an IKEA home-center. The color images have been taken from the view of the 3D camera. The database contains 3 bathrooms, 4 bedrooms, 6 eating places, 7 kitchens, 5 living rooms, and 3 offices.

the training set for this room class. This selection is repeated 10 times. The classification rates are averaged over these 10 runs. The following evaluation is done on classification curves computed for an increasing fusion window Δt (concretely from 1 to 300 frames of a test sequence) and on confusion matrices computed for $\Delta t = 70$. This fusion window might be a typical fusion window for robotic applications as this would result in a "I'm taking a quick look around"-behavior of a robot. Figure 11 shows the classifications rates originally presented in our ACCV paper [9]. As we aim in this paper for a detailed analysis of the applicability of the 3D spatial feature vector to the scene classification task, we are going to take a deeper look on the so far achieved classification results. First, our attention was drawn by the low classification rates for the bedroom class (0.53) and the office class (0.41) if using 3D and Gist features in combination (see $(\vec{x}^{3D}, \vec{x}^{Gist})$). For the bedroom class it turned out that by accident in half of the cases the room `bed.1` was selected as test room. A detailed look on this sequence revealed that only 30% of the frames show meaningful views on the bedroom. A meaningful view contains the bed like shown in Figure 12 as this is the most important furniture in a bedroom. Unfortunately, the majority of frames in sequence `bed.1` shows the front

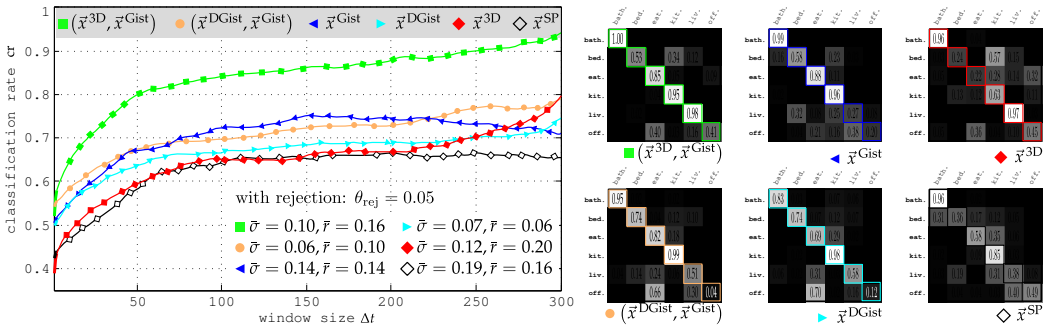


Figure 11: This figure shows the development of the classification results while enlarging the window Δt from 1 to 300 frames ($\rightarrow$ x-axis). The frames are fused according to Equation 14. Example confusion matrices have been picked for $\Delta t = 70$. The tested feature types and combinations are: ■ $(\vec{x}^{3D}, \vec{x}^{Gist})$, ● $(\vec{x}^{DGist}, \vec{x}^{Gist})$, ◀ $\vec{x}^{Gist}$, ▶ $\vec{x}^{DGist}$, ◆ $\vec{x}^{3D}$, and ◇ $\vec{x}^{SP}$. These classification results are the original results from the ACCV paper [9]. (best viewed in color)

Figure 12: The sequence of the bedroom `bed.1` consists of 30% meaningful frames showing the bed and 70% not-meaningful frames showing the front or the side of cupboards.

or the side of a cupboard which could be located anywhere in a flat. These not-meaningful views mislead the classification and as sequence `bed.1` was selected as test room disproportionately more often than the other bedroom sequences the bad classification rate of `bed.1` has a strong impact on the overall classification rate of the bedroom class. To ensure a fair comparison between the different room types, we have decided to re-run the experiment with a new sampling of test rooms. Special attention was paid to the distribution of the test rooms. Every room appears equally often over the test sets but the concrete test sets are still arranged randomly. The generated test sets are listed in Figure 14(a).

Figure 13 shows the overall classification rates for the new test sets. The effect of well-balanced test sets is directly visible in the confusion matrix of $(\vec{x}^{3D}, \vec{x}^{Gist})$. The classification rate of the bedroom class is increased

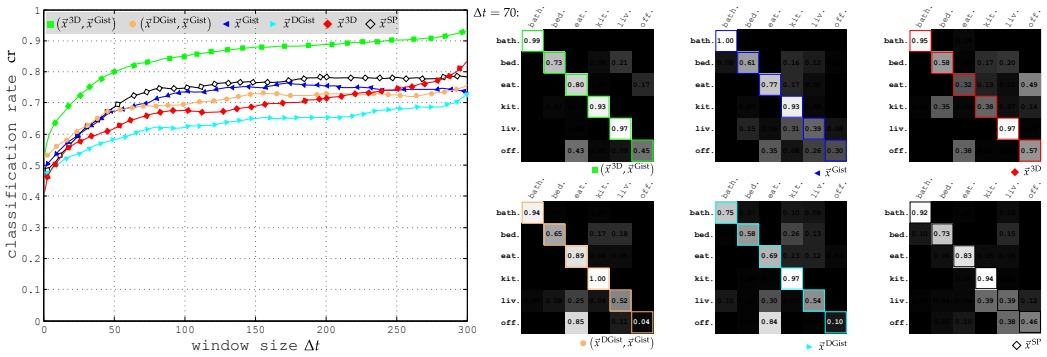


Figure 13: Overall classification rates and confusion matrices for the new generated test sequences. The fusion window Δt is varied from 1 to 300. The example confusion matrices are given for $\Delta t = 70$.

significantly from 0.53 to 0.73 while the classification rates of the other room types stay stable. The improvement mostly originates from the better classification performance of the 3D spatial feature vector $\vec{x}^{3D}$.

6.2.2. Discussion of the Results

The equal appearance of every room in the test sets allows to analyze systematically the classification rates of the specific room sequences. Figure 15 shows the classification rates for each class per run. One important observation is that the classification of test sequences is 100% correct or fails completely (curve converges to 0.00) if more and more frames are considered. There exists two types of failures. Some sequences are misclassified already for small fusion windows (like the "office"-curve in run4). Some sequences are misclassified for larger fusion windows (e.g., $\Delta t > 200$ like the "office"-sequence in run6). The first failure occurs if the trained room model does not fit the test sequence at all. The second failure occurs if the sequence partly consists of not-meaningful frames which overrule the meaningful frames. The sequences that are classified correctly if all frames are considered have a better meaningful to not-meaningful ratio. The larger the amount of meaningful frames the earlier 100%-classification rate can be achieved for a sequence. For a detailed analysis of the test sequences we have rated the classification performance of each test sequence over the 10 runs. Three labels are possible: "correct" (c) if the sequence is classified 100% correct, at least when all frames are considered, "failed" (f) if the classification failed completely already for small fusion windows,

run	1	2	3	4	5	6	7	8	9	10
bath	2	1	3	2	3	1	2	3	3	1
bed.	4	3	3	2	1	2	2	1	4	3
eat.	4	6	2	4	2	1	3	5	5	6
kit.	5	5	2	3	6	4	6	1	7	4
liv.	3	4	5	2	3	2	5	4	1	1
off.	2	3	3	2	1	1	2	3	1	3

(a) distribution of rooms over the test sets

room ♯	1	2	3	4	5	6	7
bath	c, c, c	c, c, c	c, c, c, c				
bed.	c, c	c, c, c	c, c, c	f, f			
eat.	c	c, p	c	c, c	c, c	f, f	
kit.	c	c	c	c, c	c, c	c, c	c
liv.	c, c	c, c	c, c	c, c	c, c		
off.	c, p, c	f, f, f	c, c, c, c				

(b) c = correct, f = failed, p = partly

(c) typical views for

Figure 14: The tables give the distribution of rooms from the IKEA database over the 10 test runs and the rating of their classification performance according to their classification rates shown in Figure 15. Further, typical views from the failed sequences `bed.4`, `eat.6`, and `off.2` are displayed.

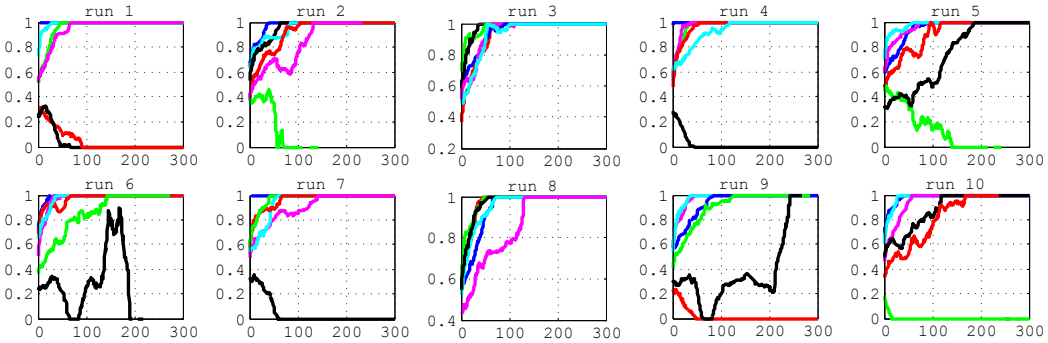
and "partly" (p) if the classification rate of a sequence drops to 0.00 for fusion windows bigger than 150 frames. Figure 14(b) shows the rating of the classification performance of each room according to this definition. It can be seen that if the classification of a specific room type fails than it is always the same specific room. In particular, the sequences `bed.4`, `eat.6`, and `off.2` are hardly classified correctly. Figure 14(c) shows typical views from these sequences.

The bad performance on the office sequence `off.2` seems reasonable as a lot of views are close-ups of the table. The spatial structure of these frames is mostly just one plane. Here, the patch pair characteristics, like angle or area ratio, are hard to compute on one planar patch resulting in an inappropriate spatial feature vector. As eating place sequences could also contain views just showing the top of a table, the confusion of office frames with eating place frames and vice versa is not surprising. A solution to this problem could be to disregard frames which contain just one dominant plane. The second typical view in the `off.2`-sequence is showing a sideboard. As sideboards could be placed anywhere in a flat this is a not-meaningful view. Only the frames showing the chair should give a strong hint for an office but seemed not to appear often enough

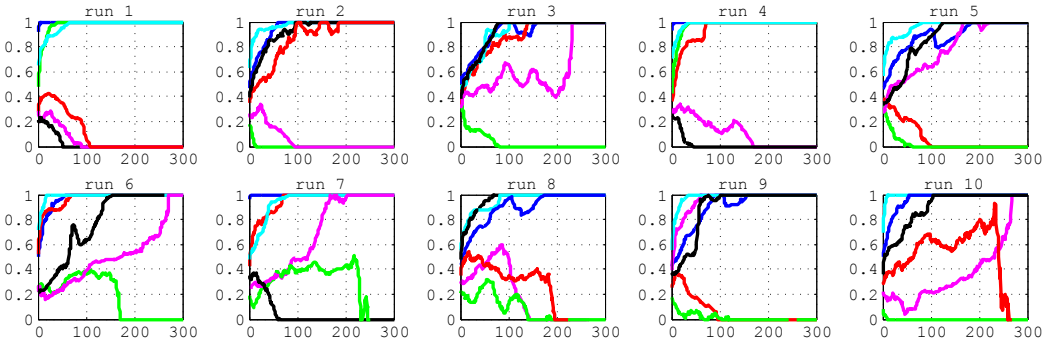
in the `off.2`-sequence (especially, in combination with the desk) to rule out the not-meaningful views. A solution could be to integrate detectors for objects that are mandatory for a specific room type (like office chair is for an office). Detector responses could increase the importance of views where these objects have been detected.

Last, one could be quite curious about the bad classification rate for the bedroom sequence `bed.4`. It seems to be a prototypical bedroom arrangement. But using the knowledge that this room is mostly mistaken for a living room we took a deeper look on the concrete arrangement of the spatial structures in this room. Most important is the constellation of the armchair, the bed, and the bench in front of the bed. The arrangement of planar patches is quite similar to a living room. The living rooms in this database typically had armchairs. The bench could be interpreted as a couch table and the bed would be a large sofa. Here, it is not so clear whether object information would improve the classification. But some statistics about the space between the patches could be helpful as table and sofa should have some space in between while this is not necessary for the bed-bench constellation.

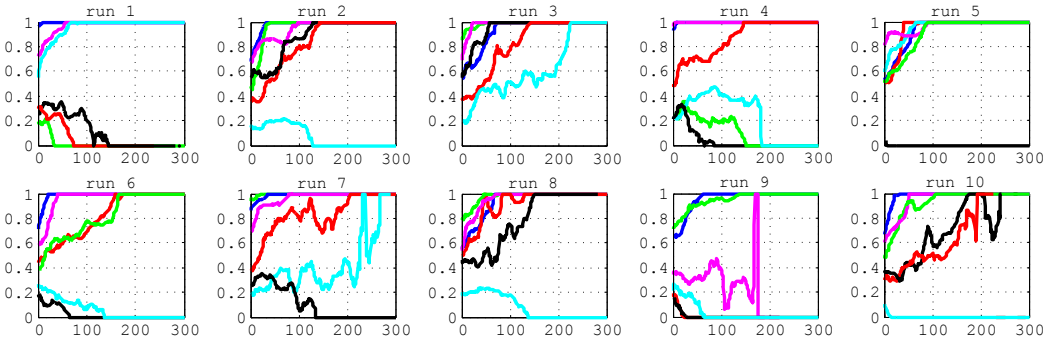
To conclude the analysis of Figure 13 and Figure 15, it could be said that with the combination of our 3D and Gist feature vector a classification performance of above 90% can be achieved on complete sequences in the IKEA database. Further, it can be seen that the 3D spatial feature vector $\vec{x}^{\text{3D}}$ often fails for kitchen and eating place sequences which are no problem for the Gist feature vector $\vec{x}^{\text{Gist}}$. For living room sequences the situation is vice versa. Here, the Gist feature vector often fails. Especially, the sequences `liv.1`, `liv.2`, and `liv.4` are misclassified. These three rooms have in common that the couch tables do not produce strong edge features and that there are a lot of curved edges compared to the other two rooms `liv.3` and `liv.5`. Such different edge information within a class is quite challenging to handle for an edge based descriptor like the Gist feature. The 3D spatial feature vector is more suitable for the living room room type as the surface structures are independent from the curvy form of the sofa or the table. The complementary behavior of both features for these two room classes leads to an improvement of the classification performance of these classes if both features are combined (see Figure 13). The overall classification rate of $(\vec{x}^{\text{3D}}, \vec{x}^{\text{Gist}})$ also outperforms the two other features chosen for comparison. Neither the Depth Gist feature $\vec{x}^{\text{DGist}}$ nor the local 2D feature based approach of Lazebnik $\vec{x}^{\text{SP}}$ can achieve comparable rates.



(a) $(\vec{x}^{3D}, \vec{x}^{Gist})$



(b) $\vec{x}^{3D}$



(c) $\vec{x}^{Gist}$

Figure 15: Classification rates of the individual rooms per run. The color-coding is **blue** for the bathrooms, **red** for the bedrooms, **green** for the eating places, **magenta** for the kitchens, **cyan** for the living rooms, and **black** for the offices. (best viewed in color)

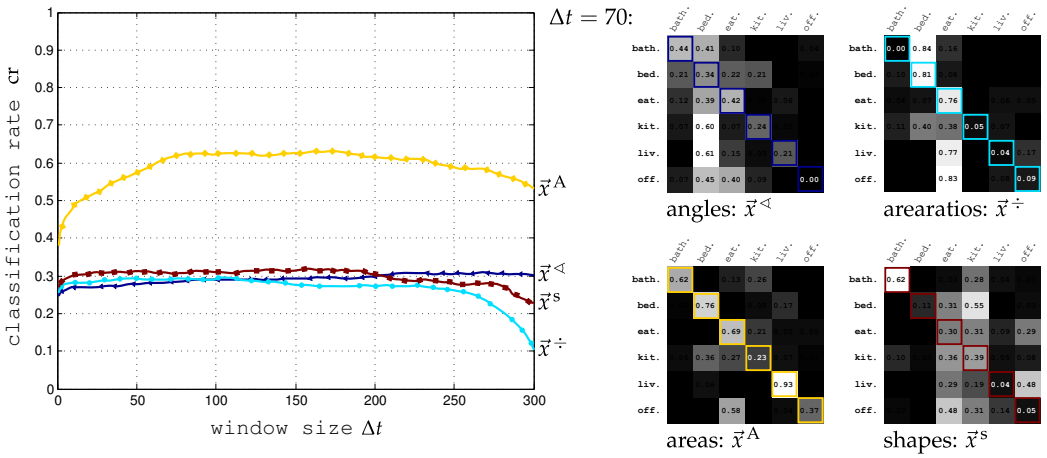


Figure 16: The figure shows the overall classification rates and confusion matrices for $\Delta t = 70$ of the four sub-vectors, $\vec{x}^{A}$, $\vec{x}^{\sphericalangle}$, $\vec{x}^{\div}$, and $\vec{x}^{s}$.

6.3. In-depth Analysis of the Spatial Sub-Vectors

The 3D feature vector $\vec{x}^{3D}$ consists of four sub-vectors capturing four properties of planar patches assembling a room (→ subsection 4.1). This section explores the contribution of each sub-vector to the classification performance of the 3D feature vector. First, the same evaluation with the same test rooms as presented above is run for the four sub-vectors $\vec{x}^{\sphericalangle}$, $\vec{x}^{\div}$, $\vec{x}^{A}$, and $\vec{x}^{s}$. Second, we will apply a boosting algorithm to determine the characteristics that contribute most to a specific room type.

6.3.1. Emperical Analysis

Figure 16 displays for the different sub-vectors the classification rates averaged over the room types and 10 runs as well as confusion matrices for $\Delta t = 70$. The curves give the classification progress when the fusion window is increased from $\Delta t = 1$ to $\Delta t = 300$. The best performing sub-vector is $\vec{x}^{A}$ with a rate of round about 0.60 followed by $\vec{x}^{\sphericalangle}$, $\vec{x}^{\div}$, and $\vec{x}^{s}$ all with a rate of round about 0.30. The observation that the classification rates start to decrease if the fusion window becomes too large indicates that over the complete sequence more frames are misclassified than labeled correctly. But the correct and wrong responses seem to be spread in a way that for smaller fusion windows enough correct responses could be accumulated or alternatively rejected successfully.

A look on the confusion matrices of the four patch characteristics reveals

further that the angle characteristic $\vec{x}^{\sphericalangle}$ contributes mostly to the bathroom and eating place models, the arearatio characteristic $\vec{x}^{\div}$ to the bedroom and eating place sequences, the shape characteristic $\vec{x}^{\mathrm{s}}$ to the bath model and little to the kitchen model. The area characteristic $\vec{x}^{\mathrm{A}}$ is the best performing one and contributes mostly to bathroom, bedroom, eating place, and living room models. The confusion matrices show also which room types share one of the four characteristics. For example, eating places and kitchen have frames with a similar patch size and shape distribution. An explanation for this effect could be the fact that in the IKEA exhibition eating places have been displayed together with kitchens which resulted in kitchen recordings where some views show also an eating place. For the eating places the situation is vice versa. In some of the eating place sequences the kitchen might be visible in the background.

6.3.2. Systematic Analysis

The empirical impression that patch characteristics contribute differently to the classification of room types can be approved systematically by applying a boosting algorithm. The basic idea is to learn per room type the optimal combination of these patch characteristics. Instead of concatenating the four sub-vectors $\vec{x}^{\sphericalangle}$, $\vec{x}^{\div}$, $\vec{x}^{\mathrm{A}}$, and $\vec{x}^{\mathrm{s}}$ to one feature vector giving them equal importance, the responses of the models $g_{\sphericalangle}$, $g_{\div}$, g_{A}, and g_{s} trained using the four sub-vectors separately are combined by a weighted sum (see Equation 19). For each room model individual weights for the four sub-vectors are learned assigning high weights to those patch characteristics that contribute most to the correct classification of the corresponding room type. The concrete algorithm is introduced in Section 5. Figure 17 shows the weights of the four sub-vectors learned for each room type. The dominant patch characteristic for the bedroom, eating place, living room, and office is the size of the patches encoded in g_{A}. The shape of patches encoded in g_{s} is mainly important for the bathroom class, but the other patch characteristics play also a significant role. The four patch characteristics contribute equally to the kitchen model. These results are aligned with the empirical observations derived from the classification rates shown in Figure 16.

As introduced in Equation 20, a collection of weighted sums, one per room type, assembles the new 3D classifier G^{3D}. Each sum adds up the four binary classifiers ($g_{\sphericalangle}$, $g_{\div}$, g_{A}, g_{s}) trained separately on the four patch characteristics. The weights are selected according to Figure 17. We refer

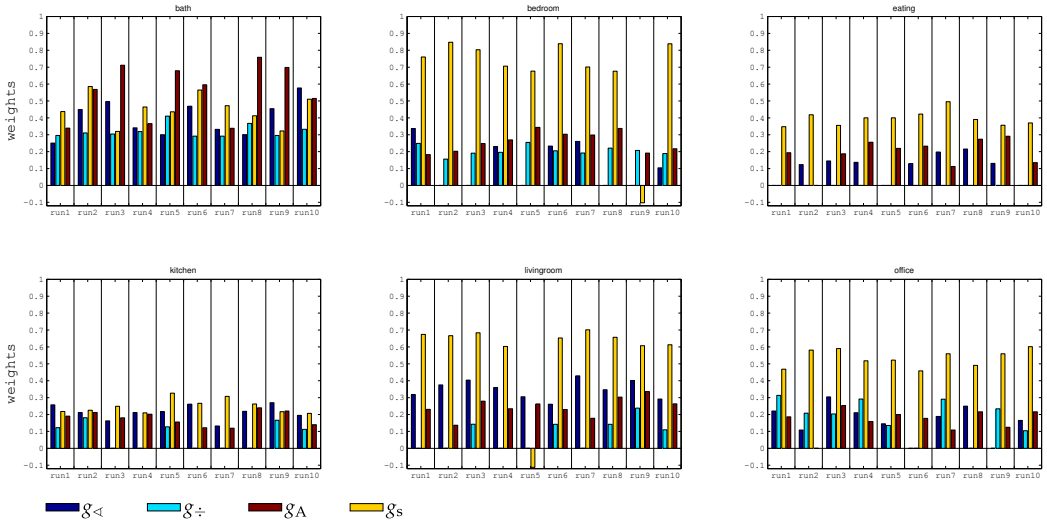


Figure 17: Weights of the sub-vectors learned for each room type using an AdaBoost algorithm. The weights are computed for 10 different training sets.

to the set of four feature vectors as $\vec{x}^{\text{3D Boost}} = (\vec{x}^{\lhd}, \vec{x}^{\div}, \vec{x}^{\text{A}}, \vec{x}^{\text{s}})$. Figure 18 shows the classification rates for $\vec{x}^{\text{3D Boost}}$ and a combination with the Gist feature $(\vec{x}^{\text{3D Boost}}, \vec{x}^{\text{Gist}})$. For comparison reasons, the baseline rates for $\vec{x}^{\text{3D}}$ and $(\vec{x}^{\text{3D}}, \vec{x}^{\text{Gist}})$ are also displayed. All rates are average values computed over the 10 test sets used in Section 6.2.1. Classification rates are similar for the baseline and boosted version of our approach. In particular, boosting does not improve the overall classification rates. But the per-class rates for the bedroom and the eating place class got improved and are decreased for the kitchen and office class. The classification of the bedroom benefits most from the boosting approach as the weight pattern for this class deviates at most from the assigning equal importance to each of the four patch characteristics. The boosting approach increases the importance of the patch size characteristic and reduces the other characteristics. The resulting classifier performs for this specific class better than the baseline classifier trained on $\vec{x}^{\text{3D}}$ which is disturbed by the other patch characteristics.

Even though no difference in the classification rates between the baseline and the boosting approach can be observed, there might be a difference in the generalization ability. To analyze this, we examine the number of learned support vectors. The assumption is that SVM models based on few support vectors generalize more than models based on many vectors.

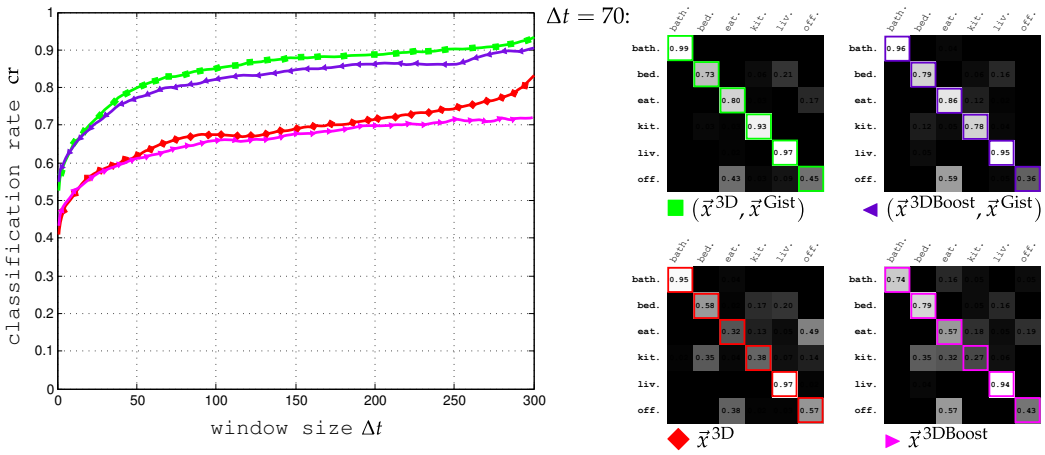


Figure 18: Comparison of the classification rates achieved by the weighted sum of the four patches ($\rightarrow \vec{x}^{\text{3D Boost}}$) and its combination with the Gist feature vector ($\rightarrow (\vec{x}^{\text{3D Boost}}, \vec{x}^{\text{Gist}})$) with the baseline features $\vec{x}^{\text{3D}}$ and $(\vec{x}^{\text{3D}}, \vec{x}^{\text{Gist}})$.

# SV	$g_{\texttt{bath.}}$	$g_{\texttt{bed.}}$	$g_{\texttt{eat.}}$	$g_{\texttt{kit.}}$	$g_{\texttt{liv.}}$	$g_{\texttt{off.}}$
$\vec{x}^{\text{3D}}$	183	221	219	273	156	160
$\vec{x}^{\prec}$	17	17	13	18	16	14
$\vec{x}^{\div}$	8	7	6	10	5	5
$\vec{x}^{\text{A}}$	10	12	9	13	9	11
$\vec{x}^{\text{s}}$	10	19	12	20	13	14
$\vec{x}^{\text{Gist}}$	228	346	399	511	461	386

Figure 19: For each room models, $g_{\texttt{bath.}}$, $g_{\texttt{bed.}}$, $g_{\texttt{eat.}}$, $g_{\texttt{kit.}}$, $g_{\texttt{liv.}}$, and $g_{\texttt{off.}}$, trained on one of the five features listed in the table the number of learned support vectors (# SV) is listed.

The reason is that models with many vectors are endangered to over-fit the database on which they are trained and might not be able to handle new data. Figure 6.3.2 lists the support vectors for the different models. It can be seen that the room models trained on the sub-vectors $\vec{x}^{\prec}$, $\vec{x}^{\div}$, $\vec{x}^{\text{A}}$, and $\vec{x}^{\text{s}}$ are much smaller than the models trained on $\vec{x}^{\text{3D}}$. Their models consist of 10 to 20 support vectors. Consequently, the boosting variant of our categorization is based on maximal 80 support vectors which is less than half of the support vectors needed in the baseline approach. To conclude, one could state that the boosting approach realizes the more abstract room models.

7. Application to Real World Data

This section is going to investigate the applicability of the 3D spatial feature vector to data from real world environments. For this purpose, the Berkeley 3D Object Dataset[10] [55] is used. This dataset is a compilation of Kinect data which was recorded and submitted by diverse people all around the world. The dataset consist of data from rooms arranged and equipped for living purposes of the residents and not for sale purposes. Section 7.1 gives further details for this dataset. Section 7.2 describes the preprocessing of the data in order to generate a suitable training base. Section 7.3 presents the classification results.

7.1. The Berkeley 3D Object Dataset

The Berkeley 3D Object Dataset is a database where people all around the world contributed Kinect recordings. Most recordings show close-up views of objects, but a subpart of the database also shows views on rooms from people's flats. Figure 20 shows the rooms we have selected to test the performance of our 3D spatial feature vector on Kinect data from real apartments recorded from people who did not have our application in mind. We have selected 37 rooms spread over the six room types. Some rooms are represented by several shots recorded from different view points. In total 74 shots are utilized for the following evaluation.

7.2. Preprocessing the Selected Data

Unfortunately, people did not provide per room a sequence of Kinect data taken while the camera is shifted and tilted simulating a robot looking around. This means that in the first place we cannot extract enough features to train our room models. But the Kinect camera provides a 3D point cloud with quite high resolution (640×480 points) and has a wider recording angle than the SwissRanger camera (see Figure 21(a)). These characteristics can be used to sample sub-windows from one Kinect point cloud. Per sub-window one spatial feature vector is computed resulting in a much bigger set of features available for training.

The raw Kinect point is quite noisy, especially, surfaces further away from the camera center are quite thickened. Therefore, the 640×480 point cloud is sampled down to 256×192. The bilinear interpolation reduces

[10] http://kinectdata.com/

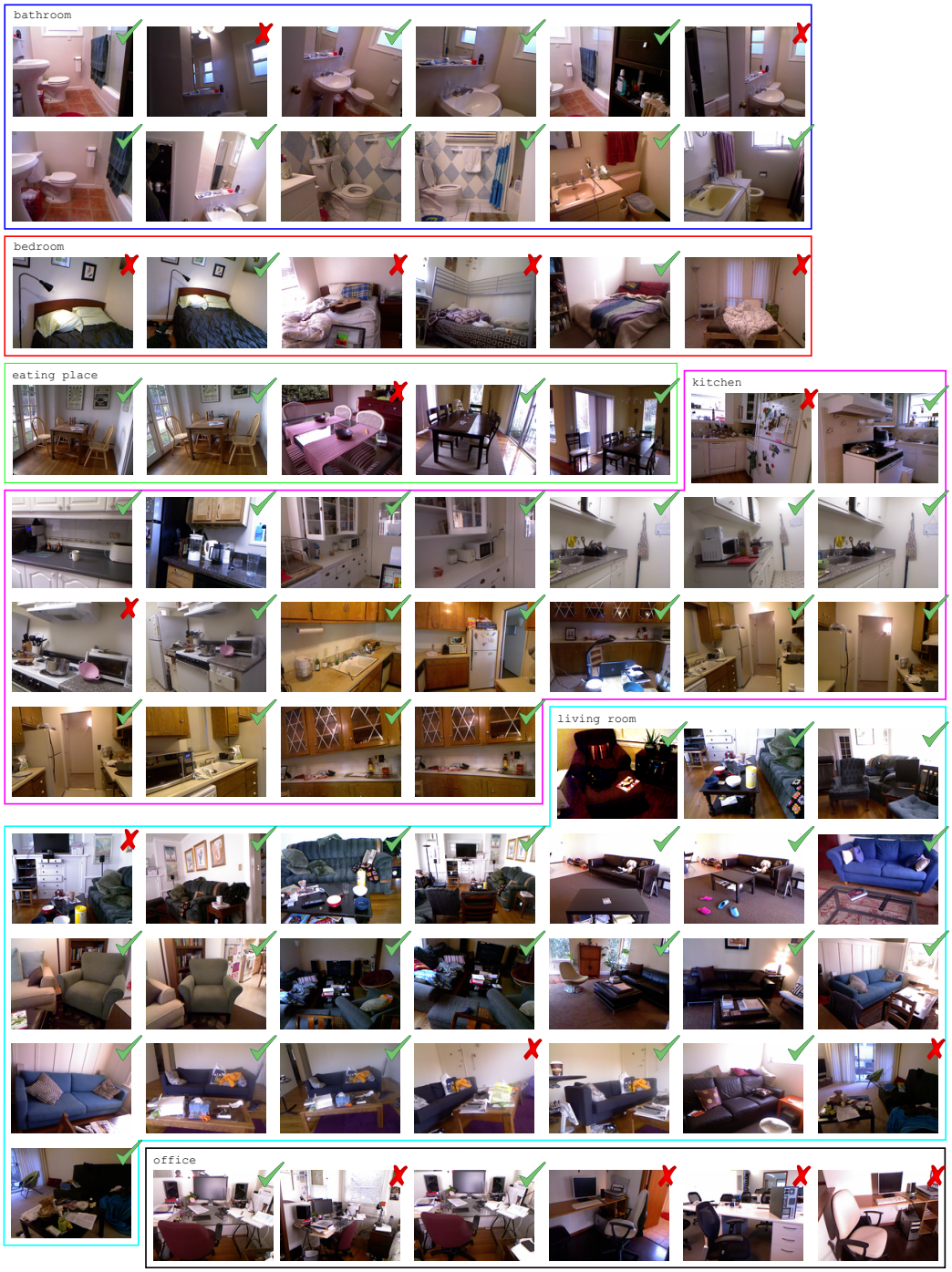
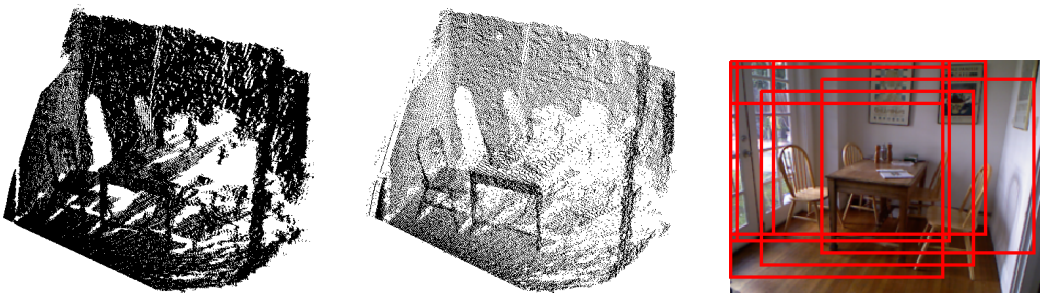


Figure 20: Part of the Berkeley 3D Object Dataset used for testing the performance of 3D spatial feature vector on Kinect data. Recordings showing a complete room were chosen. The check (correct) and the cross (failed) indicate whether this particular room was classified correctly in one of the test runs done for evaluation.



(a) raw Kinect point cloud (b) down-sampled point cloud (c) sub-window sampling

Figure 21: Preprocessing of scans from the Berkeley 3D Object Dataset.

the noise in the x, y, and z values (see Figure 21(b)). From this new point cloud sub-windows with 176×144 points are sampled from a grid with 5 pixel spacing. Example windows are shown in Figure 21(c). 187 training point clouds are extracted from one Kinect frame. For each point cloud planar patches are extracted as introduced in Section 3. The corresponding 3D spatial feature vector is computed from statistics on these patches as introduced in Section 4.1. The Gist feature is computed on the color image of the sub-window converted to a gray-scale image and scaled to 256×256 pixels.

7.3. Classification Results

Figure 22 shows the classification rates that are achievable on a subpart of the Berkeley 3D Object Dataset using Gist features $\vec{x}^{\text{Gist}}$, the 3D spatial feature vector $\vec{x}^{\text{3D}}$, and two combinations of this two features. One combination, $(\vec{x}^{\text{3D}}, \vec{x}^{\text{Gist}})$, results from summing up normalized responses of models trained on $\vec{x}^{\text{3D}}$ and $\vec{x}^{\text{Gist}}$ as introduced in Equation 15. For the second combination, $(\vec{x}^{\text{3D Boost}}, \vec{x}^{\text{Gist}})$, separate responses from the four patch characteristics are accumulated by a weighted sum as defined in Equation 19 and 20. The weights are estimated using an AdaBoost algorithm like described in Section 6.3. The classification rates are average values over 10 test sets.

The classification rates of the Gist feature and the 3D spatial feature are around 0.60 while the rate of the combined features is around 0.70. One important difference of the rate curves compared to the curves computed for the IKEA database (e.g. Figure 13) is that the fusion over consecutive windows does not result in such a large increase of the classification rate

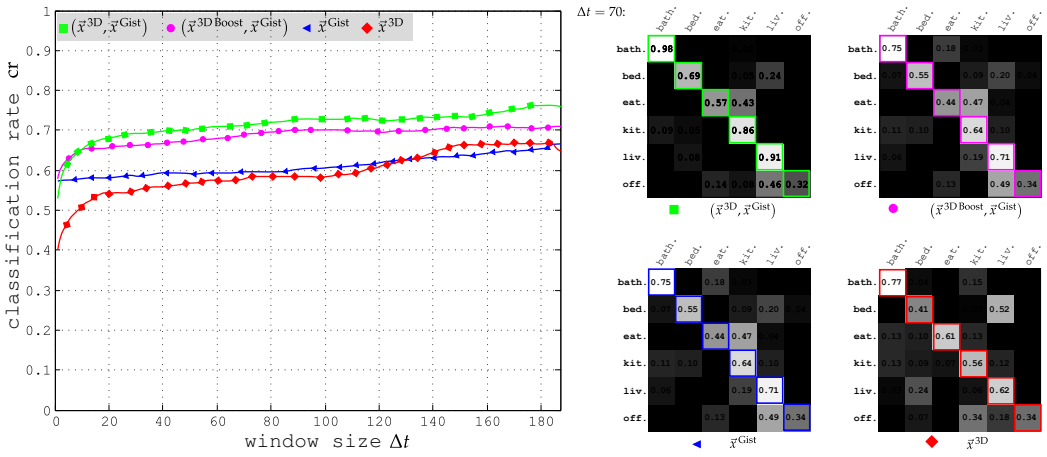


Figure 22: Overall classification rates and confusion matrices for the Berkeley 3D Object Dataset averaged over 10 test sets.

as it is the case for the IKEA database. The reason for this is that typically a smaller part of a room was captured. However, the captured views are all relevant. This is not the case in the IKEA database. This means, on the one hand, that in the IKEA database more information about a room is available but, on the other hand, more noise and irrelevant data (e.g., close-up views) needs to be handled. In contrast to that, the Berkeley 3D Object Dataset provides less information about a room but there are no irrelevant views. Therefore, quite good classification rates can be achieved for already small fusion windows (e.g., $\Delta t = 20$). But the rates do not improve much if the fusion window is enlarged because other views on a room, which might be helpful, are not available. Nevertheless, we can show that a wide variety of real world rooms (tidy and messy ones) can be classified successfully with our combination of 3D and Gist feature vector while neglecting object information completely.

In Figure 20 the check (correct) and the cross (failed) indicate whether this particular room was classified correctly in one of the test runs. We have run the training and testing 25 times to ensure that every room from the biggest class appears at least once in a test set. The categorization of one room is based on the fused responses of all 187 sub-windows. 77% of the rooms are classified correctly. Especially, most bathrooms, kitchens, and living rooms are recognized correctly. Our approach has problems with the two small classes, bedroom and office. This might be due to

the fact that not enough training data was available. To fit our purposes, the Berkeley 3D Object Dataset would need here an extension. A second reason might be that our approach to rely on global spatial information for indoor scene categorization is not best suited for this two room types as the same problem already appeared for the IKEA database. There, bedroom and office sequence were also misclassified quite often. Again one solution could be to integrate object information, especially those that are mandatory for a certain room type like office chair and desk are for offices. However, the challenge is that a good first guess of the room type can be necessary to detect such objects reliably.

8. Conclusion and Outlook

In this paper, we have presented a holistic scene model providing a rough scene impression of the robot's local environment. It relies on spatial information rather than on object information. 3D spatial features are defined encoding the 3D scene geometry given as a set of planar patches. Shape and size are a standard characteristic of patches, while angle and size ratio between two patches are a novel idea in such kind of patch analysis. We focus on analyzing the relation between patches for extracting information about the spatial layout. Evaluating the defined 25-dimensional 3D feature vector on the 3D IKEA database has shown a remarkable performance. It emphasizes the careful design of this new feature vector, especially if contrasted with the Gist feature computed on depth images. As the 3D layout of a 3D point cloud and the Gist information of the corresponding 2D image capture complementary aspects of a scene layout fusing both vectors leads to an error reduction of about 50% in recognizing the type of an indoor scene. Testing this holistic scene model on the 3D Berkeley Dataset shows that our approach also works for real-world data.

We have shown that indoor rooms, in particular bath rooms, eating places, kitchens, and living rooms, have room type specific arrangements which can be captured by the 3D feature vector. Failures in the mentioned rooms mostly arise from not-meaningful views. This not-meaningful views are either close-up views on tables, walls, and so on or show furniture that could found anywhere in a flat like a sideboard. We plan to introduce additional processing at two stages of our categorization system to handle these not-meaningful views. Close-up views could be detected during preprocessing by neglecting frames that are dominated by one pla-

nar surface. Further, the 3D spatial features of all rooms can be analyzed for the existence of large clusters. A large cluster would indicate a specific arrangement of planar patches that appears in all rooms independent from their type, e.g., a room corner. We assume that such views do not contribute to the categorization of room and should be rejected. Views of not-meaningful furniture could be handled on the classification stage of our system. Therefor, we would need to annotate such views in our database and to train a separate classifier. It is thinkable that good results can be achieved if, similar the setup of decision trees, the system first decides on the meaningfulness of a frame and estimates afterwards for the meaningful views the room type. Even further hierarchies could be interesting to investigate where room types are divided into functional subparts of a scene like "a wall with bookshelves" or "sideboard-like-furniture for placing things on it". Such scene subparts might be specific for a top level room type or could appear across different room types. Learning weights for the different subparts could help to learn which spatial layout is mandatory for a specific room type and which subparts are optional.

To further reduce misclassification of meaningful frames like frames from some bedroom and office sequences, we will concentrate in future on two approaches. The first one, will be defining further patch characteristics similar to those listed by Mozos [56] and others, e.g., analyzing the free space between the patches for capturing more details on the relations between patches. The second one, will concentrate on the question how to determine mandatory objects for a room type, how to detect them, and how to integrate them with classification results from global features. Our approach to accumulate classifier responses by a weighted sum could be in principle extended straight-forward by just introducing an additional summation term for object responses. However, the challenging question is how to determine which objects are mandatory for which room type, how to detect these objects, and how to assign suitable weights to the detector outputs. Insights from mechanisms of human cognition of rooms can give us the critical points to approach these problems.

9. Acknowledgements

This work is funded by the German Research Foundation within the Cooperative Research Center CRC673 "Alignment in Communication".

References

[1] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, M. Hebert, An Empirical Study of Context in Object Detection, in: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, IEEE, Miami, FL, USA, 2009, pp. 1271–1278.

[2] S. Kim, I. S. Kweon, Scene Interpretation: Unified Modeling of Visual Context by Particle-based Belief Propagation in Hierarchical Graphical Model, in: Proc. of the Asian Conf. on Computer Vision, Vol. 3852 of LNCS, Springer, Hyderabad, India, 2006, pp. 963–972.

[3] A. Torralba, Contextual Priming for Object Detection, Intl. Journal of Computer Vision 53 (2) (2003) 153–167.

[4] R. Epstein, N. Kanwisher, A Cortical Representation of the Local Visual Environment, Nature 392 (1998) 598–601.

[5] J. M. Henderson, C. L. Larson, D. C. Zhu, Full Scenes Produce More Activation than Close-up Scenes and Scene-diagnostic Objects in Parahippocampal and Retrosplenial Cortex: An fMRI Study, Brain and Cognition 66 (1) (2008) 40–49.

[6] D. B. Walther, E. Caddigan, L. Fei-Fei, D. M. Beck, Natural Scene Categories Revealed in Distributed Patterns of Activity in the Human Brain, Journal of Neuroscience 29 (34) (2009) 10573–10581.

[7] A. Torralba, K. P. Murphy, W. T. Freeman, M. A. Rubin, Context-based Vision System for Place and Object Recognition, in: Proc. of the Intl. Conf. on Computer Vision, Vol. 1, IEEE, Nice, France, 2003, pp. 273–280.

[8] A. Quattoni, A. Torralba, Recognizing Indoor Scenes, in: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, IEEE, Miami, FL, USA, 2009, pp. 413–420.

[9] A. Swadzba, S. Wachsmuth, Indoor Scene Classification using combined 3D and Gist Features, in: Proc. of the Asian Conf. on Computer Vision, Vol. 6493 of LNCS, Springer, Queenstown, New Zealand, 2010, pp. 725–739.

[10] S. Lee, D. Jang, E. Kim, S. Hong, J. Han, A Real-Time 3D Workspace Modeling with Stereo Camera, in: Proc. of the Intl. Conf. on Intelligent Robots and Systems, IEEE, Edmonton, Canada, 2005, pp. 2140–2147.

[11] S. X. Yu, H. Zhang, J. Malik, Inferring Spatial Layout from A Single Image via Depth-ordered Grouping, in: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops, IEEE, Anchorage, AK, USA, 2008.

[12] H. Zender, O. Mozos, P. Jensfelt, G.-J. M. Kruijff, W. Burgard, Conceptual Spatial Representations for Indoor Mobile Robots, RAS 56 (6) (2008) 493–502.

[13] P. Beeson, M. MacMahon, J. Modayil, A. Murarka, B. Kuipers, B. Stankiewicz, Integrating Multiple Representations of Spatial Knowledge for Mapping, Navigation, and Communication, in: Proc. of the Symposium on Interaction Challenges for Intelligent Assistants, AAAI Spring Symposium Series, 2007, AAAI Technical Report SS-07-04.

[14] P. Buschka, A. Saffiotti, A Virtual Sensor for Room Detection, in: Proc. of the Intl. Conf. on Intelligent Robots and Systems, Vol. 1, IEEE, Lausanne, Switzerland, 2002, pp. 637–642.

[15] A. Pronobis, Ó. M. Mozos, B. Caputo, P. Jensfelt, Multi-modal Semantic Place Classification, Intl. Journal of Robotics Research 29 (2–3) (2010) 298–320.

[16] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, P. Jensfelt, H. I. Christensen, Towards Robust Place Recognition for Robot Localization, in: Proc. of the Intl. Conf. on Robotics and Automation, IEEE, Pasadena, CA, USA, 2008, pp. 530–537.

[17] E. Topp, H. Christensen, Detecting Structural Ambiguities and Transitions during a Guided Tour, in: Proc. of the Intl. Conf. on Robotics and Automation, IEEE, Pasadena, CA, USA, 2008, pp. 2564–2570.

[18] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernández-Madrigal, J. González, Multi-Hierarchical Semantic Maps for Mobile Robotics, in: Proc. of the Intl. Conf. on Intelligent Robots and Systems, IEEE, Edmonton, Canada, 2005, pp. 3492–3497.

[19] P. Espinace, T. Kollar, A. Soto, N. Roy, Indoor scene recognition through object detection, in: Proc. of the Intl. Conf. on Robotics and Automation, IEEE, 2010, pp. 1406–1413.

[20] P. Viswanathan, D. Meger, T. Southey, J. J. Little, A. Mackworth, Automated Spatial-Semantic Modeling with Applications to Place Labeling and Informed Search, in: Proc. of the Canadian Conf. on Computer and Robot Vision, IEEE, Kelowna, Canada, 2009, pp. 284–291.

[21] A. Ranganathan, F. Dellaert, Semantic Modeling of Places using Objects, in: Proc. of the Robotics: Science and Systems, Atlanta, GA, USA, 2007.

[22] S. Vasudevan, S. Gächter, V. Nguyen, R. Siegwart, Cognitive Maps for Mobile Robots – An Object based Approach, RAS 55 (5) (2007) 359–371, from Sensors to Human Spatial Concepts.

[23] A. Oliva, A. Torralba, Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope, Intl. Journal of Computer Vision 42 (3) (2001) 145–175.

[24] A. Bosch, A. Zisserman, X. Muñoz, Scene Classification using a Hybrid Generative/Discriminative Approach, Trans. on PAMI 30 (4) (2008) 712–727.

[25] S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, in: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, IEEE, New York, NY, USA, 2006, pp. 2169–2178.

[26] I. Posner, D. Schröter, P. M. Newman, Using Scene Similarity for Place Labelling, in: Proc. of the Intl. Symposium on Experimental Robotics, Vol. 39 of Springer Tracts in Advanced Robotics, Springer, Rio de Janeiro, Brazil, 2006, pp. 85–98.

[27] L. Fei-Fei, P. Perona, A Bayesian Hierarchical Model for Learning Natural Scene Categories, in: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, IEEE, San Diego, CA, USA, 2005, pp. 524–531.

[28] F. Pirri, Indoor Environment Classification and Perceptual Matching, in: Proc. of the Intl. Conf. on Knowledge Representation, AAAI Press, Whistler, Canada, 2004, pp. 73–84.

[29] J. Vogel, B. Schiele, A Semantic Typicality Measure for Natural Scene Categorization, in: LNCS: Pattern Recognition – DAGM Symposium, Springer, Tübingen, Germany, 2004, pp. 195–203.

[30] M. A. Fischler, R. C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM 24 (6) (1981) 381–395.

[31] A. Nüchter, J. Hertzberg, Towards Semantic Maps for Mobile Robots, RAS 56 (11) (2008) 915–926.

[32] G. Schwarz, Estimating the dimension of a model, Annals of Statistics 6 (1978) 461–464.

[33] R. Triebel, W. Burgard, F. Dellaert, Using Hierarchical EM to Extract Planes from 3D Range Scans, in: Proc. of the Intl. Conf. on Robotics and Automation, IEEE, Barcelona, Spain, 2005, pp. 4437–4442.

[34] H. Andreasson, R. Triebel, W. Burgard, Improving Plane Extraction from 3D Data by Fusing Laser Data and Vision, in: Proc. of the Intl. Conf. on Intelligent Robots and Systems, IEEE, Edmonton, Canada, 2005, pp. 2656–2661.

[35] P. Dorninger, C. Nothegger, 3D Segmentation of Unstructured Point Clouds for Building Modelling, in: Intl. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36, Munich, Germany, 2007, pp. 191–196.

[36] J. Hois, M. Wünstel, J. A. Bateman, T. Röfer, Dialog-based 3D-Image Recognition using a Domain Ontology, in: Spatial Cognition, Vol. 4287 of LNCS, Springer, Freiburg, Germany, 2008, pp. 107–126.

[37] T. Rabbani, F. A. van den Heuvel, G. Vosselman, Segmentation of Point Clouds using Smoothness Constraints, in: Intl. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36, Dresden, Germany, 2006, pp. 248–253.

[38] P. Fua, From Multiple Stereo Views to Multiple 3D Surfaces, Intl. Journal of Computer Vision 24 (1) (1997) 19–35.

[39] I. Stamos, P. K. Allen, Geometry and Texture Recovery of Scenes of Large Scale, Computer Vision and Image Understanding 88 (2) (2002) 94–118.

[40] A. Swadzba, S. Wachsmuth, Categorizing Perceptions of Indoor Rooms using 3D Features, in: LNCS: Structural, Syntactic, and Statistical Pattern Recognition, Vol. 5342, Springer, Orlando, FL, USA, 2008, pp. 744–754.

[41] A. Barla, F. Odone, A. Verri, Histogram Intersection Kernel for Image Classification, in: Proc. of the Intl. Conf. on Image Processing, 2003, pp. 14–18.

[42] J. B. Kruskal, M. Wish, Multidimensional Scaling, no. 07-011 in Sage University Paper Series on Quantitative Application in the Social Sciences, Sage Publications, Beverly Hills and London, 1978.

[43] L. I. Kuncheva, Combining Pattern Classifiers, John Wiley & Sons, 2004.

[44] L. I. Kuncheva, Theoretical Study on Six Classifier Fusion Strategies, Trans. on PAMI 24 (2) (2002) 281–286.

[45] J. Kittler, M. Hatef, R. P. W. Duin, J. Matas, On Combining Classifiers, Trans. on PAMI 20 (3) (1998) 226–239.

[46] Y. Freund, R. E. Schapire, A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.

[47] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.

[48] T. Joachims, Making Large-scale SVM Learning Practical, in: B. Schölkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods - Support Vector Learning, MIT-Press, 1999.

[49] A. Wendel, A. Pinz, Scene Categorization from Tiny Images, in: Proc. of the Workshop of the Austrian Association for Pattern Recognition, 2007.

[50] B. C. Russell, A. Torralba, Building a database of 3d scenes from user annotations, in: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2009.

[51] D. Lee, M. Hebert, T. Kanade, Geometric Reasoning for Single Image Structure Recovery, in: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, IEEE, Miami, FL, USA, 2009, pp. 2136–2143.

[52] V. Hedau, D. Hoiem, D. Forsyth, Recovering the Spatial Layout of Cluttered Rooms, in: Proc. of the Intl. Conf. on Computer Vision, IEEE, Kyoto, Japan, 2009, pp. 1849–1856.

[53] A. Saxena, M. Sun, A. Y. Ng, Make3D: Learning 3D Scene Structure from a Single Still Image, Trans. on PAMI 31 (5) (2008) 824–840.

[54] D. Hoiem, A. A. Efros, M. Hebert, Recovering Surface Layout from an Image, Intl. Journal of Computer Vision 75 (1) (2007) 151–172.

[55] A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, T. Darrell, A category-level 3-d object dataset: Putting the kinect to work, in: Intl. Conf. on Computer Vision Workshop on Consumer Depth Cameras for Computer Vision, 2011.

[56] Ó. M. Mozos, C. Stachniss, W. Burgard, Supervised Learning of Places from Range Data using AdaBoost, in: Proc. of the Intl. Conf. on Robotics and Automation, IEEE, Barcelona, Spain, 2005, pp. 1730–1735.