Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Using Surfaces and Surface Relations in an Early Cognitive Vision System

Dirk Kraft · Wail Mustafa · Mila Popović · Jeppe Jessen · Anders Glent Buch · Thiusius Rajeeth Savarimuthu · Nicolas Pugeault · Norbert Krüger

Received: date / Accepted: date

Abstract We present a deep hierarchical visual system with two parallel hierarchies for edge and surface information. In the two hierarchies, complementary visual information is represented on different levels of granularity together with the associated uncertainties and confidences. At all levels geometric and appearance information is coded explicitly in 2D and 3D allowing to access this information separately and to link between the different levels. We demonstrate the advantages of such hierarchies in three applications covering grasping, view-point independent object representation, and pose estimation.

Keywords cognitive vision · deep hierarchies · surface representation · surface relations

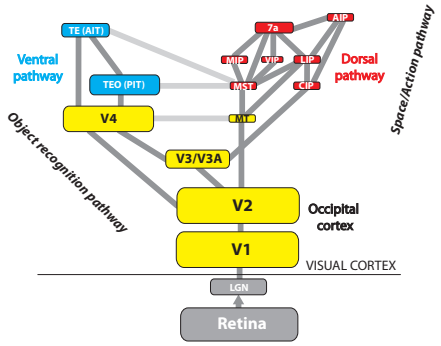


Fig. 1 Schematic hierarchy of the primate's visual system. Note that the size of each area drawn is proportional to the size of the area in the primate's brain.

1 Introduction

In this work, we present a deep hierarchical computer vision system, for which functional design decisions were made in analogy to the human visual system. The primate's visual cortex occupies approximately 50% of the neocortex [14], characterizing vision as the primary sense of humans. The visual cortex is constituted by a number of interconnected areas forming an example of a "deep hierarchy" (see Fig. 1) with more than seven levels. There is evidence that cognitive abilities and the concept of deep hierarchies are linked to each other [81]. Interestingly, it is widely accepted that the first levels of the visual hierarchy (V1–V4, MT) indicated in yellow in Fig. 1, which occupy approximately 70% of the visual cortex [14], provide a generic and largely task independent scene representation [36]. These areas feed into the ventral and dorsal pathway which have also been named 'what' and 'where' pathways since the ventral pathway has been associated to recognition and categorization while the dorsal pathway has been associated to spatial perception and action [56].

From this, we can infer as the general picture of visual processing in the human cortex (1) a deep hierarchical structure in which (2) the largest part of the processing is devoted to the extraction of a generic scene representation. The work in this paper is concerned with the development of an artificial visual system (which we have called earlier 'Early Cognitive Vision' (ECV) [1, 67]) following these two principles. Functional design choices of the ECV system are motivated by the primate's visual system architecture and it has been used in applications for robotic and computer vision tasks under real-time constraints.

Dirk Kraft, Wail Mustafa, Mila Popović, Jeppe Jessen, Anders Glent Buch, Thiusius Rajeeth Savarimuthu and Norbert Krüger
Cognitive and Applied Robotics Group
The Mærsk Mc-Kinney Møller Institute
University of Southern Denmark
Campusvej 55, 5230 Odense M, Denmark
E-mail: kraft@mmmi.sdu.dk

Nicolas Pugeault
College of Engineering, Mathematics and Physical Sciences
University of Exeter
Exeter, EX4 4QF, United Kingdom

Conceptual advantages of deep hierarchies over 'flat architectures' have been discussed in depth in, e.g., the context of matching [17, 22, 54]. Tsotsos [82] spelled out the NP completeness of unbounded visual search and gives arguments that hierarchical architectures are a promising way to approach that problem.

We provide results on three different applications, namely object grasping, object recognition and pose estimation, to exemplify the advantages using a hierarchal representation. In particular, we exemplify how different levels of the hierarchy can be used depending on the actual task. For example, in the context of pose estimation (see Sect. 5.3), it proves to be advantageous to operate with spatially extended entities of high complexity to reduce the correspondence problem at initial stages of the algorithm. In contrast, for grasping unknown objects it can be advantageous to have access to more local entities representing contact points (see Sect. 5.1). As a consequence, a good accessibility and transfer between levels is important. In our system, this is realized by ensuring that each entity at each level of the hierarchy can be accessed on request, including associated information on how it is linked to lower and higher level entities it is derived from or embedded in.

A second important issue is the representation of uncertainty at each level of the hierarchy, and the initialization of processes which reduce this uncertainty. In our framework, uncertainty of local features can be computed analytically or by means of a Monte Carlo method. From that, uncertainties for higher level features are derived. The uncertainty on local levels has been used for temporal disambiguation (see [33]), and as well as to improve estimation processes of higher level attributes, (see, e.g., [1]). Uncertainties of higher level entities can be used in selection processes (as in, e.g., pose estimation, see Sect. 5.3), preferring the use of reliable and avoiding the use of uncertain entities.

At lower levels of visual information, it is common to distinguish visual information of different kinds. In computer vision, extraction processes for structures such as edges [10] and patchlets (or texlets) [57, 60] have been discussed. They represent edge and surface information which require a different kind of parameterization. For example, a step edge divides an area into two parts corresponding to different color values or texture structure. In 3D it can represent a surface edge, an orientation discontinuity or a depth discontinuity [35]. In contrast, homogeneously colored or structured image areas correspond with high likelihood to smooth 3D patches [35] and therefore homogeneous image areas can be sufficiently described by one color vector. Hence, both kinds of structure require different kinds of hierarchical organization (e.g., local edges can be embedded in more global contours while local texlets can be embedded in surfaces). These higher level entities then require other kind of parameterization descriptors (e.g., for surfaces the two principle curvatures can be computed reliably which is much harder at a lower level of processing due to the limited spatial support).

Moreover, besides providing descriptors of the visual entities at the different levels of the hierarchy, our ECV systems provides *relational information* between pairs of such entities. Examples for such relations are relative angles, distances in position or position and orientation, and similarities in the appearance of the two entities. Conceptually, our relations are similar to the ones introduced in [85], which are computed globally between surface patches (with respect to the object context). Our ECV system however provides a different set of relations than in [85] and also, depending on the application, these relations can be used locally — in this case, relations resemble the shape context descriptors [3]. Moreover, in addition to surface features, the ECV provides relations of edge features.

This paper describes our work on the extension of the hierarchical Early Cognitive Vision (ECV) system [1, 67] (see Fig. 2, right stream)—which previously provided a visual hierarchy in the edge domain—by a parallel hierarchy in the surface domain (see Fig. 2, left stream). Our ECV system has been used in a large number of vision applications, (see, e.g., [12]), as well as in robotics, (see [40, 43, 66]). This representation, although rather elaborated, can operate in real-time through the use of GPUs [32]. The ECV representation provides a hierarchy of descriptors covering (and separating) geometric as well as appearance information in 2D and 3D. Higher levels of the hierarchy group local information into entities with larger spatial extend (see Fig. 2). The descriptors are embedded in the spatial temporal context, allowing for disambiguation, as well as semantic reasoning (see, e.g., [1, 33]).

The main contribution of this paper is to first describe a richer visual hierarchy in which a hierarchy (compared to [1]) in the edge domain is supplemented by a hierarchy in the surface domain and then to discuss aspects of this visual hierarchy for the application. Individual work making use of certain aspects of this hierarchical representation for specific applications has been published already before (see, e.g., [41, 43, 66]). In this paper however, we focus on the actual visual representation that is being applied. For that purpose, we give a much more detailed description of many technical aspects compared to the application oriented papers (for example by describing how the representation can also be derived by means of Kinect cameras) and we also show, how different aspects of the hierarchy play different roles in different applications. In addition, we arrive at a coherent formalism including in particular the uncertainties associated to visual entities. Furthermore, we relate our work to more common flat visual representations as well as to other deep hierarchical visual representations developed in the last decades and also to the recent revival of deep hierar-

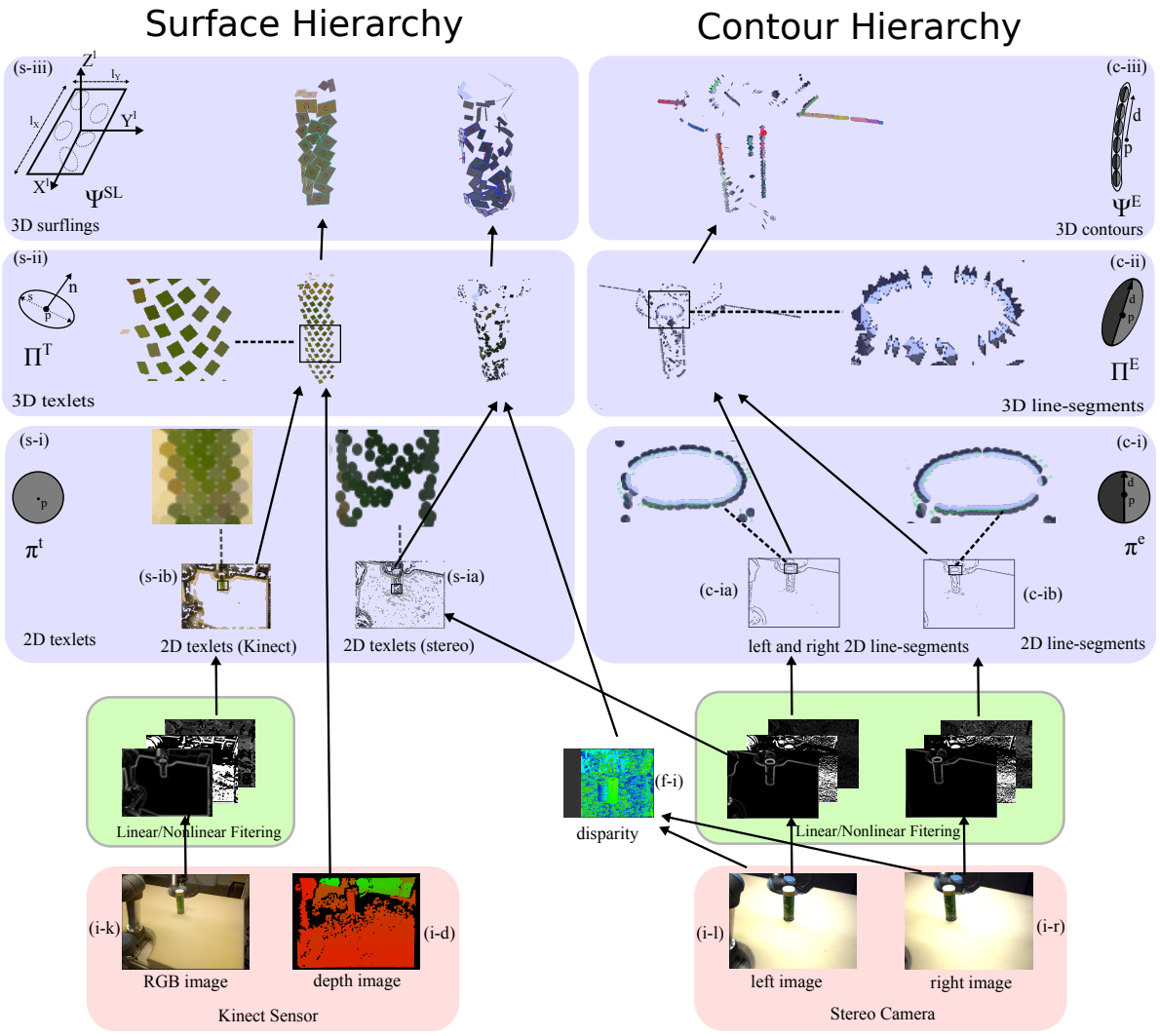


Fig. 2 The hierarchical representation of edge and texture information in the ECV system. (i-l,i-r) An example stereo image pair while (i-k,i-d) are the RGB and the depth images from Kinect . (c-i) 2D line segments for the left ((c-ia)) and the right ((c-ib)) image. (c-ii) 3D line segments. (c-iii) 3D contours. (s-i) 2D texlets for the left image ((s-ia)) and from Kinect ((s-ib)). (f-i) disparity image from stereo images. (s-ii) 3D texlets. (s-iii) 3D surflings. This figure is best viewed in color.

chical nets in the machine learning community. The link to the human visual system is also made explicit.

The paper is structured as follows: In Sect. 2, we give an overview of the work on deep hierarchies in computer vision. We then first briefly sketch the edge hierarchy (introduced in [1, 67]), in Sect. 3. A full description of the technical realization of the novel surface hierarchy is presented in Sect. 4. In Sect. 5, we describe three representative applications: the grasping of unknown objects, view point invariant object representations, and pose estimation to support our claims on the advantageous of deep hierarchies and the complementariness of the edge and surface hierarchy.

2 State of the art

In our discussion of the related state of the art we first address the predominate use of flat architectures in computer vision in Sect. 2.1 before we discuss other deep hierarchical approaches in Sect. 2.2. In Sect. 2.3, we give a brief discussion on other biologically motivated vision systems.

2.1 Flat vs. deep architectures

The major body of work in ‘mainstream computer vision’ has focused on flat hierarchies. After defining rather simple, and usually task specific feature representations, some kind of classifiers are learned as, e.g., in bag of words approaches (see, e.g., [11, 63, 87]) or in many industrial applications

(see, e.g., [86]). Such systems can lead to rather impressive results for specific tasks, but—as discussed above—face the inherent limitations of flat architectures. In this context, Nicolas Pinto et al. [64, 65] compared state-of-the-art feature descriptors, in a scenario where objects where shown from widely varying viewpoints and with artificially manipulated background (to remove contextual information), and showed a significant degradation of performance compared to common recognition datasets, hinting at a large dependence of these algorithms on context and canonical viewpoints.

The limitation of unstructured bag-of-words models has lead to number of improvements in order to include local structure information [23, 50, 59, 68, 75]. For example Savarese at al. [75] used correlograms based on local kernels to encode spatial organization of visual words for object recognition, whereas Lazebnik et al. [50] used spatial pyramids of visual words to recognize visual scenes categories. Using data mining approaches, Quack et al. [68] used spatial configurations of visual words for object recognition, and Gilbert et al. [23] used hierarchically mined discriminative feature configurations for action recognition. These systems all rely heavily on the extraction of discriminative codebooks from engineered features, and therefore lead to high performance at the cost of being completely task-specific.

The system we present here falls clearly in the category of a deep hierarchy with more than four levels starting with linear and non-linear pixel wise filtering stages (see Fig. 2(f-i))[1], then computing local 2D information (see Figs. 2(c-i) and 2(s-i)) and 3D information (see Figs. 2(c-ii) and 2(s-ii)) which are then embedded in semi-local and more global visual descriptors (see Fig. 2(c-iii) and 2(s-iii)).

2.2 Hierarchical computer vision systems

In a number of works in computer vision, the potential of hierarchical structures were successfully exploited [12, 17, 18, 20, 27, 30, 61, 70–72, 84]. Such approaches can be distinguished between designed [20, 30] and learned [17, 18, 71, 72] and hybrid models [12]. Early examples of mainly designed hierarchical models are Fukushima's Neocognitron [20] and the model of Hummel and Biederman [30]. Fukushima's work [20] has been applied to the problem of character recognition while Hummel and Biederman's work [30] has been applied to object recognition from line drawings. A characteristic of these systems is also that the information at each level of the processing is explicit, i.e., is parameterized and provides a semantic description of the information at a certain hierarchical level.

[1] Note that multiple of these early stages of processing are collapsed into one level in Fig. 2 and are in more detail described in, e.g., [67]

The ECV system we present in this paper is also fully designed but provides much richer information than the systems in [20, 30] as well others described below. In particular, we provide hierarchies providing 2D and 3D as well as geometric and appearance information both in the edge and surface domains. However, this richness comes with the price of a mainly designed system (see also Sect. 2.3). The ECV system has also been applied to a much larger variety of tasks (covering different vision as well as robotic tasks). It reflects the progress that has been made on feature processing in the last decades which has been integrated into the different processing stages of the system. Moreover based on a hybrid architecture utilizing coarse and fine grained parallel computing, the system computes lower stages of the hierarchy with up to 20 Hz, and the complete hierarchy at a speed sufficient for robot manipulation tasks.

As Dickinson and others (see, e.g., [13, 46]) pointed out, the focus of the computer vision community has—after realizing severe problems in approaching computer vision in a hierarchical paradigm (as in particular suggested by Marr [53])—shifted from explicitly designed hierarchies to the design of more efficient low level feature descriptors and classification schemes based on those leading to flat architectures. We argued in [46] that at the time Marr published his approach towards computer vision, two main reasons made his ideas unfeasible. First, there was a severe lack of knowledge on low-level processes such as, e.g., edge detection, stereo and optic flow processing. Secondly, the computational resources required for designing such complex hierarchies were not available at that time.

In [13], Dickinson gives an overview of the development of computer vision approaches pointing to the 'representational gap' between sensory information and the categorical models applied. Dickinson argues that while in the 1970's there existed a large gap between the degree of abstraction used in the applied models and the features extracted from the input image, this gap has been reduced in the 1980's and 1990's by reducing the complexity of the categorical models and to a certain extend also by progress of feature extraction algorithms. Flat architectures—as predominant today (as discussed in Sect. 2.1)—are an example of a very low degree of abstraction of the applied models. However, Dickinson argues that this has also lead to a drift of the problem statement, from categorization to the less challenging problem of exemplar recovery—and that categorization requires the formation of higher levels of visual abstraction, effectively calling for the development of deep hierarchies. Explicitly designed hierarchical models, as the one described in this paper, are a way to bridge the 'representational gap' by allowing learning algorithms to address a particular problem either at lower or at higher levels of abstraction depending on the actual problem.

Early examples of fully learned deep hierarchical systems are classical neural network algorithms such as the perceptron [71] or backpropagation [72] (note that the structure of such systems is usually designed and fixed). However, realizing deep versions of such systems (i.e., introducing many layers) has remained a challenge for decades because of the large amount of meta parameters [4] and limitations of the existing learning algorithms and in general flat structures have been shown to be more successful in many applications.

Recently, new successes have been achieved in learning deep hierarchical structures [4]. Notably, Hinton proposed a generic method for incremental and unsupervised learning of layers of simple processing units to form deep hierarchies, coined Deep Belief Networks (DBN) [26]. The hierarchical inference is learned using Restricted Boltzmann Machines (RBM), and a discriminative top layer can be added to solve typical recognition tasks. The approach has shown success for image [5, 26] and video recognition [79]. Another approach to Deep Learning is the convolutional networks proposed by LeCun et al. [51], that is based on sparsely connected layers processing alternatively convolution operations and max-pooling. Best performance is obtained by unsupervised, layer-wise pretraining and followed by a back-propagation refinement of the weights, and have recently provided leading performances in a variety of datasets, including traffic sign recognition [77], and visual recognition [38], notably on the difficult ImageNet Large Scale Visual Recognition Challenge dataset[76]. Interestingly, it was recently shown that the hierarchy learnt on the ImageNet dataset could be used successfully on other datasets [69], supporting the argument that deep hierarchy can learn generic abstractions from data.

Bengio argues that the ideal depth of the ideal architecture is dependent on the problem at hand [4]. Moreover, although the hierarchies are in principle task independent, the unsupervised nature of the hierarchical learning implies that a very large amount of training data is required for creating truly generic hierarchies. The amount of training data needed is increasing with the depth of the hierarchy to be learned. Such a computationally intense training procedure can be alleviated by introducing a considerable amount of bias by means of appropriate design decisions in the visual hierarchies.

In general, it is a worth discussing whether such hierarchies can be fully learned. Some neurophysiological evidence draws a picture of a well balanced amount of prior structure in the human visual system that is required to bootstrap visual learning (for a more extensive discussion, see, e.g., [39, 47]). One way of reducing the complexity of this learning problem is to learn different levels one after the other as, e.g., done in the work of Leonardis et al. [17, 18] on compositional architectures for 2D edge structures. Their system also allows for a certain degree of explicitness at the different hierarchical levels by attaching semantics to learned structures. As said above, our system is completely designed but provides richer information than, e.g., [17, 18] by covering not only 2D edge information but also 3D and surface information.

A hybrid approach, in which a skeleton for the different levels of the hierarchy is provided which is then fine-tuned by learning might be a way to avoid the overwhelming complexity of deep hierarchical structures but at the same time might provide a sufficient flexibility to avoid shortcomings of sub-optimal decisions of the designer. An example of such a hybrid hierarchical system is [12] which uses early stages of the ECV system described in this paper—more specifically the 3D edge primitives (see Fig. 2(c-ii))—as basis for a system that learns a hierarchy for pose estimation in a probabilistic framework. In this case, a graphical network is learnt associating object's poses to 3D configurations of visual features, from a set of examples. In this case, the use of 3D edge primitives provide a higher level of abstraction facilitating the learning problem.

2.3 Biologically motivated vision systems

Interactions between the disciplines of "biological vision" and "computer vision" have varied in intensity throughout the course of computer vision history and have in some way reflected the changing research focuses of the machine vision community (see, e.g., [13]). Without any doubt, the groundbreaking work of Hubel and Wiesel [28] gave a significant impulse to the computer vision community via Marr's work [53] on building visual hierarchies analogous to the primate visual system.

With the reorientation of mainstream computer vision from trying to solve general vision problems to focusing more on specific methods related to specific tasks, biological guidance was most often limited to individual functional modules such as the choice of Gabor wavelets. Also the gap between biological modeling and requirements of applied computer vision systems on computational efficiency have been often too large to make biological motivated systems competitive in terms of performance.

In the last decade, a number of serious attempts have been made to bridge between biological and computer vision designing systems with competitive performance. For example, Serre et al. [78] proposed a biologically inspired model of the early stages of visual processing in humans and showed: 1) competitive recognition performance compared to state-of-the-art engineered features, and 2) interestingly, their system showed a dependence on a larger number of features compared to typical Bag of Words codebooks. Also the work by Leonardis et al. (see also Sect. 2.2) arrives

at hierarchies in which individual levels carry features with biological plausibility.

Certain design choices in the ECV system described in this paper are motivated by current knowledge about the human visual system. This concerns stages of early visual processing with Gabor like filters, local descriptors which have some analogy to the concept of 'hypercolumns' [29] as well as the parameterization of information at different levels (for a more detailed discussion we refer to, e.g., [48]). However, the aim of our work is not to arrive at a detailed model of the human (or primate) visual system but to develop a system that can be used successfully for a variety of vision and vision based robotics applications.

In summary, our work presents a rich and designed visual hierarchy for which design choices are motivated by functional insight into the visual processing of primates. The system shows a sufficiently fast processing as required for applications in vision and robotic applications. The ECV system described in this paper has been applied in a variety of contexts covering vision tasks such as pose estimation [12], object learning [44] and tracking [40] as well as robotics tasks such as grasping [41, 66] as well as grounding of visual information in cognitive systems [43].

3 Hierarchy in the edge domain

In this section, we briefly sketch the edge domain hierarchy and introduce its basic notation and an intuitive understanding as given in Fig. 2. Such an understanding is required to present the applications described in Sect. 5, which show the complementary power of both domains, by making use of both kinds of hierarchies within the same or for different tasks. Sect. 3.1 describes the local 2D and 3D edge descriptors, while in Sect. 3.2, the contour level is described. For this paper, we do not give any details on the extraction of the entities in the edge domain hierarchy since this has already been described in detail in [1, 67], but we provide an intuitive understanding based on Fig. 2 and the associated notations which are analogous to the notation in the surface domain hierarchy. For any details about the computation of the entities in the edge hierarchy, we refer to [1, 67].

3.1 Local edge descriptors in 2D and 3D and their relations

A description of local 2D edge features is given in Sect. 3.1.1, while Sect. 3.1.2 presents local 3D edge features and briefly lists their relations.

3.1.1 2D edge primitive π^e

2D edge primitives π^e represent short line segments extracted from a single image. Their formalization contains geometric information G with associated covariance Σ_G, appearance information A as well as a confidence B. Formally, we have

$$\pi^e = (G^e, A^e, \Sigma_G^e, B^e).$$

Edge Primitives (see Fig. 2(c-i)) have an orientation that can be computed reliably. Their position can only be determined locally up to a one-dimensional manifold because of the aperture problem. Hence, the geometric information is described as

$$G^e = (\mathbf{p}^e, \mathbf{d}^e) = (x, y, d_1, d_2),$$

where (d_1, d_2) describe the direction vector with $||\mathbf{d}^e|| = 1$. The covariance related to this geometric informations is expressed as $\Sigma_G^e \in \mathbb{R}^4 \times \mathbb{R}^4$.

The appearance information of the line segment consists of two color triplets defining the color on the left ($\mathbf{c}^l$) and right ($\mathbf{c}^r$) side of the edge (and possibly one on the edge ($\mathbf{c}^m$) for a line structure) with $\mathbf{c}^i \in \mathbb{R}^3, i \in \{l, m, r\}$, and a phase ω defining the greyscale transition [24]. Hence,

$$A^e = (\omega, \mathbf{c}^l, \mathbf{c}^m, \mathbf{c}^r).$$

The local phase $\omega \in [-\pi, \pi)$ describes the structure of the appearance information. It can be determined from local filter responses [16, 42] allows for the differentiation between step edges (e.g., transition from dark to bright) and line structures (e.g., bright line on darker background). This information is taken into consideration when extracting and encoding the color information [67].

The confidence B^e indicates the likelihood that the local image structure corresponds to an edge (for details, see [15]).

3.1.2 3D edge primitive Π^E

The 3D edge descriptor (see Fig. 2(c-ii)) is derived from corresponding 2D edge primitives in the left and right image. It therefore represents a line-segment structure in 3D space. The following representation is used:

$$\Pi^E = (G^E, A^E, \Sigma_G^E, B^E).$$

It contains the geometric attributes $G^E = (\mathbf{p}^E, \mathbf{d}^E)$ with position $\mathbf{p}^E = (x, y, z)$ and a direction vector $\mathbf{d}^E = (d_1, d_2, d_3)$ ($||\mathbf{d}^E|| = 1$) as well as the geometry covariance $\Sigma_G^E \in \mathbb{R}^6 \times \mathbb{R}^6$ expressing the uncertainty of the geometric information.

The appearance attributes $A^E = (\omega, \mathbf{c}^l, \mathbf{c}^m, \mathbf{c}^r)$ cover the phase ω as well as three color values. Both are defined analogous to the 2D case (see Sect. 3.1.1). The actual values are found by combining the values of the corresponding 2D edge descriptors. Therefore, the middle color is again only

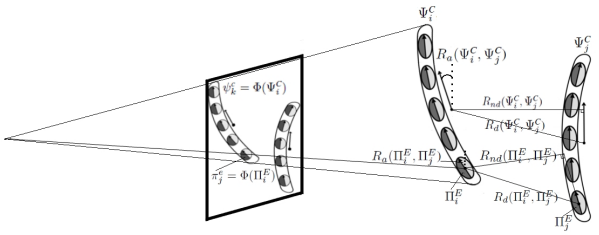


Fig. 3 Relations between 3D edge primitives (Π_i^E and Π_j^E) and relations between contour features (Ψ_i^C and Ψ_j^C). For both feature types, the following relations are shown: Euclidean distance $R_d(\Pi_i^E, \Pi_j^E)/R_d(\Psi_i^C, \Psi_j^C)$, angle $R_a(\Pi_i^E, \Pi_j^E)/R_a(\Psi_i^C, \Psi_j^C)$ and normal distance $R_{nd}(\Pi_i^E, \Pi_j^E)/R_{nd}(\Psi_i^C, \Psi_j^C)$. The correspondence relation ($\Phi(\Pi_i^E) = \pi_j^e$, $\Phi(\Psi_i^C) = \psi_k^c$) which is used to link between the corresponding features in 2D and 3D is also shown.

defined when the phase indicates a line structure. In addition, a confidence $B^E \in [0, 1]$ is associated to each 3D edge primitive representing the system's belief that the 3D entity is constructed from a correct stereo correspondence. This is set according to the matching score achieved in stereo processing (for details, see [67]).

On these entities, a number of second order relations are defined, namely Euclidean distance $R_d(\Pi_i^E, \Pi_j^E) \in \mathbb{R}$, angle $R_a(\Pi_i^E, \Pi_j^E) \in [0, \pi]$, collinearity $R_l(\Pi_i^E, \Pi_j^E) \in \mathbb{R}$, coplanarity $R_p(\Pi_i^E, \Pi_j^E) \in \mathbb{R}$, normal distance $R_{nd}(\Pi_i^E, \Pi_j^E) \in \mathbb{R}$ and co-colority $R_c(\Pi_i^E, \Pi_j^E) \in \mathbb{R}$. The relations Euclidean distance, angle and normal distance are depicted in Fig. 3. See [1] for more detailed information regarding these relations.

The **correspondence** relation between 2D and 3D edge primitives is expressed as $\Phi(\Pi_i^E) = \pi_j^e$ where π_j^e indicates the 2D edge primitive Π_i^E has been extracted from, see Fig. 3.

3.2 Contours in 2D and 3D Ψ^C

By linking different 2D and 3D edge primitives, 2D contours ψ^c and 3D contours Ψ^C (see Fig. 2(c-iii)) are extracted (we neglect the 2D contours here since we do not make use of them in the applications described in Sect. 5). [1] describes this process in detail.

3D contours are coded as

$$\Psi^C = (G^C, A^C, \Sigma_G^C, B^C)$$

with $G^C = (\mathbf{p}^C, \mathbf{d}^C, G_v^C) = ((x, y, z), (d_1, d_2, d_3), G_v^C)$ and where $\Sigma_G^C \in \mathbb{R}$ represents a value computed from the uncertainty of the individual edge primitives.

The appearance attributes A^C contain averaged color and phase values, derived from the attributes of the primitives the contour consists of, i.e., $A^C = (\omega, \mathbf{c}^l, \mathbf{c}^m, \mathbf{c}^r)$.

Between contours the following relations are defined: angle $R_a(\Psi_i^C, \Psi_j^C) \in [0, \pi]$, distance $R_d(\Psi_i^C, \Psi_j^C) \in \mathbb{R}$, normal distance $R_{nd}(\Psi_i^C, \Psi_j^C) \in \mathbb{R}$, coplanarity $R_p(\Psi_i^C, \Psi_j^C) \in \mathbb{R}$, and finally co-colority $R_c(\Psi_i^C, \Psi_j^C)$. See [1] for more detailed information regarding these relations and Fig. 3 for an illustration.

4 Hierarchy in the surface domain

In this section, we describe the feature hierarchy in the surface domain as a novel contribution of our paper in detail. Sects. 4.1.1 and 4.1.2 describe the local 2D and 3D texlet descriptors, while Sect. 4.1.3 defines the relations between texlet features. The creation of surfling features is presented in Sect. 4.2.1, and relations between surflings are defined in Sect. 4.2.2. Finally, a definition of surface descriptors is presented in Sect. 4.3.

4.1 Local surface descriptors

As in the edge domain, our system computes local 2D and 3D descriptors as described in the following two subsections.

4.1.1 2D texlet π^t

A 2D texlet π^t is used to represent small textured image patches. Therefore, a 2D texlet is extracted from a position in the image, if a specified area around the sampling point is classified as containing texture [15]. This simple 2D feature only consist of a point in image coordinates and some basic appearance information. To ensure uniform sampling a hexagonal grid is used to determine the sampling points. The 2D texlet is formalized as:

$$\pi^t = (G^t, A^t)$$

with the geometric information $G^t = (x, y)$ and the appearance information $A^t = (\mathbf{h}^n, \mathbf{s}^n, \mathbf{v}^n, n)$, $n = 5$. A^t is represented as a color histogram with five bins for each of the three channels in the HSV color space. Alternatively, the mean color can be used as a simpler appearance descriptor. The most important use case of 2D texlets is the provisioning of appearance information in the 3D texlet extraction process which is described in the next section.

4.1.2 3D texlet Π^T

3D texlets represent small, flat textured surface patches in Euclidean space. They are constructed by fitting a plane to the 3D points in the neighborhood of the 3D point related to a 2D texlet's position. Their computation therefore requires 2D texlets and a 3D point cloud (see Fig. 2(s-ii)). These

point clouds are created by means of classical stereo processing or using a Kinect camera. A 3D texlet is described as:

$$\Pi^T = (G^T, A^T, \Sigma_G^T, B^T).$$

In order to minimize the propagation of noise from the 3D point cloud to the texlet, RANSAC [19] is used during the reconstruction process. The texlet extraction algorithm is thus not executed on the whole local 3D cloud, but an optimal sub-sample set chosen by a modified RANSAC algorithm. This algorithm searches for a plane with best support in the point cloud and then discards points that are too far away from the plane.

The geometric information is encoded in a 7D attribute vector G^T:

$$G^T = (\mathbf{p}^T, \mathbf{n}^T, s^T) = ((x, y, z), (n_1, n_2, n_3), s^T)$$

with surface normal $\|\mathbf{n}^T\| = 1$ and the size $s^T \in \mathbb{R}$. The geometric uncertainty is coded by a matrix $\Sigma_G^T \in \mathbb{R}^6 \times \mathbb{R}^6$.

The position $\mathbf{p}^T$ is defined as the center of gravity calculated from the local neighborhood within the 3D point cloud. The orientation $\mathbf{n}^T$ is computed using principal component analysis (PCA) applied to the 3D positions of the points in the neighborhood. PCA projects the data onto a new coordinate system such that the first axis coincide with the direction in which the largest variance occurs, the second axis with the second largest, and so forth. In this case, the direction with the smallest variance will be orthogonal to the plane constituted by the local 3D point cloud neighborhood, and thus gives the texlet normal $\mathbf{n}^T$.

The texlet normal $\mathbf{n}^T$ defines both orientation and direction of a surface in space, i.e., the side of the surface. This direction is implied by the viewing point, where any visible surface will always have a normal that forms an obtuse angle, to a ray connecting the optical center and the texlet, see Fig. 4. As outlined in Sect. 5.2, this direction vector can be used, once related to other texlets, to extract valuable indications about object properties such as 'openness' and 'closeness'.

The size s^T of a texlet is another outcome of the PCA process. The direction with the largest variance will be in the texlet plane in the direction of the largest extend. The corresponding Eigenvalue is thus used to describe the size of the texlet.

The uncertainty Σ_G^T associated to the geometric information of the texlet is computed differently for the kinect and the stereo case since the reconstruction geometry as well as the underlying noise models differ in both cases. Due to page number limitations we cannot give a precise definition of the uncertainties here but refer to [60]. In the following paragraphs we sketch the computation in both cases.

When standard stereo is used, Gaussian noise is added to the 2D points and propagated from 2D to 3D during the reconstruction process. By means of Monte Carlo simulation, a set of texlets is calculated. From these the uncertainty covariance matrix is calculated.

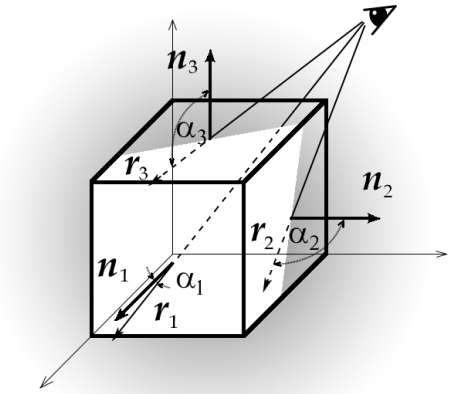


Fig. 4 Choosing the correct surface normal. Note that only two out of three sides of the box that are visible on the illustration, are going to be visible from the marked point of view on the top right. $\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$ are outward surface normals marking the sides of the cube visible on the illustration. $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are camera rays, vectors originating from the marked point of view and pointing to the surface normals. α_1, α_2 and α_3 are the angles between corresponding camera rays and normals. The following statement holds for any point of view: A normal of a visible surface will always form an obtuse angle to the camera ray—the vector connecting the point of view and the point on the surface where the normal is measured. This observation is used to assign correct normal orientations to reconstructed texlets.

When using a Kinect camera, 3D data is provided directly and the stereo method cannot be utilized. In [60] the point-wise reconstruction uncertainty has been investigated, leading to a Kinect noise model dependent on the distance to the observed object and distance to the principal point. This noise model is used to add Gaussian noise to the reconstructed points and, as in the stereo case, by using Monte Carlo simulation, calculate the uncertainty covariance matrix. We can achieve a computational speed of around 5–10 Hz for scenes consisting of ∼4000 texlets, depending on the number of RANSAC iterations used in and the Monte Carlo simulation for uncertainty calculation (for details, see, see [60]).

The appearance attribute is propagated from the 2D level $A^T = A^t$ and consists of either the mean color, or simple color histogram information computed from the HSV color space (see Sect. 4.1.1).

The confidence $B^T \in [0, 1]$ can be set according to the supporting matching confidence or matching cost from the underlying dense vision algorithm in the stereo case. The output from the Kinect however, does not provide us with any extra information on confidence or reliability, so we use a prior confidence of correct observations with the Kinect.

4.1.3 Texlet relations

In this section, we present the relations defined for texlets. As shown in Sect. 5, these relations coding contextual infor-

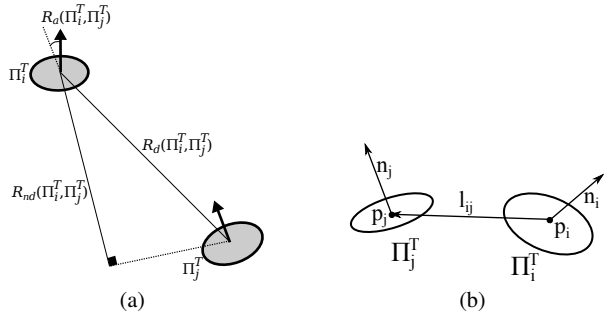


Fig. 5 Texlet attributes and relations. (a) Euclidean distance between texlets $R_d(\Pi_i^T, \Pi_j^T)$, angle between texlets $R_a(\Pi_i^T, \Pi_j^T)$ and normal distance for texlets $R_{nd}(\Pi_i^T, \Pi_j^T)$. (b) Coplanarity relation for texlets. Used for creating texlet links for surflings creation. p_i and p_j are the positions, n_i and n_j are the normals of texlet the primitives Π_i^T and Π_j^T and $\mathbf{l}_{ij}$ is the direction of the line connecting the two texlets.

mation will provide relevant information for various tasks such as grasping or pose estimation.

The **correspondence** relation between 2D and 3D texlets is expressed as $\Phi(\Pi_i^T) = \pi_j^t$ where π_j^t indicates the 2D texlet Π_i^T has been extracted from (analogously to the edge domain, see Fig. 3).

The **neighboring** relation is connecting the 2D texlets that originate from the neighboring cells on the hexagonal grid. The relation is then propagated to 3D and can be written as $R_{nb}(\Pi_i^T, \Pi_j^T) \in \{0, 1\}$.

The **co-colority** relation between two texlets is denoted as $R_c(\Pi_i^T, \Pi_j^T) \in \mathbb{R}$. We can choose to calculate this relation either directly using RGB differences or using the CIE 1994 color difference [31], depending on the application.

The **Euclidean distance** between texlets is defined as $R_d(\Pi_i^T, \Pi_j^T) = \|\mathbf{p}_i^T - \mathbf{p}_j^T\| \in \mathbb{R}$, see Fig. 5(a).

The **angle** between two texlets is computed as the angle between the texlets' normals $R_a(\Pi_i^T, \Pi_j^T) = \measuredangle(\mathbf{n}_i, \mathbf{n}_j) \in [0, \pi]$, see Fig. 5(a). In some cases, when the noise present in the data is high, the texlet orientation can not be extracted reliably enough. In such cases we base the computation of angle on the positions of neighboring texlets instead of texlet normals.

The **normal distance** for texlets $R_{nd}(\Pi_i^T, \Pi_j^T) \in \mathbb{R}$ is defined by the distance between one texlet's position and the plane created by the other's position and normal, see Fig. 5(a). Therefore, the normal distance is computed as: $R_{nd}(\Pi_i^T, \Pi_j^T) = (\mathbf{p}_i - \mathbf{p}_j) \cdot \mathbf{n}_j$.

The **coplanarity** relation $R_p(\Pi_i^T, \Pi_j^T) \in [0, 1]$ aims to determine if two texlets lie in the same plane by combining the position and orientation info. Fig. 5(b) shows two texlets Π_i^T, Π_j^T with their normals $\mathbf{n}_i, \mathbf{n}_j$. $\mathbf{l}_{ij}$ is the direction of the connecting line $\mathbf{l}_{ij} = \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|}$. The coplanarity score is a combination of two scores. The first one is a cosine between normals: $S_1 = \mathbf{n}_i \cdot \mathbf{n}_j$ and is favoring texlets with similar orientation. The second score, $S_2 = max(|\mathbf{n}_i \cdot \mathbf{l}_{ij}|, |\mathbf{n}_j \cdot \mathbf{l}_{ij}|)$, is a distance which is ideally zero, and tells both about the similarity of orientations, but also filters out parallel texlets, i.e., texlets that do have a similar orientation, but do not belong to the same surface. The final coplanarity relation is computed as $R_p = \frac{S_1 - S_2 + 2}{3}$, which leads to a normalization ($R_p \in [0, 1]$).

The values of the geometric relations mentioned in this section will stay constant if the relative geometric position of the two texlets stays constant. In case the texlets move relative to each other (e.g., in case of a non-rigid object) they will not be constant. Systems interested in dealing with non-rigid objects need to capture this on higher levels of representation as many of the patch based non-rigid object representations do.

4.2 Surflings Ψ^{SL}

Sect. 4.2.1 explains how surfling features Ψ^{SL} are created and defines the surfling parametric description. Sect. 4.2.2 presents surfling relations.

4.2.1 Computation of surflings

Surflings in the ECV system are semi-global surface features derived by grouping similar 3D texlets using geometric and appearance information. Grouping is done by creating links between similar neighboring texlets and propagating this connection using the transitivity relation. The resulting large sets of texlets represent premature surfaces. When creating links between neighboring texlets, due to noise it is possible that connections between texlets that do not belong to the same surface are created. Since an unconstrained use of the transitivity relation over long distances can lead to non-optimal grouping, the ECV system subdivides derived large sets into smaller sets of texlets. Surflings represent a step in between texlets and surfaces.

Surflings are created in fours steps S1–S4:

Creating links (S1): Based on co-colority, Euclidian distance and coplanarity, as defined in Sect. 4.1.3, we derive three criteria for creating links $\boldsymbol{L}_{ij}^T(\Pi_i^T, \Pi_j^T) \in \{0, 1\}$ between pairs of 3D texlets. The criteria are controlled by the three parameters: $t_1^{SL}, m^{SL}, t_2^{SL}$ which put limits to the geometry and appearance differences aiming at grouping texlets belonging to the same surface. For two texlets to be linked the following criteria need to be fulfilled:

Co-colority: The co-colority relation score is below some fixed threshold: $R_c(\Pi_i^T, \Pi_j^T) < t_1^{SL}$.

Euclidean Distance: The 3D distance between two texlets is below a threshold. The threshold varies with the size s^T of the texlets, (see Sect. 4.1.2). This variation originates from the fact that the back-projected size of a 2D texlet has an influence on the size of a 3D texlet.

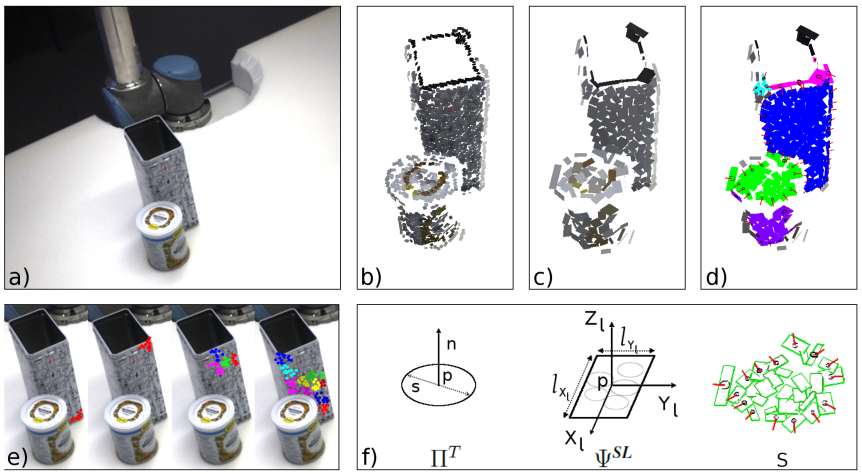


Fig. 6 Creation of the feature hierarchy in the texture domain. a) Image of an example scene. b) Corresponding texlets representation. c) Surflings representation. d) Segmented surfaces are highlighted in different colors. Boundary surflings are marked with black circles and red lines are indicating boundary normals. e) Intermediate steps in the process of creating surflings from texlets. The four figures show large sets of texlets connected by relations co-colority, coplanarity and Euclidean distance, reprojected to the original image. In the first two figures, the derived groups of texlets are small and will each constitute only one surfling. In the second two figures, the derived large sets of texlets are subdivided into smaller sets and each of the smaller sets will form one surfling. f) Illustrations of a single texlet, a single surfling (based on five texlets) and a surface (consisting of tens of surflings).

The larger the distance between a 3D texlet and the camera is, the larger size the 3D texlet will have (note that not only the distance but also the orientation has an influence on the size), and thus the larger expected distance to the neighboring texlet. Hence, we define

$$R_d(\Pi_i^T, \Pi_j^T) < t(s_i^T, s_j^T), \text{ with}$$

$$t(s_i^T, s_j^T)) = m^{SL} \cdot \frac{(s_i^T + s_j^T)}{2},$$

where s^T is the size of the 3D texlet and m^{SL} is a parameter, controlling the desired distance, independently of the position in 3D space[2]. Instead of the averaged texlet size, the maximum of the two can also be used: $t(s_i^T, s_j^T)) = m^{SL} \cdot max(s_i^T, s_j^T)$.

Coplanarity: The neighboring texlets have a coplanarity relation score below a certain threshold $R_p(\Pi_i^T, \Pi_j^T) < t_2^{SL}$. Although curved surfaces can have a deviation in orientations, at the level of local neighboring texlets, the coplanarity can still be used as a criterion for determining continuity of the surface.

Creating large sets of texlets (S2): The created links between the pairs of similar texlets are propagated using the transitivity relation to derive large sets of connected texlets, see Fig. 6(e).

Subdividing large sets of texlets into small sets of texlets (S3): Large sets of texlets are subdivided in the following way. The 3D positions $\mathbf{p}_i^T$ of texlets belonging to one large set are back-projected to 2D: $\Phi(\mathbf{p}_i^T) = \mathbf{p}_i^t$. Based on their 2D coordinates, texlets are clustered using the K-means algorithm. The average number of texlets per cluster n_t is used as a parameter. n_t, typically ranges from 5–10 and controls the granularity of the surflings, see Fig. 6(e).

Surfling parametrization (S4): A surfling Ψ^{SL} is a rectangular planar patch that has a position $\mathbf{p}^{SL}$, orientation $\mathbf{o}^{SL}$, size (width and length) $\mathbf{s}^{SL}$, uncertainty Σ_G^{SL} and average color c^{SL}, see Figs. 6(c) and 6(f). The boundary label b^{SL} tells if the surfling is located at the boundary of the segmented surface (see Sect. 4.3) and is used in Sect. 5.1 to indicate finger poses in the context of grasping. The boundary normal $\mathbf{d}_b^{SL}$ defines the direction of the local boundary and is assigned to each boundary surfling. Hence:

$$\Psi^{SL} = (G^{SL}, A^{SL}, \Sigma_G^{SL}),$$

where $G^{SL} = (\mathbf{p}^{SL}, \mathbf{o}^{SL}, \mathbf{s}^{SL}, b^{SL}, \mathbf{d}_b^{SL})$, $\mathbf{s}^{SL} \in \mathbb{R}^2$, $A^{SL} = c^{SL}$ and $\Sigma_G^{SL} \in \mathbb{R}$. The position is parametrized as $\mathbf{p}^{SL} = (x, y, z)$, and the orientation $\mathbf{o}^{SL} = (\mathbf{X}^l, \mathbf{Y}^l, \mathbf{Z}^l)$. Width and length $\mathbf{s}^{SL} = (l_X, l_Y)$ are derived by means of PCA applied to the surflet's member texlets' positions. The results of the PCA are stored as a local coordinate frame, where the origin is the center of mass of the member texlets' positions. The axes of the local coordinate system are determined by the components of the PCA. The local $\mathbf{X}^l$ and $\mathbf{Y}^l$ axis take the largest and the second largest direction. The third axis $\mathbf{Z}^l$ is orthogonal to the first two axes and its direction is implied by the viewing point, similarly to the case of 3D texlets (Sect. 4.1.2 and Fig. 4). The direction on the second axes is chosen to derive a right hand coordinate system $\mathbf{Y}^l = \mathbf{Z}^l \times \mathbf{X}^l$. $\mathbf{X}^l$ and $\mathbf{Y}^l$ together define the plane of a surfling, while $\mathbf{Z}^l$ defines the normal and is also marked with $\mathbf{n}^{SL}$. The length and width of the surfling, in $\mathbf{X}^l$ and $\mathbf{Y}^l$ direction of the local coordinate frame, is given with (l_X, l_Y),

[2] The parameter m is typically in the range [2,4].

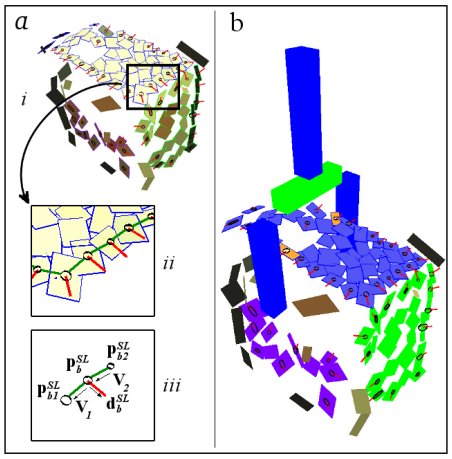


Fig. 7 Surfaces and boundaries. (a) shows the ECV surfling representation of a box, where the sides of the box have been segmented into surfaces. (a-i) Detail of the top surface boundary. (a-ii) Boundary normals $\mathbf{d}_b^{SL}$ are drawn in red and are pointing out of the surface. (a-iii) Boundary direction computation. (b) The surface segmentation has been made more explicit for the purpose of illustration. The figure shows an example two-fingered grasping action, triggered by two boundary surflings of the top surface, highlighted in orange.

$l_X = 4 \cdot \sqrt{E_X}$, $l_Y = 4 \cdot \sqrt{E_Y}$, where E_X and E_Y are the Eigenvalues in the respective directions, see Fig. 6(f). The color property of the surfling is derived by averaging the color of the underlying texlets: $c^{SL} = \frac{1}{N} \sum_{i=1}^{N} c_i^T$.

Once surfaces are computed (see Sect. 4.3), the system identifies a subset of member surflings that are positioned on the boundary of the surface, see Fig. 7(a). A boundary surfling, Ψ_b^{SL}, is labeled with $b^{SL} = 1$ and has the additional boundary normal ($\mathbf{d}_b^{SL}$) property. Fig. 7(a-ii) shows a detail of the boundary of a top surface of a box. $\mathbf{d}_b^{SL}$ lies within the surface plane of the surfling and points out of the surface. It is computed as follows (see Fig. 7(a-iii)): The two vectors connecting the center of a boundary surfling $\mathbf{p}_b^{SL}$ with the centers of its two closest boundary surflings $\mathbf{p}_{b1}^{SL}$, $\mathbf{p}_{b2}^{SL}$ are drawn: $V_1 = \mathbf{p}_{b1}^{SL} - \mathbf{p}_b^{SL}$ and $V_2 = \mathbf{p}_b^{SL} - \mathbf{p}_{b2}^{SL}$, and normalized: $V_1' = V_1/|V_1|, V_2' = V_2/|V_2|$. Their normalized sum is given with $V'' = (V_1 + V_2)/|V_1 + V_2|$, and it is further projected to the surfling's plane $V_p'' = V'' - (V'' \cdot \mathbf{z}^{SL}) \cdot \mathbf{z}^{SL}$. The normal vector $\mathbf{d}_b^{SL}$ is determined as a direction orthogonal to V_p'' lying inside the surfling plane $\mathbf{d}_b^{SL} = V_p'' \times \mathbf{z}^{SL}$, where the sign is chosen so that the normal is pointing out of the surface.

The geometric surfling uncertainty is represented as a sum of traces of individual texlet uncertainties:

$$\Sigma_G^{SL} = \frac{1}{N} \sum_{i=1}^{N} Trace(\Sigma_{Gi}^T).$$

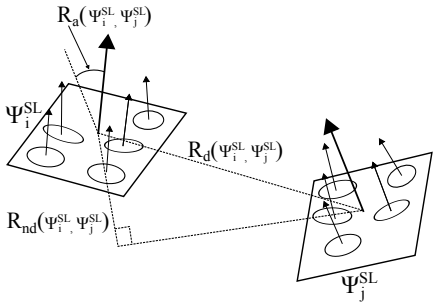


Fig. 8 Surfling relations.

4.2.2 Surfling relations

Similarly to the texlet case, we define the following relations between surfling features.

The **Euclidean distance** between surflings is defined as $R_d(\Psi_i^{SL}, \Psi_j^{SL}) = \|\mathbf{p}_i^{SL} - \mathbf{p}_j^{SL}\| \in \mathbb{R}$, see Fig. 8.

Co-colority $R_c(\Psi_i^{SL}, \Psi_j^{SL}) \in \mathbb{R}$ is calculated using RGB differences or using the CIE 1994 color difference [31].

The **angle** between surflings is defined as $R_a(\Psi_i^{SL}, \Psi_j^{SL}) = \measuredangle(\mathbf{n}_i^{SL}, \mathbf{n}_j^{SL}) \in [0, \pi]$, see Fig. 8.

Coplanarity $R_{cs}(\Psi_i^{SL}, \Psi_j^{SL}) \in [0, 1]$ is the defined analog to the texlet case $R_p(\Pi_i^T, \Pi_j^T)$ described in the Sect. 4.1.3.

The **normal distance** $R_{nd}(\Psi_i^{SL}, \Psi_j^{SL}) \in \mathbb{R}$ between surflings is defined as: $R_{nd}(\Psi_i^{SL}, \Psi_j^{SL}) = (\mathbf{p}_i^{SL} - \mathbf{p}_j^{SL}) \cdot \mathbf{n}_j^{SL}$.

4.3 Global surfaces $\mathcal{S}^S$ representation

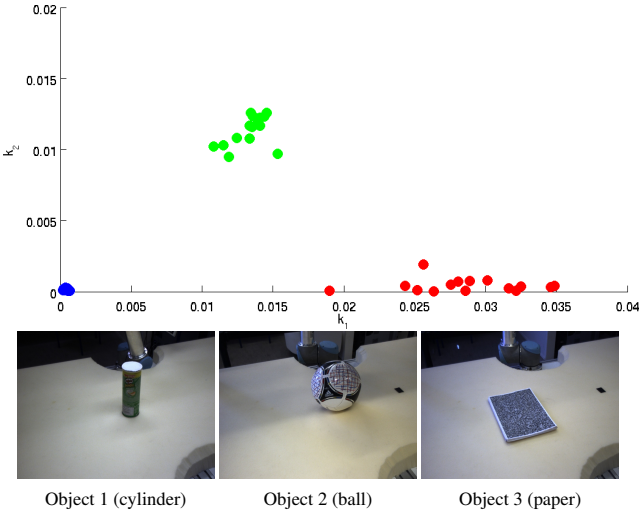
At the final stage of the hierarchy in the surface domain, we have surfaces $\mathcal{S}^S$ (see Figs. 2(s-iii), 6f). In the ECV system $\mathcal{S}^S$ are constructed from surflings, in a similar fashion as surflings are created from texlets. The system creates links between surflings with proximate position and orientation and performs grouping using the transitivity relation. The color information is not considered when grouping surflings into surfaces, allowing for surfaces with differently colored regions. The color change is nevertheless an important cue when performing segmentation on the lower level, as it occasionally does mark an end of a surface. By disabling grouping over differently colored texlets the system more often prevents wrong fitting of a surfling over two adjacent surfaces.

A link between two surflings is created if the following criteria—guided by parameters m^S and t^S—are satisfied:

The *Euclidean distance* is below a threshold which is computed individually for each pair of surflings, and is varying with the size s^{SL} of the surflings:

$$R_d(\Psi_i^{SL}, \Psi_j^{SL}) < t(s_i^{SL}, s_j^{SL}), \text{ with}$$

$$t(s_i^{SL}, s_j^{SL}) = m^S \cdot \frac{(s_i^{SL} + s_j^{SL})}{2}$$



Object 1 (cylinder) Object 2 (ball) Object 3 (paper)

Fig. 9 Principle curvatures of three different objects. The x axis and y axis refer to the principal curvatures k_1 and k_2 respectively. The red points correspond to object 1 (cylinder), the green point to object 2 (ball), and the blue points to object 3 (paper). The lower row contains an image of each object.

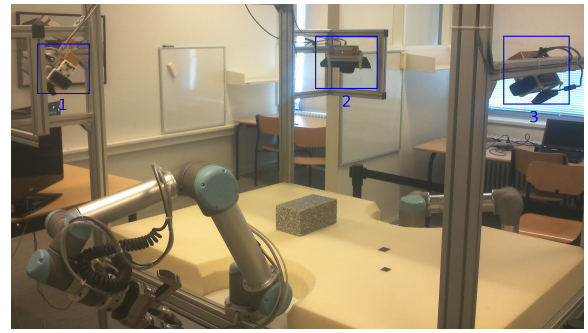
where s^{SL} is the size of the surfling and m^S is a parameter, analog to the surfling case.
The *coplanarity* relation score is below a threshold: $R_sc(\Psi_i^{SL}, \Psi_j^{SL}) < t^S$.

A surface is defined as a set of surflings and its two principal curvatures, k_1 and k_2.

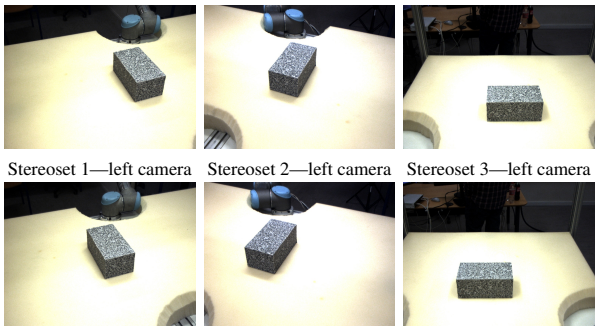
$$\mathcal{S}^S = \{\{\Psi_0^{SL}, ..., \Psi_n^{SL}\}, \{k_1, k_2\}\}$$

The principle curvatures for any surface are orthogonal and indicate the maximum and the minimum curvature values respectively [62]. The approach we present in this paper considers the positions of the 3D texlets of which the surface is composed as the data points of this surface. These texlets' positions are exploited to fit a quadratic polynomial to obtain a continuous and differentiable surface approximation. To do this properly, the texlets positions need to be relative to the surface local frame. The pose of the local frame, to which the surface will be rotated, is obtained through PCA, on the positions. The principle curvatures of the surface are evaluated at the center of the local frame. The curvature is calculated from the polynomial by means of differential geometry as described in [62].

Fig. 9 shows the principle surface curvatures extracted from three objects; a cylinder, a ball and a flat piece of paper. For each object, five instances with different poses recored in the set-up shown in Fig. 10 are included. For each instance, the two principal curvatures of the main surface computed from each of the three stereo cameras resulting in 15 curvature estimates are shown. Intuitively, for the ball k_1 and k_2 should be equal and non-zero values (related to the ball's radius), while for the paper both values should be zero. In the case of the cylinder, we expect k_1 to have non-zero value with larger curvature compared to the ball (since the radius of the circle generated by a planar horizontal cross section with the cylinder is smaller than the radius of the ball). k_2, on the other hand, is expected to be zero (the cylinder along its hight has a flat surface). From the figure we can see clearly that our surface curvature estimation comes in line with that. We can also see that the principle curvatures of the three object form three distinctive clusters reflecting the different curvature nature of the objects.



(a)



Stereoset 1—left camera Stereoset 2—left camera Stereoset 3—left camera
Stereoset 1—right camera Stereoset 2—right camera Stereoset 3—right camera

(b)

Fig. 10 Setup for object representation experiments. (a) Image of the setup showing the three stereo cameras which are placed around one of the objects used for the experiments. (b) Views from the different cameras.

Once surfaces are computed, the system identifies a subset of member surflings that are positioned on the boundaries of the surfling set, see Figs. 6(f) and 7. Boundary surflings (Ψ_b^{SL}) are labeled with $b^{\Psi^{SL}} = 1$ as explained in Sect. 4.2.1.

5 Applications

In this section, we present three applications that make use of the two parallel hierarchies introduced in this paper: grasping of unknown objects based on elementary 'reflexes' (Sect. 5.1), view point invariant object representations (Sect. 5.2) and pose estimation (Sect. 5.3). In all three applications, we can demonstrate the complementary potential

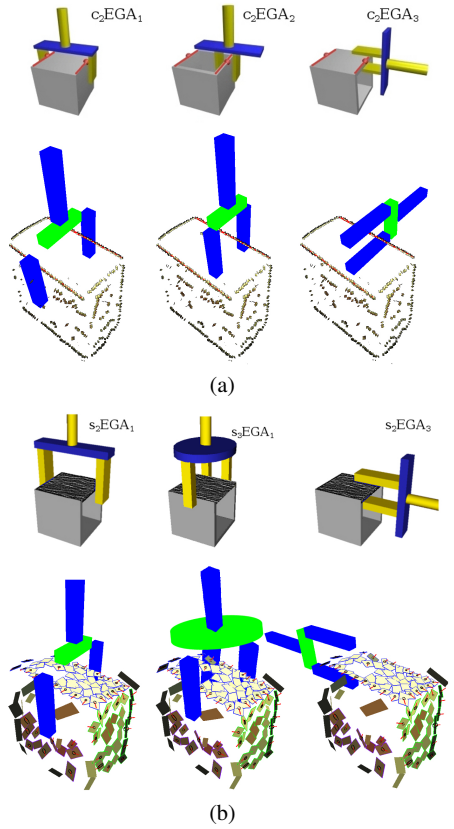


Fig. 11 Elementary grasping actions (EGA) are based on the contour and surface features from the ECV image representation. The top row shows the symbolic grasp types, while the bottom row shows the actual grasps generated from the real data. (a) Contour EGAs based on pairs of co-color and co-planar contours. (b) Surface EGAs based on the segmented top surface of a box. See Fig. 12 for examples of real grasps.

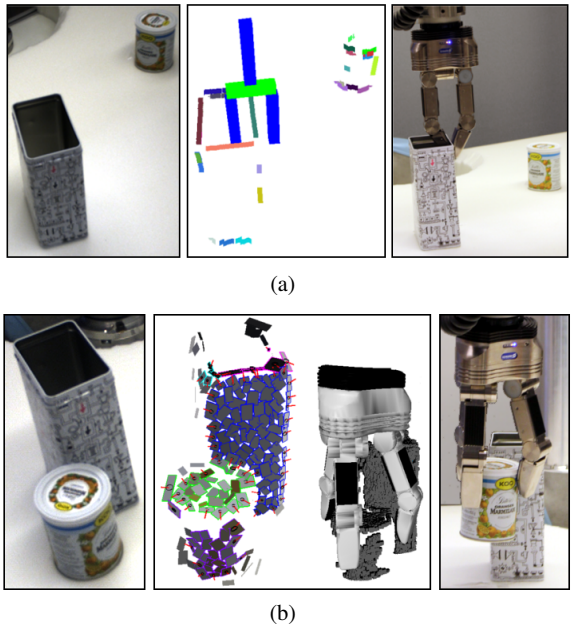


Fig. 12 Two example grasping actions performed in a real experimental setup. (a) Left: Original scene, Middle: ECV contour representation and the chosen two-finger grasping hypothesis, Right: Successful contour-based grasping action. (b) Left: Original scene, Middle: ECV surflings and surfaces representation and the chosen three-finger grasping hypothesis, Right: Successful surface-based grasping action.

of the two domains and the advantage of having access to the different levels of the hierarchy. Note that the required computational time is sufficient to allow for applications of the ECV system in robotics applications. For example, the computation of the complete hierarchy from three stereo cameras in parallel (as in the set-up shown Fig. 10) takes approximatively 2 seconds. The computation of the first levels of the hierarchy—which is already sufficient for some tasks—can be done in more than 5 Hz making use of GPU technology (for details see, [32, 60]). Here we give only the essence of the applications to stress the aspects of the hierarchy we used. For more detail we refer to [7–9, 41, 58].

5.1 Grasping unknown objects

As the first application, we present and test three methods for generating visual based grasps of unknown objects. The first grasping method is based on the contour descriptors derived in edge hierarchy (see Fig. 11(a)), the second two grasping methods are based on the surfling and surface descriptors derived in the texture hierarchy (see Fig. 11(b)). The grasping algorithm is described only briefly here, neglecting all robotic related issues. We refer to [41, 66] for more technical details and an extensive experimental evaluation. Here we want to exemplify the complementary role of the two hierarchies in the grasp generation process.

The contour based method searches for pairs of contours (Ψ_i^C, Ψ_j^C) that are both co-planar $(R_p(\Psi_i^C, \Psi_j^C) < t_1)$ and co-color $(R_c(\Psi_i^C, \Psi_j^C) < t_2)$, see Figs. 11(a), 12(a). Such contour pairs are likely to originate from the same surface on an object. The grasps are generated with respect to the selected contours and the common plane fitted to the two contours (for details, see [66]).

Surface based grasps are constructed around individual surfaces $\mathcal{S}^S$, see Figs. 11(b), 12(b). The first surface based grasp method (in the following called PCA method) creates simple actions aiming at grasping a surface as a whole. We perform PCA on the positions ($\mathbf{p}^{SL}$) of the surflings belonging to the surface $\mathcal{S}^S$. The grasps are generated with respect to the main directions of the surface derived from the PCA.

The second surface based grasp method makes explicit use of the boundary information associated to the surflings within a surface to find optimal contact points. Hence, the boundary method operates on a single surface $\mathcal{S}^S$, but uses more fine grained information from the underlying surfling features represented on a lower level of the hierarchy. Boundary surflings Ψ_b^{SL}, (i.e., $b^{SL} = 1$), provide details

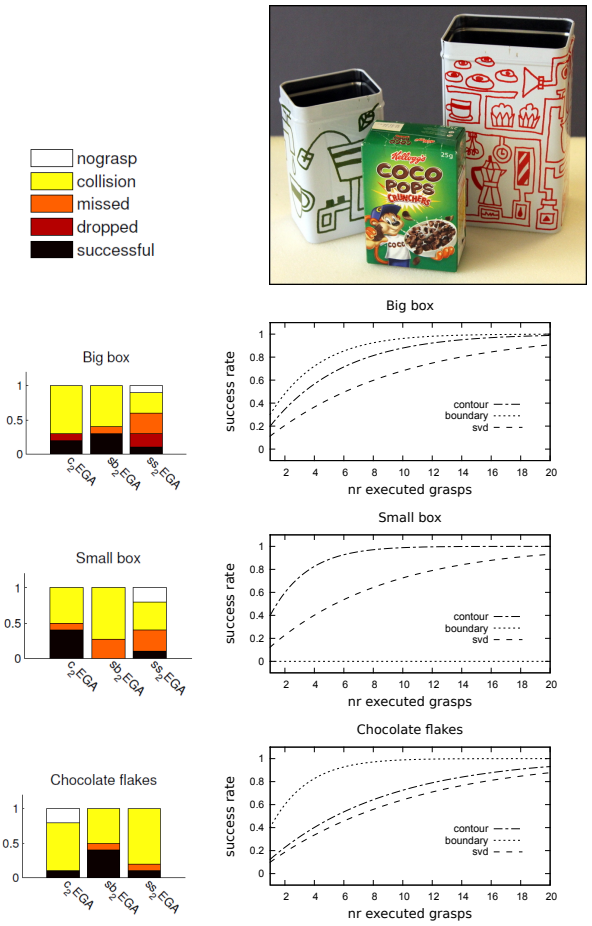


Fig. 13 Top right: Three objects grasped with a parallel jaw gripper. The relative success rate of the different grasp types (left) as well as the accumulative success rate are given as a function of the number of executed grasps (right). In the right figures, success is coded by whether the object was grasped successfully in the current attempt or any of the previous attempts.

about the local structure of the surface boundaries. The positions and the normals of the boundary surflings are in this method used to construct the contact points for grasping. Pairs and triplets of contact points are selected to produce stable two-finger and three-finger grasps (see Figs. 11(b) and 12(b)).

In [41], we have tested and compared the performance of the three grasping methods on a variety of objects. Fig. 13 presents results for the three objects selected for this article, with the aim of illustrating the complementary nature of edge and surface information. When performing experiments, objects were placed in a random orientation for each grasping attempt. On the left side, the relative distribution of successful grasps as well as grasps that did not succeed due to different reasons as indicated in the legend is shown. On the right side, the accumulated success is shown when only applying grasps based on boundary and surface information only and their combination. These results on the left show clearly that for example for the small box – due to the lack of texture primarily the edge based grasps are successful while for the chocolate flakes - which has a lot of texture – the surface based grasps are successful. The big box is somehow in-between possessing texture as well as clear edges. These findings are directly mapped to the accumulated success rate shown at the right in Fig. 13. Accumulated success means that the object has been grasped either at the current grasp attempt or at one of the grasp attempts tried before. In the case of the big box object, the surface boundary method performs best, while for the small box the contour method performs best. In the combined experiment the different grasp types are executed in an alternating way. This always performs better demonstrating the complementary nature of the different grasp types.

The results show in particular that the the two grasping methods complement each other. The contour-based methods perform better in situations where objects are not textured, while different surface-based methods complement each other in the situations where objects are textured. The PCA method suggests a limited number of grasps and performs well when the objects have a regular shape. The surface boundary method is built upon two levels of the hierarchy, i.e., both the surfling and the surface information is used. The boundary method suggests larger numbers of grasps and can deal with a larger variety of shapes.

As a consequence, we can show that the parallel use of contour and surface information for grasping unknown objects allows to grasp a larger variety of objects. In the process of grasp generation, the simultaneous use of different levels of the representation hierarchy is advantageous.

5.2 Viewpoint-invariant object representation in the ECV system

The visual representation presented in this paper has a number of interesting properties in the context of classical vision problems such as object recognition, pose estimation as well as a number of learning problems. In this section, we want to make two of those properties explicit, namely: (1) the separation between geometric and appearance information and (2) view point invariance representations in terms of relational information.

The viewpoint-invariance of our representation can be demonstrated by the system's ability to maintain stable appearance and geometric representation under viewpoint transformation. We show this stability by means of a histogram approach. In the following experiment, three stereo cameras (1024x768 resolution) organized in a close to equilateral triangle around a confined workspace have been used in order to have three views of objects inside the workspace. The setup, which models an 'intelligent production cell' relevant in, e.g., industrial assembly processes, is shown in Fig. 10(a). The setup allows for a rather complete

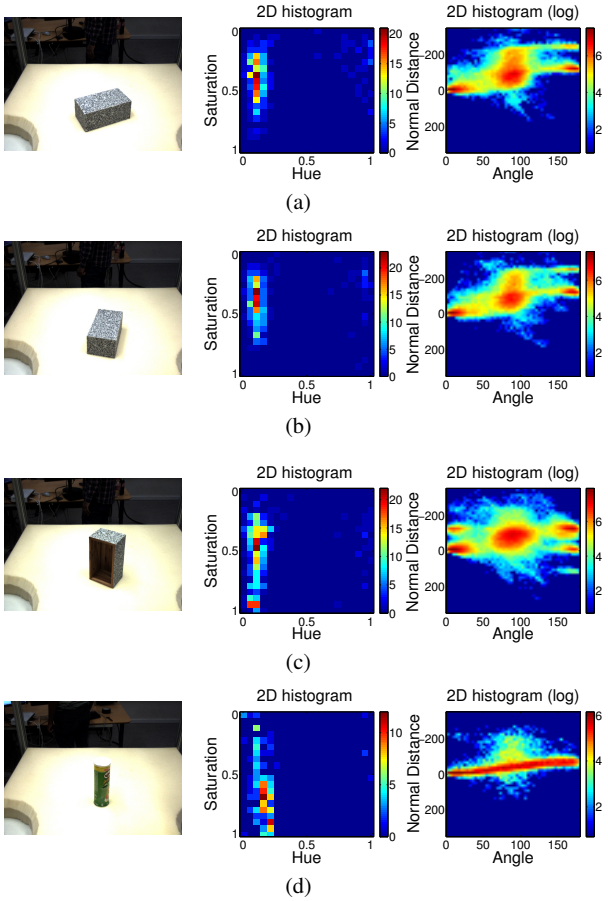


Fig. 14 Four different scene configurations and corresponding histograms. The histogram blocks for each scene consists of two-dimensional histograms of hue vs. saturation and angle vs. normal distance for all possible pairs of texlets.

representation of object (except for the surface in contact with table) when the ECV 3D features, extracted from all views, are combined.

The objects in study here are two boxes with the same dimensions. One box is a closed (see Figs. 14(a) and 14(b)) and the other is open at one side (Fig. 14(c)). In addition, we use a cylindrical object (Fig. 14(d)) for comparison. The first two examples (see Figs. 14(a) and 14(b)) show closed boxes with the same color but with different poses. The third recording is the open box (i.e., the same as the first box but with a missing side) where the inside surfaces have a different color. Fig. 14 shows histograms corresponding to a subset of texlet attributes (color) and the second order geometric relations $R_a(), R_{nd}()$ for all three views of the stereo camera.

The histograms show that the appearance and the geometric information for the first and the second examples in (see Fig. 14(a), 14(b)) are very similar despite the difference in pose. This exemplifies the view point invariance of the geometric second order relations. The change in color in Fig. 14(c), in which the brown inside surfaces are visible, with respect to the first and the second is reflected on the appearance histograms being different (an additional peak appears at hue 0.9 and saturation 0.1). The shape histogram for the cylindrical object (Fig. 14(d)) is significantly different from the box-like objects.

For the first three cases (Figs. 14(a), 14(b) and 14(c)), the two-dimensional $R_a()$ / $R_{nd}()$ histograms reveal 4 peaks. These peaks correspond to the dimensions of the box, i.e., the distances of the parallel planes. Note that negative value indicates an outward direction (see Sect. 4.2.2). Peaks falling at an angle of about 0° represent parallel surfaces pointing in the same direction (i.e., the table and the top surface) and peaks at about 180° reflect opposite directions (the four side planes) and indicate the existence of additional surfaces.

When comparing this to the histograms for object 14(c), another peak (at about $180^\circ/120\,mm$) appears. The openness of the in this case box allows for the extraction of texlets from the inside faces. As a result of that, we obtain texlets pointing in opposite directions. In the case of the fourth object, we can see again that we obtain a completely different geometric histogram making a clear distinction from the other cases.

The above examples shows that very similar histograms for color and geometry are obtained despite their pose differences. Furthermore, it shows that the shape relations code properties of the object in an easily identifiable way as individual peaks in histograms. Hence, the ECV system provides view-point invariant representations of objects in which appearance and geometric information is separated .

Based on the visual representation described here, we developed in [58] an object recognition system in which a random forest classifier [6] determines the most relevant relations (binned in histograms) as shown in Figs. 14 for object classification. We established a dataset of 56 objects (shown in Figs. 15(a)) and investigated the classification rate with texlets extracted from both stereo and Kinect, and for one and three views. Figs. 15(b) shows the classification rate on the test set versus the number of instances used for training. The figure shows two important aspects of our representation: First, we only need few training examples to reach the steady-state of the classification rate. This is due to the fact that the feature relations used are highly view-point invariant as already indicated in Figs. 14. Second, because we use global relations, it is advantageous to have a multi-view camera system in such an intelligent production-cell environment in which the object recognition is performed.

5.3 Pose estimation

This section presents an application of the ECV system for computing the pose of known 3D objects. At different stages

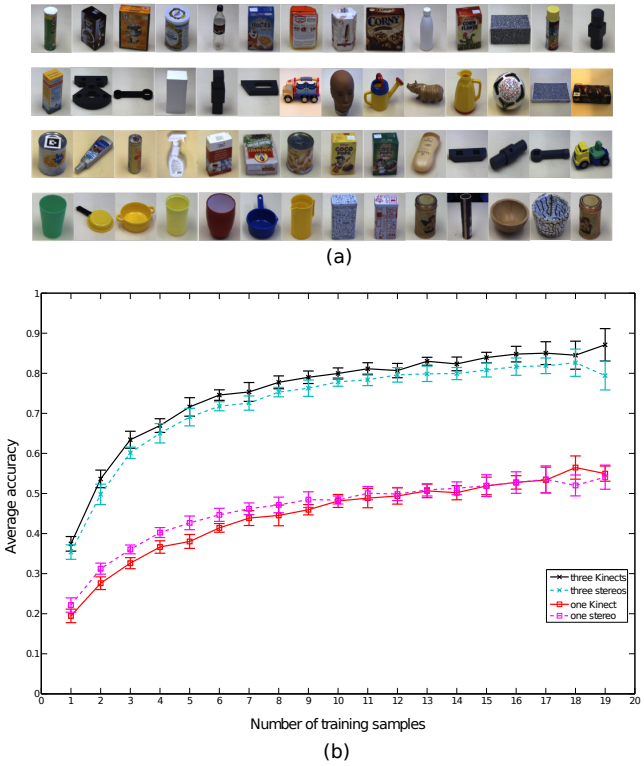


Fig. 15 Object recognition results (a) objects used. (b) classification performance on one and three views taken from both Kinect and Stereo cameras.

of the process, we use 3D descriptors both at the primitive level (see Figs. 2(c-ii) and 2(s-ii)) and higher levels of the hierarchy (see Figs. 2(c-iii) and 2(s-iii)). Additionally, we show (as analogously done for grasping in Sect. 5.1) how the two visual domains presented in Sects. 3 and 4 complement each other for this task and also make combined use of shape and appearance information.

While the space spanned by the ECV hierarchy provides the visual input for pose estimation, a model representation needs to be extracted prior to pose estimation. Having this representation available, we search for local feature correspondences between the model and the scene and finally solve the alignment problem as outlined in Sect. 5.3.1. In Sect. 5.3.2, we present results both for a large set of controlled experiments as well as for a real setup.

5.3.1 Local contextual representation

The representation used for pose estimation is based on 3D ECV features extracted from a view, i.e. a training view of the object or a scene view for testing. For each feature, we calculate contextual information based on local appearance and geometry relations described in Sects. 3 and 4. The use of contextual or local descriptors is a well-established approach, which has been investigated thoroughly both for appearance-based keypoints in image data (see, e.g., [2, 3, 52]) and for 3D point clouds (see, e.g., [25, 34, 74]).

Our local descriptors make use of the advantages of both approaches attempting to utilize both appearance and shape provided by the entities in the visual hierarchy. We use image data for extracting the appearance part (Figs. 2(c-i) and 2(s-i)) in the edge/surface domain, resulting in a more dense representation than typically used by keypoint detectors. In contrast to regular 3D shape-based approaches, however, the number of point descriptors is lower, since we apply our local 3D descriptors to the entities in the edge/surface domain (Figs. 2(c-ii), 2(s-ii), 2(c-iii) and 2(s-iii)), and not to every point in the point cloud.

For the appearance relations, we use the three color channels. To speed up computations, we use simple RGB differences. For each channel, we generate a 16-bin histogram of the local distribution of color differences between all possible feature pairs in the neighborhood.

In addition to this, we utilize geometric relations. Again, we identify all feature pairs (i, j) in a local neighborhood and calculate three angular relations $R_a()$ as follows: 1) between the 3D orientations of the features, 2) between the first feature orientation and the line direction vector $\mathbf{l}_{ij}$ between the features, and 3) between the second feature orientation and $\mathbf{l}_{ij}$. The orientation of an feature is defined by either the 3D direction **d** or the 3D surface normal **n**, depending on whether the feature is in the contour or in the surface domain. As with the appearance relations, we generate 16-bin histograms for each of these three geometric relations.

5.3.2 Experiments

The method for testing the strength of a descriptor by matching two different views of the same object described in the previous section was performed systematically for a limited set of objects from the RGB-D database. We chose 10 different objects of varying shape, appearance, texture and geometry. For these objects, which were all captured on a turntable, we considered the first and the fifth frame in the generated sequence, the first frame representing a "natural" frontal view of the object [49]. Now the test is performed in exactly the same way as the common benchmark for 2D descriptors [55], namely by counting the number of matches which are within a small distance threshold.

We used two complementary representations of objects in these tests, namely the texlet-based context descriptors first presented in [9] as well as the line segment-based context descriptors presented in [7]. Note that for the line segments, the absolute number of features computed in an object view is fairly low. This is also the case for interest point based descriptors such as SIFT and SURF. The results of the descriptor matching experiments are shown in Fig. 16, given as recall or true correspondence rate. We compare our

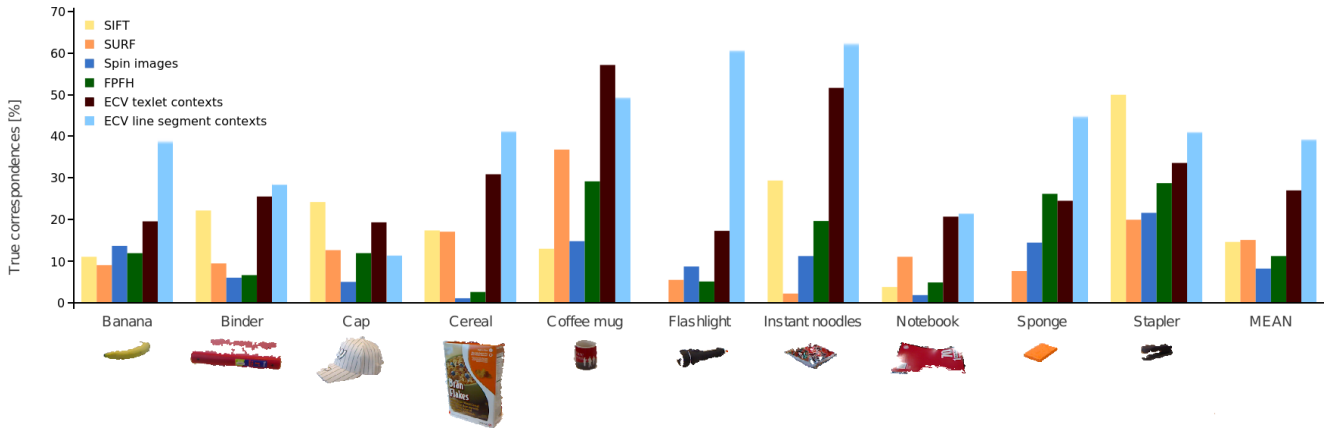


Fig. 16 Descriptor matching results for our ECV-based descriptors and a set of image (SIFT and SURF) and shape (Spin images and FPFH) descriptors from the literature.

descriptors with state of the art image descriptors, SIFT [52] and SURF [2], and shape descriptors, Spin images [34] and FPFH [73]. The image descriptors come with dedicated keypoint descriptors, but for the shape descriptors we compute descriptors at all surface points. Although the SIFT descriptor shows good performances, the ECV-based descriptors consistently outperform the other methods, due to the fact that they capture both the variation in geometry and appearance in the test set. Note also that an object such as the cap is better described by the texlet descriptors, most likely due to the many ambiguities in the appearance of the edges on the surface.

For testing pose estimation performance, we first show results for a controlled experiment using rendered stereo images of textured CAD models acquired from a real setup. For this purpose, we have used the Karlsruhe Institute of Technology (KIT) database of household objects [37]. Finally, we apply our method to a real scene provided by the RGB-D database.

In the first controlled experiment, we extract local descriptors at the high level based on contours and surflings. The right part of Fig. 17 shows an rendered view (left stereo image only) of an example KIT object and its extracted 3D contour/surfling representation in the edge/surface domain. Using the extracted features, we generate local descriptors which are used for the correspondence search. In these experiments, we extract an object representation from one training view. We then perform pose estimation on a test set of stereo frames showing incremental out of plane rotations of the object by $\{5°, 10°, \ldots, 40°\}$ relative to the training view. Examples of training/test views are shown in Fig. 17.

For the same object, we now repeat this process of generating one training view and eight incremental test images until we have visited all possible viewpoints around the object. For each object, this results in eight training views, each with eight associated test views. We thus have 64 estimation experiments per object with rotational displacements ranging from $5°$ to $40°$. At the time of writing, the KIT database consists of 112 textured object, resulting in a total of 7168 test views, 896 for each of the eight training views in the rendered environment. For this data, we can thus compare the output pose of our algorithm with the ground truth pose used during rendering, and report absolute error estimates.

The pose estimation is done using a baseline n-point RANSAC algorithm, where we repeatedly sample three feature points, estimate a hypothesis pose and validate the pose model. The pose estimation process has been performed using only contour features, only surflings and both in combination. In all three cases, we use both appearance and shape information to generate the local description of a feature for matching.

During the validation phase of each RANSAC iteration, we use the standard RANSAC criterion, namely by counting the number of features, or *inliers*, that support the sampled pose model. We have found that instead of performing this step at the high level (contours/surflings), we can achieve a much more reliable validation of a hypothesis pose at the primitive level, i.e. using the line segments and texlets. This corresponds to the levels shown in Fig. 2(s-ii) and Fig. 2(s-iii). Looking back at Fig. 17 (rightmost), we also see that the high-level representation is very sparse. Although this leads to a good performance during correspondence matching and pose hypothesis computation, the primitives provide a more fine-grained representation which is more suitable for inlier validation.

In Fig. 18 we show the position error between the estimated pose and the ground truth object pose meaned over all the experiments. As the objects are rotated farther away from the training views, we see an increase in the estimation error, which is expected since features extracted from the training views become occluded. We observe that for the combined case of both high-level feature types, the errors are substantially lower. We see this as a clear indication that features in both domains complement each other, especially

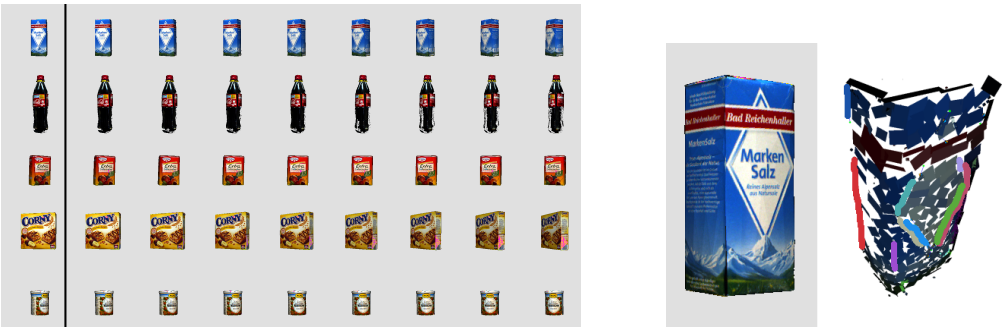


Fig. 17 Left: Example prototype views (leftmost column) and test images, all shown only by left view of the stereo images, for a subset of the tested KIT objects. Right: left view of a virtual stereo input image of the KIT object "BlueSaltCube" (leftmost) and the extracted contours and surflings from the virtual stereo image (rightmost).

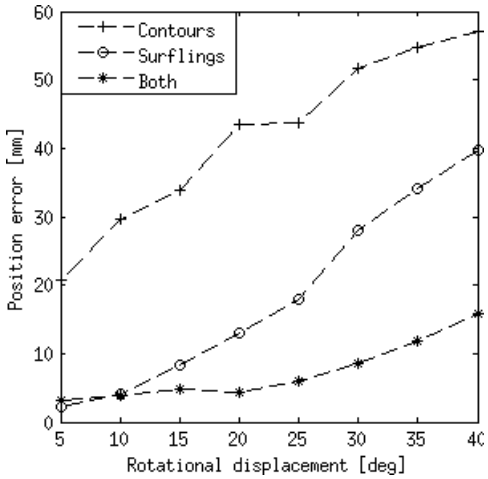


Fig. 18 Ground truth position errors for each angular displacement away from the prototype, meaned over all 112 objects considered.

when dealing with a large range of different objects in terms of shape and appearance.

6 Conclusion

We described a hierarchical Early Cognitive Vision system in which two parallel hierarchies express surface and edge information. This hierarchy provides input to higher level tasks in terms of different visual modalities in 2D and 3D with different amounts of granularity. We have demonstrated the usefulness of such a hierarchical representation for three rather different tasks. This reflects the perspective that features can be shared across tasks and by that computational resources in complex system can be saved as it is done in the occipital areas of the human visual system.

In our approach, we do not make use of any learning for deriving the hierarchy which, when compared to the human visual system, is clearly not a realistic assumption. Instead a lot of 'engineering intelligence' has been used to design features at various levels and for different modalities. By that, system complexity necessarily increases. Recent successes of deep hierarchical neural networks (see, e.g.,[4]) give the perspective to replace such engineering intelligence by learning. It might be, that the bias/variance dilemma formulated by Geman et al. [21] and being responsible for a severe set-back of the idea of neural network as general problem solvers has to be seen in a different perspective when huge amounts of data are available. However, fundamental problems of such generic approaches have also been surfaced recently [80]. Hence it remains an open question how much engineering is required in building up deep hierarchical systems. There is evidence that also in the human visual system, there exist quite an amount of genetic precoding. However, there is also clear evidence for learning, even in rather early areas such as V1 and V2 (see, e.g., [45]). In our future work, we intend to relax some of the hard-wired assumptions by learning. In particular we feel that in our approach we have thrown too much information away by insisting on a symbolic description at a too early stage, for example by using a very condensed edge descriptor which does not express fine differences in appearance.

There exists overwhelming evidence that the human visual system computes a large variety of aspects in the two large areas V1 and V2 in parallel, covering both edge and surface aspect (see, e.g., [45]). That means that at least for local low-level feature processing on the level of V1 and V2, no major selection process seems to take place for reducing computational complexity. Accordingly in our system we do process both kinds of information – surface and edge based – in parallel. However, which of the possible higher order and more global features to use for a certain visual task might well be subject to a selection process. This is related to the problem of attention (see, e.g., [83]). In our pose estimation and grasp approach, we still only use very premature techniques for such selection, where we basically use both kinds of information by brute force combination. However, we are aware that an appropriate modelling of attention to deal with the vast amount of information provided by the early visual areas for different visual tasks is an important issue to be addressed.

On the positive side, we can say that the visual representation is applicable in multiple task contexts and also can be computed fast enough to be applied on robots. However, we admit that we went probably too far with 'designing' instead of 'learning'. Hence an important aspect of future research will be to replace some of the hard-coded aspects and to introduce learning for example to acquire the statistical relationship between visual entities as utilized in, e.g., monocular depth cues, and the learning of higher level entities formed by the lower level entities in the hierarchy. Another aspect is that so far the edge and surface hierarchy are operating independently. However, on a higher level of the visual hierarchy these two representations should be merged into entities that cover both surface as well as edge information. Also for that engineering design would be very cumbersome and extraction algorithms which at least include some learning would be required.

Acknowledgements This work has been supported by the European Community's Seventh Framework Programme FP7/ICT under grant agreement no. 270273, Xperience. We would like to thank Antonio Rodriguez Sanchez for providing an initial version of Fig. 1.

References

1. Başeski, E., Pugeault, N., Kalkan, S., Bodenhagen, L., Piater, J.H., Krüger, N.: Using Multi-Modal 3D Contours and Their Relations for Vision and Robotics. Journal of Visual Communication and Image Representation **21**(8), 850–864 (2010)
2. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (surf). Computer Vision and Image Understanding **110**(3), 346 – 359 (2008). DOI 10.1016/j.cviu.2007.09.014. URL http://www.sciencedirect.com/science/article/pii/S1077314207001555
3. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. Pattern Analysis and Machine Intelligence, IEEE Transactions on **24**(4), 509–522 (2002). DOI 10.1109/34.993558. URL http://dx.doi.org/10.1109/34.993558
4. Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning **2**, 1–127 (2009)
5. Bengio, Y., Lamblin, P., Popovici, P., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems, vol. 19, pp. 153–160 (2007)
6. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
7. Buch, A.G., Jessen, J.B., Kraft, D., Savarimuthu, T.R., Krüger, N.: Extended 3d line segments from rgb-d data for pose estimation. In: Image Analysis, pp. 54–65. Springer (2013)
8. Buch, A.G., Kraft, D., Kämäräinen, J.K., Krüger, N.: Pose estimation using a hierarchical 3d representation of contours and surfaces. In: VISAPP (1), pp. 105–111 (2013)
9. Buch, A.G., Kraft, D., Kamarainen, J.K., Petersen, H.G., Kruger, N.: Pose estimation using local structure-specific shape and appearance context. In: Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp. 2080–2087. IEEE (2013)
10. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **8**(6), 679 – 698 (1986)
11. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: In Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22 (2004)
12. Detry, R., Pugeault, N., Piater, J.: A probabilistic framework for 3D visual object representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(10), 1790–1803 (2009)
13. Dickinson, S.: The evolution of object categorization and the challenge of image abstraction. In: S. Dickinson, A. Leonardis, B. Schiele, M. Tarr (eds.) Object Categorization: Computer and Human Vision Perspectives, pp. 1–37. Cambridge University Press (2009)
14. Felleman, D., Essen, D.V.: Distributed hierarchical processing in primate cerebral cortex. Cerebral Cortex **1**, 1–47 (1991)
15. Felsberg, M., Kalkan, S., Krüger, N.: Continuous dimensionality characterization of image structures. Image and Vision Computing **27**, 628–636 (2009)
16. Felsberg, M., Sommer, G.: The monogenic signal. IEEE Transactions on Signal Processing **49**(12), 3136–3144 (2001)
17. Fidler, S., Boben, M., Leonardis, A.: Learning hierarchical compositional representations of object structure. In: S. Dickinson, A. Leonardis, B. Schiele, M. Tarr (eds.) Object Categorization: Computer and Human Vision Perspectives, pp. 196–215. Cambridge University Press (2009)
18. Fidler, S., Boben, M., Leonardis, A.: A coarse-to-fine taxonomy of constellations for fast multi-class object detection. In: ECCV (5), pp. 687–700 (2010)
19. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
20. Fukushima, K., Miyake, S., Ito, T.: Neocognitron: a neural network model for a mechanism of visual pattern recognition. IEEE Systems, Man and Cybernetics **13**(3), 826–834 (1983)
21. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural Compu-

tation **4**, 1–58 (1995)
22. Geman, S., Potter, D., Chi, Z.: Composition systems. Quarterly of Applied Mathematics **60**(4), 707–736 (2002)
23. Gilbert, A., Illingworth, J., R., B.: Action recognition using mined hierarchical compound features. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(5), 883–897 (2011)
24. Granlund, G.H., Knutsson, H.: Signal Processing for Computer Vision. Kluwer Academic Publishers, Dordrecht (1995)
25. Hetzel, G., Leibe, B., Levi, P., Schiele, B.: 3D object recognition from range images using local feature histograms. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. II–394–II–399. IEEE Computer Society, Los Alamitos, CA, USA (2001). DOI 10.1109/CVPR.2001.990988. URL http://dx.doi.org/10.1109/CVPR.2001.990988
26. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation **18**, 527–1554 (2006)
27. Huang, F.J., LeCun, Y.: Large-scale learning with SVN and convolutional nets for generic object categorization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 284–291 (2006)
28. Hubel, D., Wiesel, T.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Phyiology **160**, 106–154 (1962)
29. Hubel, D., Wiesel, T.: Anatomical demonstration of columns in the monkey striate cortex. Nature **221**, 747–750 (1969)
30. Hummel, J., Biederman, I.: Dynamic binding in a neural network for shape recognition. Psychological Review **99**, 480–517 (1992)
31. Hunt, R.: Measuring Colour. 3rd edition. Fountain Press, Kingston-upon-Thames (1998)
32. Jensen, L.B.W., Kjær-Nielsen, A., Pauwels, K., Jessen, J.B., Hulle, M.V., Krüger, N.: A two-level real-time vision machine combining coarse and fine grained parallelism. Journal of Real-Time Image Processing **5**(4), 291–304 (2010)
33. Jessen, J.B., Pilz, F., Kraft, D., Pugeault, N., Krüger, N.: Accumulation of different visual feature descriptors in a coherent framework. In: Scandinavian Conference on Image Analysis (SCIA), pp. 79–90 (2011)
34. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **21**(5), 433–449 (1999). DOI 10.1109/34.765655. URL http://dx.doi.org/10.1109/34.765655
35. Kalkan, S., Wörgötter, F., Krüger, N.: Statistical analysis of local 3d structure in 2d images. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1114–1121 (2006)
36. Kandell, E., Schwartz, J., Messel, T.: Principles of Neural Science (4th edition). McGraw Hill (2000)
37. Kasper, A., Xue, Z., Dillmann, R.: The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. International Journal of Robotics Research (IJRR) **31**(8), 927–934 (2012). DOI 10.1177/0278364912445831
38. Kavukcuoglu, K., Sermanet, P., amd K. Gregor, Y.B., Mathieu, M., LeCun, Y.: Learning convolutional feature hierachies for visual recognition. In: Advances in Neural Information Processing Systems (NIPS 2010), vol. 23, pp. 1090–1098 (2010)
39. Kellman, P., Arterberry, M.: The Cradle of Knowledge. MIT-Press (1998)
40. Kjær-Nielsen, A., Buch, A.G., Jensen, A.E.K., Møller, B., Kraft, D., Krüger, N., Petersen, H.G., Ellekilde, L.P.: Ring on the hook: Placing a ring on a moving and pendulating hook based on visual input. Industrial Robot: An International Journal **28**(3), 301 – 314 (2010)
41. Kootstra, G., Popovic, M., Jørgensen, J., Kuklinski, K., Miatliuk, K., Kragic, D., Kruger, N.: Enabling grasping of unknown objects through a synergistic use of edge and surface information. The International Journal of Robotics Research **31**(10), 1190–1213 (2012). DOI 10.1177/0278364912452621. URL http://ijr.sagepub.com/content/31/10/1190.abstract
42. Kovesi, P.: Image features from phase congruency. Videre: Journal of Computer Vision Research **1**(3), 1–26 (1999)
43. Kraft, D., Detry, R., Pugeault, N., Başeski, E., Guerin, F., Piater, J., Krüger, N.: Development of object and grasping knowledge by robot exploration. IEEE Transactions on Autonomous Mental Development **2**(4), 368–383 (2010)
44. Kraft, D., Pugeault, N., Başeski, E., Popović, M., Kragic, D., Kalkan, S., Wörgötter, F., Krüger, N.: Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes. Special Issue on "Cognitive Humanoid Robots" of the International Journal of Humanoid Robotics **5**, 247–265 (2009)
45. Krüger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., , Rodríguez-Sánchez, A.J., Wiskott, L.: Deep hierarchies in the primate visual cortex: What can we learn for computer vision? IEEE PAMI **35**(8), 1847–1871 (2013)

46. Krüger, N., Pugeault, N., Başeski, E., Jensen, L.B.W., Kalkan, S., Kraft, D., Jessen, J.B., Pilz, F., Nielsen, A.K., Popović, M., Asfour, T., Piater, J., Kragic, D., Wörgötter., F.: Early cognitive vision as a front-end for cognitive systems. ECCV 2010 Workshop on "Vision for Cognitive Tasks" (2010)
47. Krüger, N., Wörgötter, F.: Different degree of genetical prestructuring in the ontogenesis of visual abilities based on deterministic and statistical regularities. In: Proceedings of the Workshop On Growing up Artifacts that Live (SAB 2002), pp. 5–14 (2002)
48. Krüger, N., Wörgötter, F.: Multi-modal primitives as functional models of hyper-columns and their use for contextual integration. In: Proceedings of the 1st International Symposium on Brain, Vision and Artificial Intelligence, Lecture Notes in Computer Science, LNCS 3704, pp. 157–156 (2005)
49. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1817–1824 (2011)
50. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2169–2178 (2006)
51. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
52. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision (IJCV) **2**(60), 91–110 (2004)
53. Marr, D.: Vision: A computational investigation into the human representation and processing of visual information. Freeman (1977)
54. Mel, B.W., Fiser, J.: Minimizing binding errors using learned conjunctive features. Neural Computation **12**(4), 731–762 (2000)
55. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **27**(10), 1615–1630 (2005)
56. Milner, A., Goodale, M.: Separate visual pathways for perception and action. Trends in Neuroscience **15**, 20–25 (1992)
57. Murray, D., Little, J.: Patchlets: Representing stereo vision data with surface elements. In: Application of Computer Vision. WACV/MOTIONS Volume 1. Seventh IEEE Workshops on, vol. 1, pp. 192–199 (2005)
58. Mustafa, W., Pugeault, N., Krüger, N.: Multi-view object recognition using view-point invariant shape relations and appearance information. In: IEEE International Conference on Robotics and Automation (ICRA) (2013)
59. Niebles, J., Fei Fei, L.: A hierarchical model of shape and appearance for human action classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
60. Olesen, S.M., Lyder, S., Kraft, D., Krüger, N., Jessen, J.B.: Real-time extraction of surface patches with associated uncertainties by means of kinect cameras. Journal of Real-Time Image Processing pp. 1–14 (2012). DOI 10.1007/s11554-012-0261-x. URL `http://dx.doi.org/10.1007/s11554-012-0261-x`
61. Ommer, B., Buhmann, J.M.: Learning the compositional nature of visual objects. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
62. O'Neill, B.: Elementary Differential Geometry. Elsevier Academic Press (2006). URL `http://books.google.dk/books?id=OtbNXAIve\_AC`
63. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
64. Pinto, N., Barhomi, Y., Cox, D., DiCarlo, J.: Comparing state-of-the-art visual features on invariant object recognition tasks. In: IEEE Workshop on Applications of Computer Vision (WACV 2011), pp. 463–470 (2011)
65. Pinto, N., DiCarlo, J., Cox, D.: How far can you get with a modern face recognition test set using only simple features? In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2591–2598 (2009)
66. Popović, M., Kraft, D., Bodenhagen, L., Başeski, E., Pugeault, N., Kragic, D., Asfour, T., Krüger, N.: A strategy for grasping unknown objects based on co-planarity and colour information. Robotics and Autonomous Systems **58**(5), 551 – 565 (2010). DOI DOI:10.1016/j.robot.2010.01.003
67. Pugeault, N., Wörgötter, F., Krüger, N.: Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics. International Journal of Humanoid Robotics (Special Issue on Cognitive Humanoid Vision) **7**(3), 379–405 (2010)
68. Quack, T., Ferrari, V., Leibe, B., Gool, L.V.: Efficient mining of frequent and distinctive feature configurations. In: Proc. of the International Conference in Computer Vision (ICCV), pp. 1–8 (2007)
69. Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: IEEE CVPR Workshop on DeepVision (2014)
70. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience **11**(2), 1019–1025 (1999)

71. Rosenblatt, F.: The Perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review **65**, 386–408 (1958)
72. Rumelhart, D., Hinton, G., Williams, R.: Learning representation by back–propagating errors. Nature **323**(9), 533–536 (1986)
73. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pp. 3212–3217. IEEE (2009)
74. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, pp. 3384–3391 (2008)
75. Savarese, S., Winn, J., Criminisi, A.: Discriminative object class models of appearance and shape by correlatons. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2033–2040 (2006)
76. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., Le Cun, Y.: OverFeat: Integrated recognition, localization and detection using convolutional networks. In: International Conference on Learning Representations (2014)
77. Sermanet, P., LeCun, Y.: Traffic sign recognition with multi-scale convolutional networks. In: Proceedings of International Joint Conference on Neural Networks (IJCNN'11), pp. 2809–2813 (2011)
78. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Object recognition with cortex-like mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(3), 411–426 (2007)
79. Sutskever, I., Hinton, G.E.: Learning multilevel distributed representations for high-dimensional sequences. In: AI and Statistics, pp. 544–551 (2007)
80. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. International Conference on Learning Representations (2014)
81. Tenenbaum, J.B., Kemp, C., Griffiths, T.L., Goodman, N.D.: How to grow a mind: Statistics, structure, and abstraction. Science **331**, 1279–1285 (2011)
82. Tsotsos, J.K.: Analyzing vision at the complexity level. Behavioral and Brain Sciences **13**(3), 423–469 (1990)
83. Tsotsos, J.K.: A Computational Perspective on Visual Attention, 1st edn. The MIT Press (2011)
84. Ullman, S., Epshtein, B.: Visual classication by a hierarchy of extended fragments. In: Towards Category-Level Object Recognition., pp. 321–344. Springer-Verlag (2006)
85. Wahl, E., Hillenbrand, U., Hirzinger, G.: Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification. In: 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on, pp. 474–481. IEEE (2003)
86. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: Proc. of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 270–279 (2010)
87. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision **73**(2), 213–238 (2007)