FP6-004381-MACS

MACS

Multi-sensory Autonomous Cognitive Systems Interacting with Dynamic Environments for Perceiving and Using Affordances

Instrument: Specifically Targeted Research Project (STReP)

Thematic Priority: 2.3.2.4 Cognitive Systems

D3.1.1 Top-down and bottom-up symbol grounding

Due date of deliverable: August 31, 2005
Actual submission date: October 10, 2005

Start date of project: September 1, 2004 Duration: 36 months

Joanneum Research (JR_DIB)

Revision: Version 1

Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

EU Project 

Deliverable D3.1.1

Top-down and bottom-up symbol grounding

Lucas Paletta, Gerald Fritz, Manish Kumar, Joachim Hertzberg, and Frank Schönherr

Number: **MACS/3/1/1**
WP: *3.1*
Status: *draft, version 3*
Created at: *September 18, 2004*
Revised at: *October 11, 2005*

Top-down and bottom-up symbol grounding

FhG/AIS	Fraunhofer Institut für Autonome Intelligente Systeme, Sankt Augustin, D
JR_DIB	Joanneum Research Graz, A
LiU-IDA	Linköping Universitet, Linköping, S
METU-KOVAN	Middle East Technical University, Ankara, T
OFAI	Österreichische Studiengesellschaft für Kybernetik,Vienna, A

This research was funded by the European Commission's 6th Framework Programme IST Project MACS under contract/grant number FP6-004381. The Commission's support is gratefully acknowledged.

Author addresses:

Lucas Paletta, Gerald Fritz, and Manish Kumar
Joanneum Research
Institute of Digital Image Processing
Computational Perception (CAPE)
Steyrergasse 9
A-8010 Graz, Austria

Fraunhofer Institut für
Autonome Intelligente Systeme
Schloss Birlinghoven
D-53754 Sankt Augustin
Germany

Tel.: +49 (0) 2241 14-2683
(Co-ordinator)

Contact:
Dr.-Ing. Erich Rome

Joanneum Research
Institute of Digital Image Processing
Computational Perception (CAPE)
Steyrergasse 9
A-8010 Graz
Austria

Tel.: +43 (0) 316 876-1769

Contact:
Dr. Lucas Paletta

Linköping Universitet
Dept. of Computer and Info. Science
Linköping 581 83
Sweden

Tel.: +46 13 24 26 28

Contact:
Prof. Dr. Patrick Doherty

Middle East Technical University
Dept. of Computer Engineering
Inonu Bulvari
TR-06531 Ankara
Turkey

Tel.: +90 312 210 5539

Contact:
Prof. Dr. Erol Sahin

Österreichische Studiengesellschaft
für Kybernetik (ÖSGK)
Freyung 6
A-1010 Vienna
Austria

Tel.: +43 1 5336112 0

Contact:
Prof. Dr. Georg Dorffner

Contents

1 Executive Summary

This deliverable report aims at presenting a complete overview on the ontology, the representations and the functions that will be used for perception in the MACS-specific scenarios. It presents a framework that provides a hierarchical structure on representations and processing, that can be easily extended with any future developments concerning the detection, attention, the monitoring and the tracking of entities under investigation by the robotic system.

The goal is to build a *MACS computational perception toolbox* that will incorporate the processing of camera and other sensor based information. This report gives a framework, i.e., a logical structure to relate entities under observation, an ontology in terms of a hierarchical representation and the related definitions, and the functions to operate on the attributed representation in order to provide a consistent methodology on perception, as a basis for affordance perception.

We first outline an ontology of the entities under study, the representation structure and the functionality, in order to pinpoint the notions and principles that underlie all further developments on perception. Secondly, we give an overview on the state of the art in the MACS representation structure in terms of giving an outline of the feature vector hierarchy that builds up from simple features characterising points to complex object models about the individual physical entities.

This report is supposed to represent a 'living document', providing a first starting point which will become augmented from developments on representation and functionality during the course of the MACS project.

2 Symbol Grounding

According to [Harnad, 1990], representing affordances is related to the symbol grounding problem in terms of embodying symbol systems [Newell, 1980], i.e., how are a systems internal symbols causally connected to the external objects, events, and relations they are supposed to represent. In robotics, this is solved for special instances like in localisation, which means implicitly grounding a symbol current-pose in sensor data. In general, it is unsolved, but is receiving continuous attention [Special Issue:, 2003], [Cangelosi et al., 2002]. Current work focuses on object anchoring (where symbols represent individual objects) [Coradeschi and Saffiotti, 2001]. Having affordances as first-class objects of perception offers a new way of approaching the symbol grounding problem as affordances have by definition a perception as well as a symbol and an action facet. It also allows the dual of object anchoring, action anchoring, to be addressed.

In the following Sections, we approach symbol grounding from two different viewpoints. Firstly, *bottom-up symbol grounding* is starting from raw feature sampling, providing a number of reference feature vectors that represent key prototypes of the underlying sample distribution. These prototypes provide meaning from the viewpoint of the experienced data distribution. Bottom-up information processing dependencies are reflected by the implemented feature vector hierarchy (Sections 6.7-6.8). From another point of view, *top-down symbol grounding* is tackled from *supervised* determination of prototypical reference features that structure visual perception in a manner that is meaningful either from the viewpoint of the engineer, or the cost function of the corresponding machine learning

methodology - e.g., the entropy function in decision tree generation, described in Section 6.9 for top-down texture feature determination.

3 Ontology

In the following Sections we outline a framework for the description of perceiving individual entities in the physical world, for representing their characteristic attributes, and for the associated functions to attend, detect, track and monitor these entities over space and time.

This Section provides a brief ontology about the key phenomenological entities regarding the perception in MACS specific scenarios that are underlying the perceptual representations in the *MACS Computational Perception toolbox*. We do not claim that this ontology will propose general purpose definitions, but in contrast understand it as a basis for further investigations within the research on affordance perception. In addition, these entities have to be related to key functions in the context of affordance perception (Section 5) and further refer to the representation (Section 4) selected for the existing and the forthcoming implementations (Section 6) chosen as parts of the *MACS Computational Perception toolbox*.

A first short Section 3.1 describes now the implicit relation between entities, representation and functions, and will hereby introduce a more detailed ontology on the different kinds of entities under investigation presented thereafter in Sections 3.2-3.6.

3.1 Entities, Representations, and Functions

In this ontology we refer to entities in the physical world that are observed by decision making agents, and that are related to perceptual representations of these agents in general. We will refer here to aspects in the real world in terms of *points*, *regions*, *grouped regions*, *objects* and *object configurations*, i.e., a scene.

Entities are phenomenological objects in the real world that can be described via feature values of attributes in representations of underlying object models. Entities and associated features are structured in a hierarchical manner, i.e., more global entities or representations are built by integrating lower level entities and representations, respectively (Fig. 1). In a similar way, the use of functions is organised in hierarchical dependency, calling lower level operations (e.g., *detectEntity*) from a more global level of observation (e.g., *trackEntity*).

The entity is the physical reality underlying any observation, while the feature will only capture an attribute value of the representation of the observation itself - which may be noisy, ambiguous and erroneous in general. In this sense, the feature can be understood either (i) just as a hypothesis on the projection of the entity being in real world onto the observer's sensors (*hypothetical realist's* point of view on perception), or (ii) as a matter of perceptual fact without the possibility to ever proof in an 'objective way' whether there is an entity behind it or not (*constructivist's* point of view on perception). However, in our framework we maintain the notion of an entity since we need to consider hypotheses about objects that would exist in a strict sense together with associated confidences, in order to pursue the observations with respect to a common reference.

Definitions on the entities under investigations are found in detail in the following Sections. The features of the associated representations will be outlined in detail in Section 4.

Finally, the functions are described in Section 5.

3.2 Point Entity

We define a *point* simply as a selected single entity in real world that can be described in terms of singleton measurements. In general, this might be any 3D point in space, but it might at the same time refer to a 2D point on a surface of a real world object. However, this point can be characterised in various ways, e.g., in terms of a single pixel's value (referring to *point features* in Section 4.1), or in terms of a pattern that is centered at this point's (sub-)pixel position in the image array (referring to *region features* in Section 4.2).

3.3 Region Entity

We define a *region* to be any spatially extended and inherently *connected* part of the real world environment (in some sense, referring to a set of point features) that can be uniquely determined applying appopriate algorithms, and therefore be discriminated from its surrounding environment. Regions might be associated just to any area of a real world's surface, or even to the surface being part of multiple objects in the environment.

3.4 Grouping Entity

A *grouping* is defined as a configuration of areas in the real world that are associated to each other, e.g., by a spatial, i.e., geometric relation. One can imagine relations between areas of different modality of observation, e.g., relating a surface observation based on colour with another pattern based observation using texture based features. The grouping actually involves at least a region-to-region relation, it can be determined in terms of any spatio-temporal dependency - e.g., either through direct observation, or in a statistical sense.

3.5 Object Entity

An object is usually defined as a physical unity with extended perceptual constancy with respect to its consistent appearance over space and time, i.e., it can be perceived as a unity in spatiotemporal feature space. An object can be observed from various viewpoints in 3-dimensional space, assuring a deterministic sequence of view appearances when being moved or by moving autonomously. This is according to the *classical object notion* as it has been proposed in text books on computer vision during the last decades.

In another sense, the notion of an object structures perception in a way that it relates to a configuration of grouping entities that may either frequently occur or be of relevance either for discrimination or for prediction purposes in a task under investigation. This is according to an *advanced object notion* that is also related to the context of affordances: In the *affordance based object notion*, object entities refer to characteristic groups of individual groupings, each of which are typical cues to specific functionalities that are in relation with the physical entity that is attached with the grouping based cue.

One of the major goals of affordance perception is to investigate the *affordance based object notion* and compare the results with those produced with the *classical object notion*.

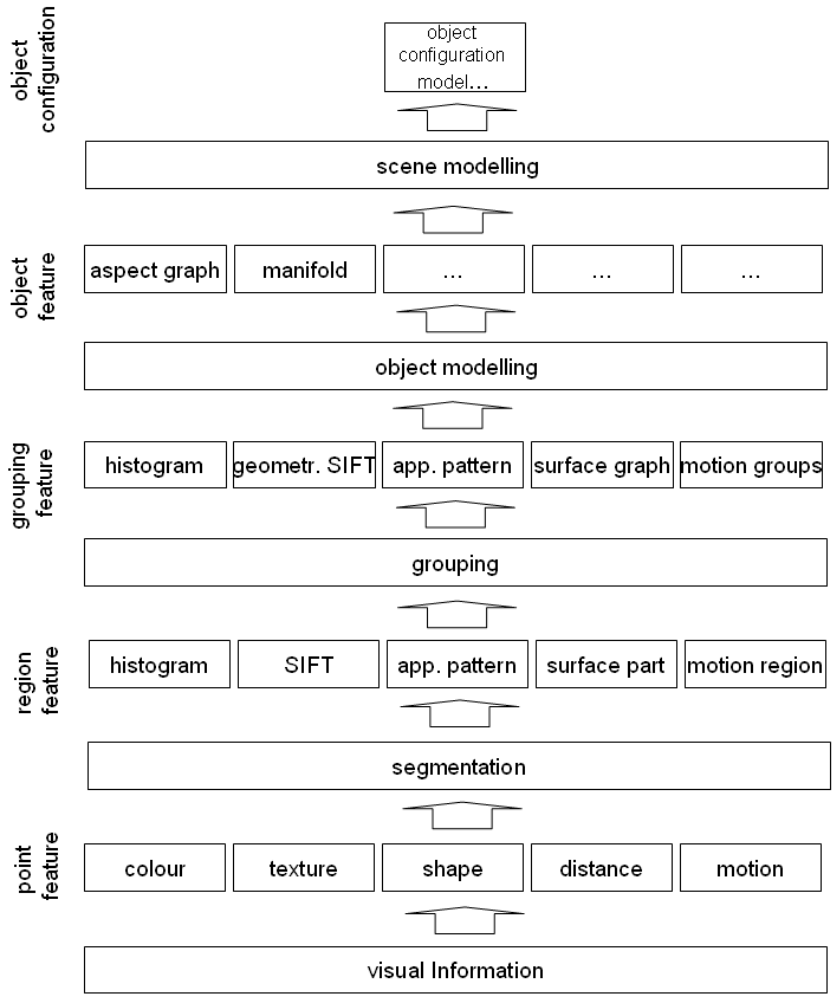


Figure 1: Hierarchy of entities and representations in visual information processing.

3.6 Object Configuration Entity

Objects are always perceived within a surrounding that can be interpreted as well in terms of object entities or as an integration of any other lower level entities that are in a characteristic relation to each other. Object configurations denote therefore in particular the specific relations (spatially, geometrically, temporally, etc.) between the individual entities. This configuration is perceived in a spatio-temporally consistent manner, i.e., entities that are related to other entities within the configuration must (in a statistical sense) 'always' be found to be dependent to the other entity in the specifically described manner.

4 Representations

We consider now representations and attributes with respect to the entity definitions stated above. These representations are per se data structures that present feature values

from the observations about an entity under investigation. While we may assoiate various representations to a specific entity, it should be clear that these may finally contribute to different results and beliefs, however, one should be able to eventually integrate them to a fused description and belief state.

Representations may not only hold feature values of observations about partial, selected aspects of the real world. We will also make use of *symbolic representations* in a sense of representing characteristic aspects of extracted descriptions using *prototypical references* and a measure of uncertainty, e.g., using probabilistic, evidence theoretic or fuzzy measures for quantifying uncertainty in the agent's belief.

4.1 Point Features

Point features are simply related to feature attributes that characterise a point of (sub-)pixel accuracy within the measurement array. We intend to discriminate between the following associated measurement modalities,

- colour (e.g., using RGB or any related colour space representation),
- orientation (e.g., using Gabor or wavelet filter analysis),
- brightness (using the raw brightness value),
- distance (using laser range or stereo based 3D information recovery),

that may characterise a point observation.

4.2 Region Features

Region features characterise the perception in terms of an area of connected image pixels - either in terms of patterns or of any kind of blob based region perception. E.g., the regions might be extracted by descriptor operators (SIFT [Lowe, 2004], MSER [Matas and Obdrzalek, 2004], etc.) or by histograms that characterise the region with respect to specific point features. Region features can be determined either by bottom-up information processing - e.g., in region growing - starting to build up information about an area of interest without any predefined knowledge - or by top-down information processing, by focusing on a specific region within an attention window (see Deliverable D3.1.2) that has been determined by means of task or user specific knowledge.

4.3 Grouping Features

Grouping features are characterised by capturing the attribute values of the relation between different regions to each other - either by geometrical relations, or, e.g., by including other regions into the own spatial support. Typical grouping representations are

- Vector based feature-geometry representations (capturing feature attriutes and geometrical relations),
- Graph based representations (edges between features represent spatial relations),
- Sensorimotor, i.e., feature-action representations, i.e., features and directional activities integrated into vector representation [Paletta et al., 2005].

4.4 Object Features

Object features have to take into consideration that physical objects require the representation of different views, to model the transition between various aspects, and the discrimination between different object hypotheses. Typical representations for object features include

- Manifold representations (continuous transitions between views are represented by trajectories in feature space).
- Graph representations (edges between aspect views represent spatial relations).

4.5 Object Configuration Features

Object configurations can be represented either by relations in space, or by transitions between object type responses.

5 Functions

Basic functionalities are required in order to support the succesful perception of entities. On the one hand the toolbox must first provide capabilities to determine an observation vector that on the one hand includes raw signal monitoring, on the other hand integrates feature capturing of any kind of entities in the field of observation. Monitoring entities therefore implicitly involves detection of entities of any kind. The detection might be constraint by any attentive selection of features, i.e., parametrising the search for entities in the field of observation by some top-down information processing, e.g., by setting weights to bias color processing in the attention mechanism. Finally, detected entities should be tracked over time while decision making will be involved to decide about whether the change in the appearance would still support the entity model or whether the tracking should be abandoned.

5.1 Monitoring of Entities

The most basic functionality that should be provided is to extract a feature vector from the observation of a predetermined area of interest, or a specific level of entity according to the users needs.

The monitoring process is therefore set using the function

monitorEntities(Rect BBox, Entity AttEntity, Param AttParam, Time tWindow)

in its general form, where *Bbox* denotes the rectangular bounding box that designates an area of interest, *AttEntity* denotes the specific entity (entities) under observation (might also be a vector of entities that should be observed), and *tWindow* determines a window of observation in time. *AttParam* is a parameter vector that is used for the attention method to focus processing on highly specific features and attribute values within the perceptual information stream. Output of the function is time series of feature values in a time-stamped list of feature vectors for further analysis in spatiotemporal context. *monitorEntities()* is executing *detectEntities()* for the actual extraction of entity instances,

attendEntity() for applying the focus of attention on a pre-specified parameterisation, and *trackEntity()* in order to be capable to follow the appearance of an entity over time.

5.2 Detection of Entities

The existence of an entity in the perceptual array must be verified in terms of applying algorithms that are particularly tuned to detect the specific kind of entity under investigation. However, invoking high level entity detectors will use lower level detectors and therefore include all levels of processing and detecting - but also involving the complete level of complexity associated to this complex search strategy. The functionality for detecting entities can therefore be parameterised to focus on the extraction of specific types of entities, such as, points, patterns, or groupings in the field of observation.

The detection process is invoked using the function

detectEntity(Rect BBox, Entity AttSingleEntity, EntityParam AttParam)

This involves the extraction of a specific kind of entity *AttSingleEntity* within an area of interest *Bbox*. An attention parameter vector (*AttParam*) can be provided to receive only results of a specific attribution, or to guide search along the extraction of these parameter values. As noted before, this function can be executed within the *monitorEntities()* functionality for the actual extraction of specific entities according to the attention parameterisation.

5.3 Attention to Entities

Attention on relevant perceptual information is highly important to restrict the processing time in monitoring the analysis of the sensor information stream, on the other hand, restricting the complexity is at the same time a valid compression and discrimination model that can be used to structure any access and search strategy involved in organising the existing knowledge of the perceiving and acting cognitive agent.

Attending to a selective partition of the incoming stream of information is therefore modeled by the function

attendEntity(Rect BBox, Param AttParam)

which applies a selective attention methodology on the perceptual input stream. *BBox* is spatially restricing the input information, while *AttParam* may include the parameterisation of various feature and search attributes in general in order to tune and constrain the entity search. This parameterisation could contain 'saliency maps' that weight each pixel of the input information array with a factor that reflects the relevance of feature response values with respect to the user's task requirements.

5.4 Tracking of Entities

The tracking of entities over time requires not only the invocation of a detection operation but also the tracking of the resulting attributes plus the localisation over time. In addition, a decision making is involved since the appearance - i.e., the extracted feature attributes

that were initially associated to the entity - must be updated and checked whether it should still represent the entity or whether the tracking process must be terminated due to an insufficient support of the entity features.

The tracking function is applied using the formalism

trackEntity(Rect BBox, Entity TrackEntity, Features StartFeatures, Param AttParam)

where *BBox* denotes a bounding box for monitoring the entity, *trackEntitity* is the entity under investigation (can be set after a detection operation), *StartFeatures* is the associated feature set received from the detection event, and *AttParam* can be a parameter set for further maintaining a focus of attention on the tracking process.

6 Implementation: A Feature Vector Hierarchy

This Section provides an overview on representations that have been determined as becoming part of the *MACS Computational Perception Toolbox*. As this report is defined to be a living document, it will become extended during the course of the MACS project, i.e, the presented methods form the state-of-the-art in the development of this toolbox.

We will present region descriptors of different kinds, outlining *feature based* and *symbolic representations* on the basis of pattern and histogram descriptions. In addition, we present several grouping features, e.g., on the basis of patterns that are in geometrical relation with each other, and using multi-channel representations.

This Section will start with amotivation for the selected feature repesentations. The individual descriptors, histograms, symbolic representations, and grouping feature are then described in the sequel.

6.1 Motivation

Affordance perception from visual cues needs first a set of useful features and second a representation of different affordances based on that set. In the next Sections we will describe this set is described in more detail. Starting with some basic features, a complete feature vector hierarchy is developed by combining these features into more and more complex descriptions. The representation of affordance cues is then described in terms of these complex descriptions.

6.2 Overview on Region Representations

Low level visual features that we use for region representations are, firstly, the 'Scale Invariant Feature Transform' (SIFT) features proposed by Lowe [Lowe, 2004], and, secondly, colour histograms as described in [Swain and Ballard, 1991].

The SIFT algorithm is an interest point detector with a descriptor that uses the gradient information in a local neighbourhood around the interest point. Colour histograms are calculated within a predefined attention window (see also: MACS deliverables D3.1.2 and D3.1.4). The colour information in the specified sub-window is used to generate the histogram. These kind of features are static, they are extracted out of only one image in terms of a snapshot of the actual scene.

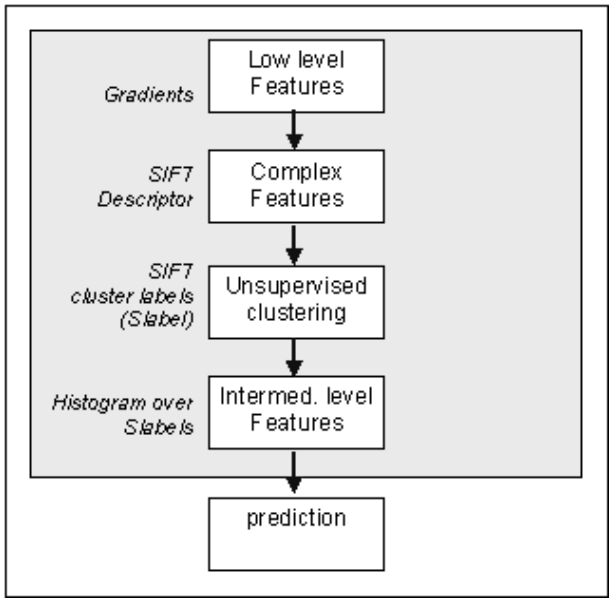


Figure 2: Bottom-up information processing in a hierarchy of representations (simple, complex in terms of region - feature based and symbolic - representations).

In order to deal with dynamics in the scene (i.e., motion detection and grouping, etc.) we use KLT features proposed [Tomasi and Kanade, 1991; Shi and Tomasi, 1994]. These dynamic features require less much computation time than SIFT or colour histograms.

Other relevant low level feature descriptors, planned or under development in the project are, the 'Scale and Affine Invariant Interest Point Detector' [Mikolajczyk and Schmid, 2004] and the 'Maximally Stable Extremal Regions' [Matas et al., 2002; Matas and Obdrzalek, 2004].

In the following we present symbol grounding from a bottom-up as well as from a top-down perspective of information processing. Bottom-up symbol grounding can be presented in the way of determining symbols via unsupervised clustering methods (Section 6.7). Top-down symbol grounding is applied when using supervised learning techniques as in the example of decision tree based classification learned from the presentation of labeled image training data (Section 6.9) to form classifiers for rectangular and circular regions in the attention window.

Given an image or image stream from the cameras, features are extracted in order to process these features through different levels of complexity as shown in Fig. 2 for SIFT descriptors. Clustering is performed to, firstly, generalize and, secondly, group similar descriptors together. Each cluster center can therefore be seen as prototype of some typical image structure, or for the case of colour histograms, as prominently coloured region.

The following Sections describe in detail how the algorithms are parameterised and implemented, in order to get the results described in the deliverable D3.1.2 and D3.1.4 on affordance recognition and detection experiments.

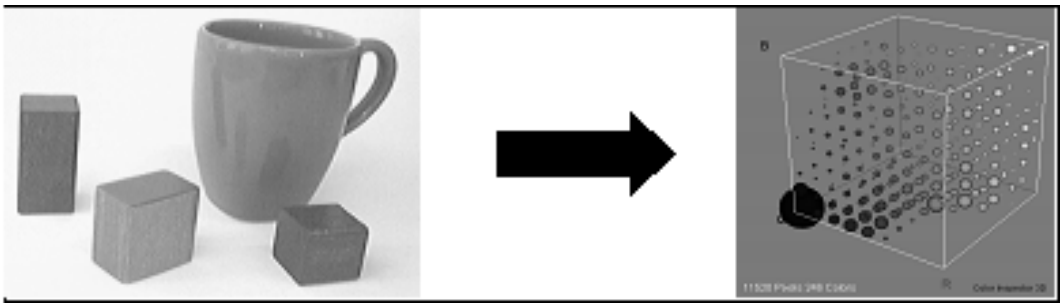


Figure 3: Color histograms from color images.

6.3 Colour Histograms

Colour Histograms provide a robust and efficient cue for recognising regions and objects, in particular when the illumination conditions are under control. They even provide stable object representations in the presence of occlusion and over changes in view, and they can differentiate among a large number of objects. It is shown in [Swain and Ballard, 1991] that these features are invariant to translation and rotation about the viewing axis, and only slowly change under changes view angles, scale, and degrees of occlusion.

Given a discrete colour space defined by some colour axis (e.g., red , green, blue), the colour histogram is obtained by discretizing the image colour and counting the number of times each discrete colour occurs in the image array (a sub-window of predefined size). Fig. 3 shows the principle of the algorithm assuming the colour histogram (right side) is calculated over the whole image on the left side.

The Colour Histogram algorithm takes a sub-window of size 8×8 pixels from the input image and shifts the sub-window with a step size of 4 pixels (in both x and y direction) over the image in a scan line principle from left to right. The colour axis used for the histograms were taken from the RGB colour space. Each axis was divided into 8 equal bins. This gives a feature vector,

$$\varphi = \{f_1, \dots f_\Gamma\} \tag{1}$$

where $\Gamma = 512$, and each f_i is referring to a bin in the histogram representation. The histograms are calculated over the whole image or with respect to a predefined attention window. The different strategies are described in more detail in deliverable D3.1.2.

6.4 Scale Invariant Feature Transform (SIFT) Features

Recently the use of SIFT features has become popular in the computer vision community. The keypoints (points in the image and their descriptors) have been shown to be invariant to image rotation and scale and robust across a substantial range of affine distortion, addition of noise, and change in illumination. Large number of keypoints can be extracted from typical images, which leads to robustness in recognition. One drawback of the algorithm is the comparably still high computational cost in contrast to other methods, although SIFT features are also used for tracking [Kamath et al., 2005].

Fig. 4 shows the major stages of the algorithm from left to right: The first stage is the detection of interest points in the scale space in order to be invariant to scale changes. These points are then refined to sub-pixel accuracy and gradients are calculated in the local

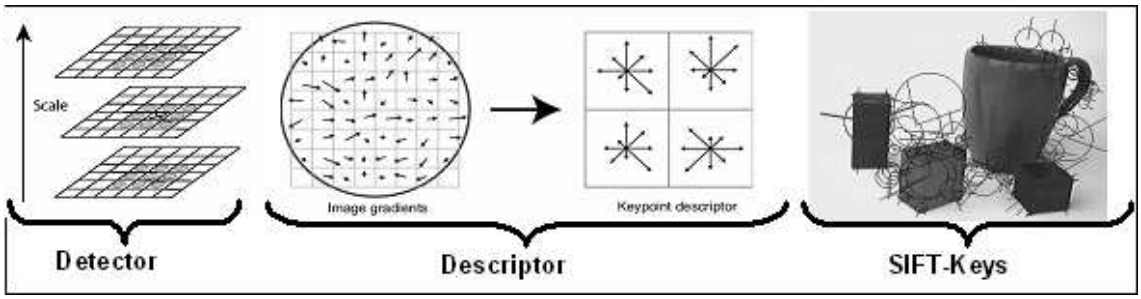


Figure 4: Stages of *feature based* SIFT descriptor processing.

neighbourhood of each point location. The local gradients are extracted at the selected scale in order to extract the most prominent orientations. These orientations are then transformed into a representation that allows for significant levels of local shape distortion and changes in illumination. The extracted keypoints are shown in Fig. 4 in terms of red circles with the major orientation shown as line originating from the circle centre. Note that one keypoint location can result in two or more descriptors if more than one prominent orientation is found. The resulting descriptor has the following form

$$\varphi = \{f_1, \dots f_\Phi\}, \tag{2}$$

with $\Phi = 128$, and each f_i referring to a bin in the orientation histogram representation. Further detail of the algorithm can be found in [Lowe, 2004].

Our implementation of the algorithm uses the suggested doubling of the input image, from 640×480 pixel to 1280×960 pixel with an anti-aliasing $\sigma = 0.5$. The number of Difference of Gaussian images is two (s=2 in the terminology of Lowe), and the edge ratio is less restrictive. The latter gives a greater number of keypoints per image,especially on lower scales.

Principle Component Analysis (PCA) is a well known technique for dimensional reduction. In PCA every new feature can be seen as linear function of the old features. They construct a lower dimensional linear subspace that best explains the variation of these features from their mean. This method is a classical technique from statistical pattern recognition. In contrast to [Ke and Sukthankar, 2004] where PCA is performed on gradient image patches in x and y direction, here the descriptor is built up as Lowe proposed (a kind of histogram over the orientation patch) and as a post processing step these descriptor is projected into the PCA eigenspace of dimension N.

Our implementation takes every SIFT descriptor (of dimension $\Phi = 128$) from a training dataset and builds up a PCA eigenspace. The first 40 dimensions (N=40) are taken into the feature vector, which is a sufficient number to achieve accuracy in reconstruction. Then the feature vector is built as follows,

$$\varphi = \{f_1, \dots f_\epsilon\}, \tag{3}$$

with $\epsilon = 40$ being the Eigendimension, and each f_i is referring to a value of a specific eigenvector dimension.

6.5 KLT Features

Static image features are not likely to be sufficient for dynamic environments since useful feature hierarchies have to deal with motion. Tracking parts or objects in the scene can be useful for many tasks of a robot vision system.

It requires an algorithm to track the motion of features in an image stream. Given the small inter-frame displacement made possible by the factorisation approach, the best tracking method turns out to be the one proposed by [Tomasi and Kanade, 1991]. The method defines the measure of match between fixed-size feature windows in the past and current frame as the sum of squared intensity differences over the windows. The displacement is then defined as the one that minimises this sum. For small motions, a linearization of the image intensities leads to a Newton-Raphson style minimization.

The descriptor of one KLT feature has the following form:

$$\varphi_{n,t} = \{f_1, \dots f_\kappa\}, \tag{4}$$

with $\kappa = 64$, and $\varphi_{n,t}$ denotes the n-th feature out of the whole feature set, while t signals the timeframe when this feature was tracked, and f_i denotes one grey value in the 8×8 neighbourhood around the feature location.

Additional improvements are given by selecting initial features from images where some motion event happens. The same strategy is used to gain additional features during the tracking process when features get lost in the image due to occlusion or other conditions (see deliverables D3.1.2 and D3.1.4).

6.6 Descriptors on Distance Information

It is planned to extract distance information about physical objects from, e.g., laser range or stereo vision based distance estimates, and to analyse the distance information pattern thereafter in terms of region features. E.g., we may apply the SIFT feature method on these patterns, or extract histograms of distance derivatives, etc., in order to determine regions of interest on surface information.

6.7 Bottom-up SIFT Symbol Grounding

In contrast to generalization as shown in the last paragraph, clustering similar features together to form references in the manner of *prototypes* is an alternative approach. This can be done by different clustering techniques, supervised and unsupervised ones. The k-means cluster algorithm is one method to group features together. The major disadvantage of the algorithm is that the number of clusters is one parameter, and should be known beforehand. This can be done by (i) having prior knowledge of the dataset or (ii) by trying to vary the number of clusters until a performance measure is optimised. The procedure of clustering can be seen as generating N different SIFT descriptor classes, similar features are grouped together in one specific class with class-prototypical reference vector $\mathbf{F}_\pi$.

In our experiments, the original SIFT descriptors (128 dimensions) from the trainings dataset were first projected into an N dimensional PCA eigenspace (in our experiments N=40) and then clustered, e.g., into 200 cluster (k=200) specific sample groups. Every new descriptor then is matched against these prototypical cluster centers and the feature vector is represented by the label of the associated reference vector as shown in Fig. 5.

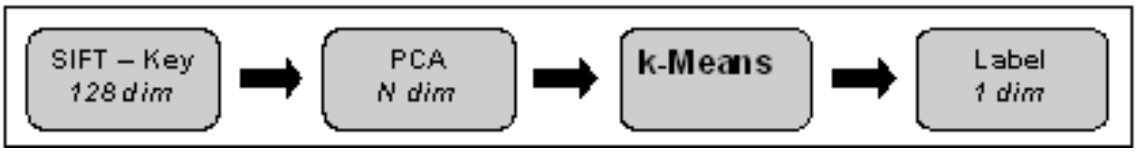


Figure 5: Stages of *symbolic representation* based SIFT descriptor processing.

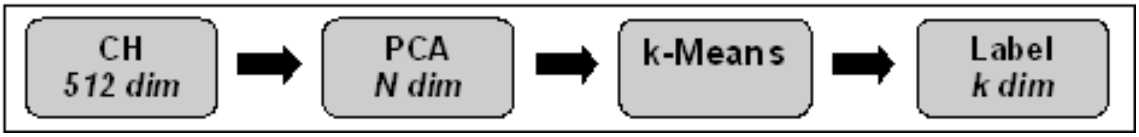


Figure 6: Clustering of color histograms in Eigenspace.

6.8 Bottom-Up Colour Symbol Grounding

Colour histograms are quite sparse when the sub-window they are calculated on is small, like in our implementation. This can be shown by performing PCA on those samples. Only a small number of principal components or basis vectors have a sufficient large eigenvalue. This important property of colour histograms is used here to group similar regions in the feature space together.

The procedure is shown in Fig. 6 (from left to right): From a given training dataset first every colour histogram is projected, i.e., into an 10 dimensional ($N = 10$) PCA eigenspace. The projected dataset is then clustered using the k-means algorithm into k different clusters ($k = 15$ in our experiments). The cluster centers are then reference prototypes for assigning labels to samples out of any test set.

6.9 Top-Down SIFT Symbol Grounding

A model for for identifying features with typical properties determined by the human operator can be developed by supervised clustering techniques, training a classifier on a specific subset of the trainings dataset. In our implementation, we determined dataset partitions by selecting objects with either rectangular or circular regions. The features extracted out of those images correspond then to the specific properties of the trainings subset. Prior knowledge about the object structure is used to distinguish between features which are more likely to be present on one of the subsets.

The SIFT descriptor is based on gradients in the neighbourhood of the key point location. Therefore one can interpret the descriptor to represent the local structure or shape of the underlying image structure. Using the knowledge about present objects in the scenario we subdivided the trainings set into images containing rectangular and circular objects. This is equivalent to annotating every descriptor whether it is related to more rectangular or circular objects. This is the general two class problem and a classifier was learned to distinguish between those classes.

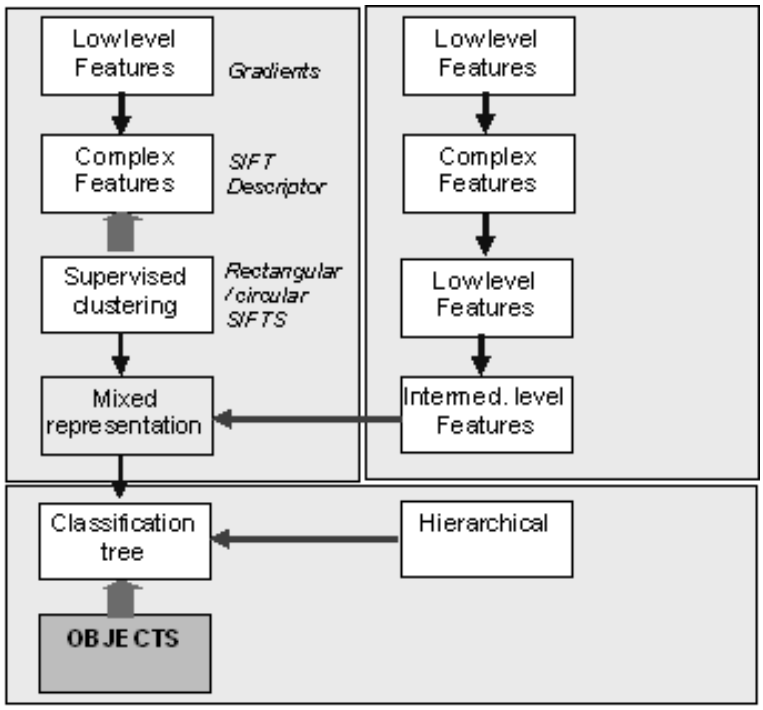


Figure 7: Bottom-up and top-down processing paths in feature vector hierarchy.

6.10 Grouping Features

6.10.1 Geometric SIFT Descriptors

Simple features at a specific location in the image as well as descriptors representing larger structures at a location can not necessarily be sufficient to identify objects, their parts or an informative configuration in the image scene. Combining several simple descriptors under a gemetric relation can encode more complex structures (e.g., [Weber et al., 2000; Paletta et al., 2005]).

Combining n- tuples of equal descriptors is one possible way to end up with more complex features. An arbitrary large amount of tuples of equal descriptors can be combined in order to get a feature chain with geometric constraints. A good choice is the angle between two feature locations because of its invariant properties against scale and rotation. The complete feature vector is outlined as follows,

$$\varphi = \{f_1^1 \alpha_1 f_1^2 \dots f_k^1 \alpha_k f_k^2\} \tag{5}$$

where f_1^1 is the first feature of the first tuple and f_1^2 is the second descriptor - both combined by an angle α_1 between the first and the second pattern center of reference, etc.

In our experiments we generated one tuple by combining two SIFT keys. The angle between those two was calculated and discretised into eight principal directions (Fig. 8 shows a sample geometric SIFT configuration).

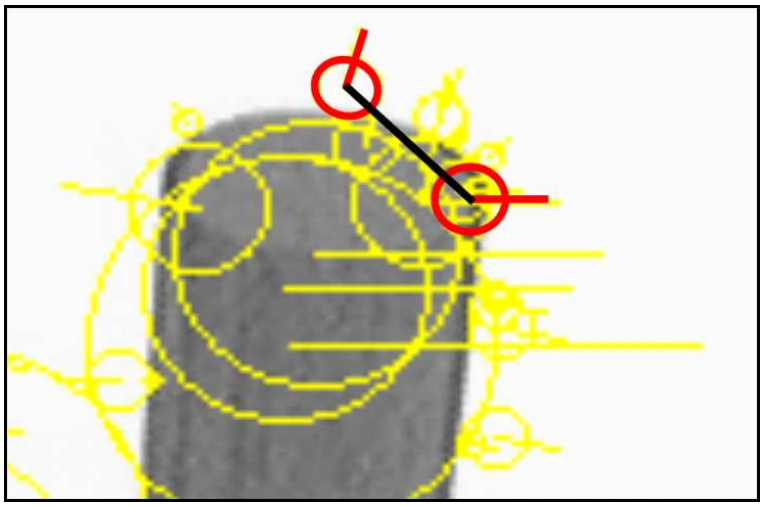
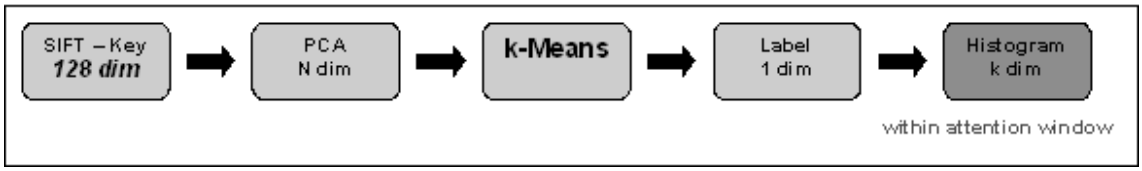
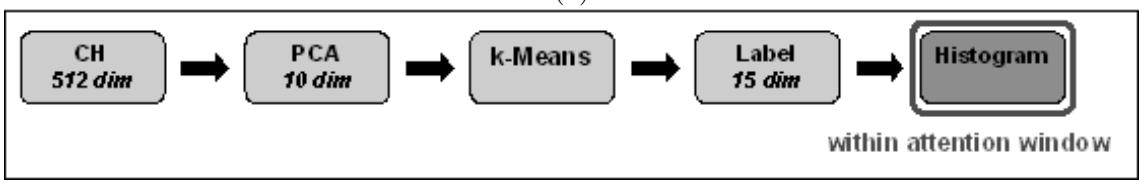


Figure 8: A sample geometric-SIFT grouping representations (two SIFT keys related by angle of the center-center orientation axis).



(a)



(b)

Figure 9: Histograms on symbolic region representations, using a histogram on (a) SIFT prototypical symbolic keys, and on (b) colour prototypical symbolic keys.

6.11 Histograms of Symbolic Region Features

Histograms can also be built on symbolic representations of SIFT and colour histogram reference prototypes (Fig. 9).

6.12 Multi-Channel Features

Various representations can be integrated into a multi-channel feature representation, such as for the case of integrating colour and SIFT key histograms into a combined feature vector (Fig. 10). The investigation of this kind of feature representation isunder construction - it is planned to automatise the geenration of these kind of representations using a feature selection methodology.

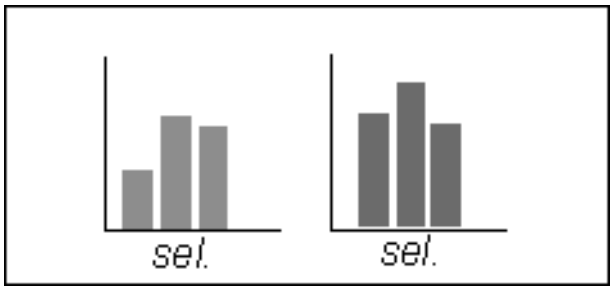


Figure 10: Sketch of a multi-channel discretised feature representation, e.g., for a histogram on discretised SIFT (left) and color (right) histograms.

6.13 Object Representations

Object represenations according to classical and affordance based object entity notion are currently still under construction.

7 Summary and Outlook

This deliverable presented an ontology on perceptual entities under investigation, a hierarchy of representations and the functions to operate on feature representations in order to provide information as a basis for affordance perception.

The deliverable will be extended according to the following guidelines: firstly, we will include classical object recognition methodology, and a framework to proceed from region grouping to object representations. Secondly, we will provide a particular framework on affordance based object representations in order to be able to compare the performance when using different representations.

References

[Cangelosi et al., 2002] Cangelosi, A., Greco, A., and Harnad, S. (2002). *Symbol grounding and the symbolic theft hypothesis*, In Simulating the Evolution of Language, A. Cangelosi and R. Parisi (ed.), Chapter 9, pages 191–210. series. Springer, London, UK.

[Coradeschi and Saffiotti, 2001] Coradeschi, S. and Saffiotti, A. (2001). Perceptual anchoring of symbols for action. In *Proc. 17th International Joint Conference on Artificial Intelligence, IJCAI 2001*, pages 401–412, Seattle, WA.

[Harnad, 1990] Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42:335–346.

[Kamath et al., 2005] Kamath, C., Gezahegne, A., Newsam, S., and Roberts, G. M. (2005). Salient points for tracking moving objects in video. In *Proc. Image and Video Communications and Processing*, volume 5685, pages 442–453, San Jose. SPIE, Electronic Imaging.

[Ke and Sukthankar, 2004] Ke, Y. and Sukthankar, R. (2004). Pca-sift: A more distinctive representation for local image descriptors. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 506–513.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.

[Matas et al., 2002] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proc. British Machine Vision Conference*, pages 384–393.

[Matas and Obdrzalek, 2004] Matas, J. and Obdrzalek, S. (2004). Object recognition methods based on transformation covariant features. In *Proc. 12th European Signal Processing Conference*, pages 1333–1336.

[Mikolajczyk and Schmid, 2004] Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

[Newell, 1980] Newell, A. (1980). Physical symbol systems. *Cognitive Science*, 4:135–183.

[Paletta et al., 2005] Paletta, L., Fritz, G., and Seifert, C. (2005). Xxx. In *Proc. International Conference on Machine Learning, ICML 2005*, page XXX, Bonn, Germany.

[Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600.

[Special Issue:, 2003] Special Issue: (2003). Perceptual anchoring. *Robotics and Autonomous Systems*, 43(2–3).

[Swain and Ballard, 1991] Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7(1):11–32.

[Tomasi and Kanade, 1991] Tomasi, C. and Kanade, T. (1991). Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University.

[Weber et al., 2000] Weber, M., Welling, M., and Perona, P. (2000). Unsupervised learning of models for recognition. In *Proc. European Conference on Computer Vision*, volume 1, pages 18–32.