Project Acronym:	GRASP
Project Type:	IP
Project Title:	Emergence of Cognitive Grasping through Introspection, Emulation and Surprise
Contract Number:	215821
Starting Date:	01-03-2008
Ending Date:	28-02-2012

Deliverable Number:	D7
Deliverable Title :	Monitoring the environment for surprises
Type (Internal, Restricted, Public):	PU
Authors	D. Burschka, Ch. Papazov, O. Ruepp, J. Bohg, D. Kragic, L. Szumilas, and M. Vincze
Contributing Partners	TUM, KTH, TUW

Contractual Date of Delivery to the EC:	28-02-2009
Actual Date of Delivery to the EC:	28-02-2009

Contents

1 Executive summary **5**

2 Objectives **7**

3 Knowledge Representation **9**

3.1 Working Memory - Representation of the Current Scene 12

3.1.1 Hybrid Representation of the Environment 13

3.1.1.1 View-Point Based Image-Database 13

3.1.1.2 Texture Modulation 15

3.1.2 Definition of the Foreground 16

3.1.2.1 Segmentation Strategy 16

3.1.2.2 View-Planning for Efficient Exploration based on Attention 16

3.1.2.3 Related Work 17

3.1.2.4 3-D Reconstruction 17

3.1.2.5 View Planning Method 19

3.1.2.6 Results 22

3.2 Discussion 23

3.3 Object Ontology Representation - Long-Term Memory (Experience) 24

4 Mismatch-Based Surprise Detection **27**

4.1 Self-Localization in the Environment 28

4.1.1 Homing based on three images 29

4.1.2 Homing based on four images 30

4.2 Synthesis of the Expectation (Mismatch-Based Surprise) 30

5 Conclusions and Future Work **33**

Chapter 1

Executive summary

Deliverable D5.1 - "Monitoring the Environment for Surprises" - represents an initial implementation of the cognitive layer in the GRASP project within WP5. The work in this deliverable is strongly interleaved with the research in the deliverables in WP4 where the low level processing layers for the perception are defined and with the work in WP2 where initial understanding of the ontology representation in the GRASP project is defined.

WP5 is responsible to detect novelty in object description and in the actions performed in the scene through a matching process in the Mismatch-Based Layer. A surprise event in this layer triggers an object identification process responsible for data abstraction and labeling. This process initiates the transfer of geometric data structures from the background model into the foreground model representation that contains mission- or task-relevant objects. WP5 is responsible for the construction of the *working knowledge* about the geometry of the environment and for modelling of the typical actions in the local area. This information is essential to detect mismatches between an expectation of the system and the actual perception from the sensory input. The mismatch represents more than just a static change in the geometry or appearance of the environment. We monitor unexpected actions to reduce the number of mismatch triggers in dynamic envrionments, where objects move and only unexpected actions should generate a trigger forcing the system to update its knowledge.

The main input originates from the perception developed in WP4 which will be completed by the haptic information from the *Multimodal Grounding* in WP3 (Task 3.2) later in the project. WP5 implements the cognitive aspects of the project distinguishing it from other existing manipulation approaches, where hard-wired. pre-defined actions are implemented. In GRASP, we put stong emphasis on interaction of the system with its environment. We aim to develop a system that defines its actions as a response to the perception under consideration of its knowledge about the current context. A cognitive system is one that is capable of interacting with humans and other systems in an environment and that is capable to respond to a *surprise*. Our system uses the *surprise* to control the learning about the scene and to trigger its own actions as responses to the external stimuli in the environment. We use this to allow the system to deal with a possible high complexity of the scne. Our system observes a human operator who specifies the mission relevant objects through a direct interaction with them (manipulation). This way, our system does not need to identify and to learn about all objects in the scene but only about the objects that were used by the human. These objects define the *foreground* layer of our representation while the geometrical model of the entire scene remains as a global three-dimensional structure in the *background* layer. Therefore, the results of WP1 are very important to reduce the processing complexity within the WP5. Only a contact of a human hand with an object followed by a change of its position renders the action as something that the system should know about.

According to the Technical Annex of the project, the deliverable 5.1 includes activities in the context of tasks 5.1 and 5.2. The objectives of these tasks are the following:

- **Task 5.1 - Implementation of Surprise Event Hierarchy** Implementation of the surprise event hierarchy including a simple mismatch-layer, a passive prediction-based layer, and a implausibility layer. The work will focus on control of the attention to predict possible events and view planning strategies based on findings about human behaviors in similar situations in neurosciences. The

sensor model of the used sensor will be mapped on the requirements of the abstract task definition to find and optimal strategy suited for a given imaging model.

- **Task 5.2 - Evaluation of Efficient Methods to Monitor Changes** Evaluation of efficient methods to monitor changes in the environment that will be insensitive to sensor inaccuracies and that compensate eigen-motions /actions of the system in the environment. In collaboration with WP4, an internal representation of the environment will be generated that will define the expectations of the system. This representation will go beyond a geometric representation of the world and will define also contextual and dynamic information about the world. This representation will allow to generate expectations in collaboration with WP6 that will help the system to define a *surprise* that can happen at different levels ranging from an unexpected change in the 3D description to unknown new motion types that need to be learned by the system.

The main focus of our work was on Task 5.2, because it defines the basic representation of the knowledge that is necessary to predict expectations of the system. This is essential for the mismatch detection in Task 5.1. The following text is structured as follows:

Chapter 2 describes the goals that were implemented in the first twelve months of the GRASP project. We present the goals and necessities that need to be considered in the first phase of the project.

Chapter 3 reports on activies that lead to the initial implementation of the knowledge layer. We distinguish between the *Working Memory* that stores the geometric and dynamic representation of the scene and the actions observed in it. This representation results from analysis of the preceptual input combined with the knowledge stored in the *Atlas Representation*, where a-priori knowledge about the environment is stored. This chapter describes activities related to the Task 5.1.

Chapter 4 reports on activies in the field of the Task 5.1. We present our initial implementation of the mismatch-based layer in the surprise detection hierarchy. We report the results of evaluation of various modelling approaches that aim to simplify the monitoring task. Since this processing represents a continuously running task, a special emphasis was put on finding a light-weight process that will not create too high load on the manipulation system in the GRASP project.

Chapter 5 concludes with a discussion of the role of the processing accomplished in Deliverable 5.1 in the context of the processing control of the GRASP system.

Chapter 2

Objectives

Cognitive systems need to be capable of identifying the mission relevance and of learning the model description of objects by themselves during joint action with a human operator. Most generally, a model of context specifies the entities to observe, the properties to measure and the relations to detect [Win]. Dey [Dey01] proposed an operational model for context aware perception. In this model, a situation is defined as a configuration of entities and relations relative to a task. The task serves to determine which entities and relations are of interest and should be observed. We transfer these findings into our foreground/background representations, which allows to decouple complex object recognition loops from the low level 3D reconstruction. The deliverable discusses the following problems:

Structure of the Mismatch-Based Layer. The reconstructed 3D model of the environment is used to predict the expectation for a given camera view. This expectation is compared to the actual perception to detect mismatches (Fig. 2.1). The mismatch-based layer is the lowest level in the Surprise Event Detection Hierarchy suggested in Technical Annex of the GRASP project (Fig. 3.5). Our goal in this funding period is to implement an appropriate knowledge representation that will allow to store efficiently the geometric and dynamic knowledge about the environment and to use this data to detect *surprise events* in a typical manipulation scenario. An example of a simple surprise detection is the detection of unexpected glasses in the table scenario in Fig. 2.1.

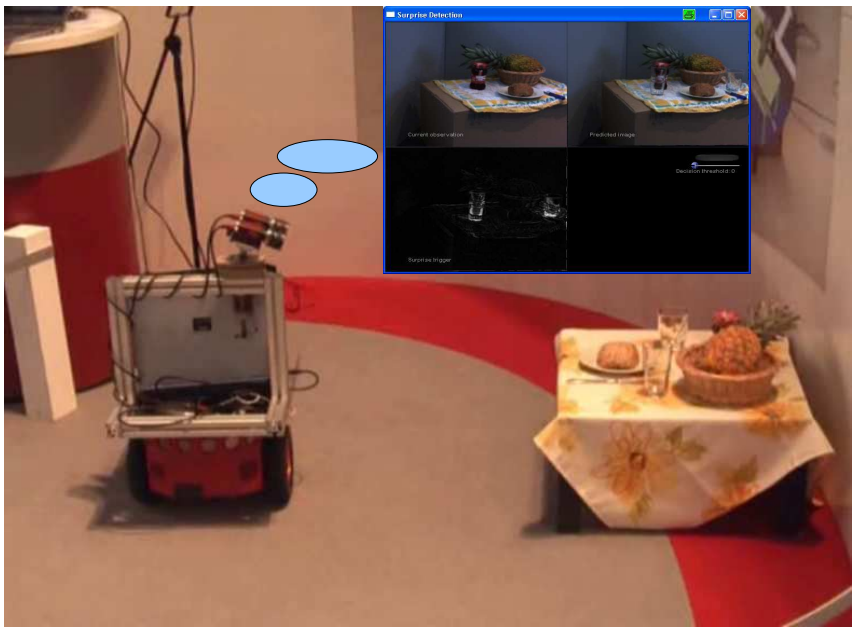

Figure 2.1: Mismatch detection in a typical household scenario - glasses were placed on the table.

Segmentation into Foreground/Background Model. The *background model* represents a geometric description of the scene that can be extracted from the sensor perception without any additional

assumptions about the world. The *foreground model* stores information that can be seperated from the *background model* because of human actions that cause motion of geometric scene elements relative to each other, e.g., human picks up a cup and the cup becomes in this case part of the foreground model. In parallel, the system can run object recognition algorithms on the background model to identify additional mission relevant objects similar to the objects moved by the human. The objective of this funding period is to find efficient representations for these knowledge types that will allow a fast access to the information and an efficient storage of the data.

Generation of Expectations. There are very good claims and studies in the literature on surprise, like the idea that it depends on expectations, the claim that its intensity depends on the "unexpectedness" of the stimulus [OP87, WMS97]. The importance of having a precise characterization and formalization of expectations is very relevant for modeling cognitive agents [Bra88, RG92]. Different categories of expectations can be identified depending on the degree of the belief. We let implicit the notion of value of a goal. We call forecast a belief about a future state of the world and we distinguish it from a simple hypothesis. We have a prediction when the degree of certainty is close to 100 per cent [MC02]. Expectations in our ontology are not indifferent hypotheses, forecasts or predictions. They imply a subjective concern in the realization of a given situation.

WP5 defines triggers for the processing in the manipulation system using the ontology defined in WP2 to interact with various components of GRASP. The framework is depicted in Fig. 2.2.

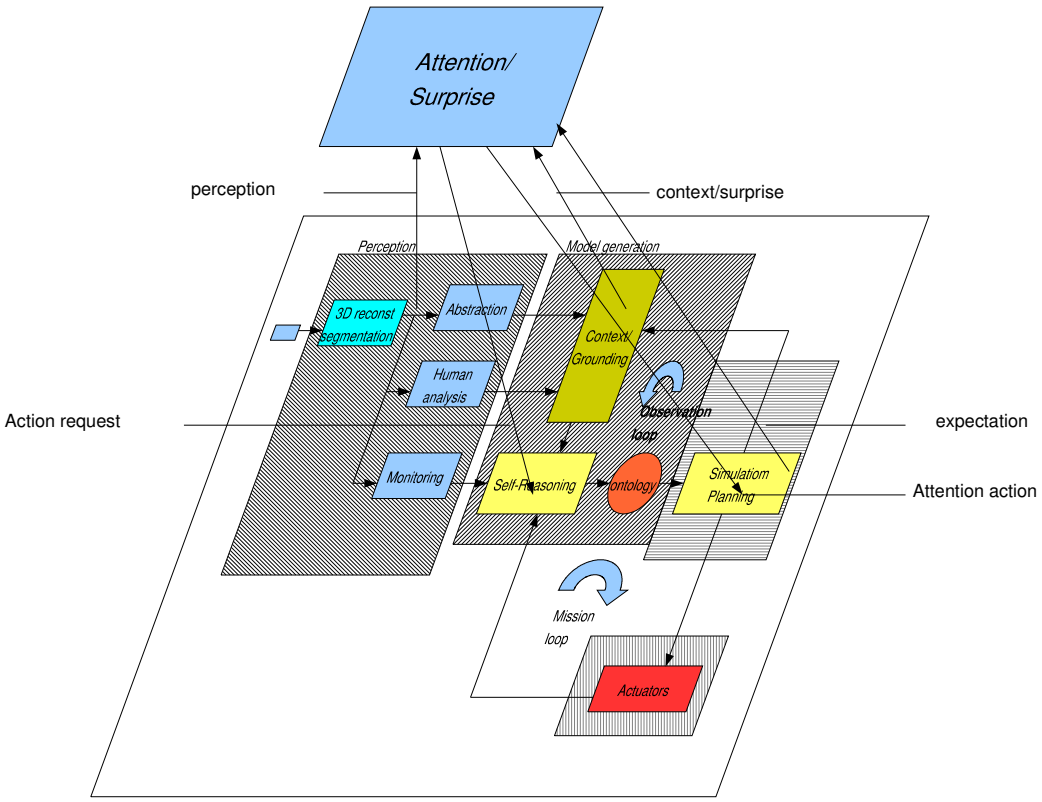

Figure 2.2: Communication structure of the Surprise/Attention Layer in GRASP.

The cognitive surprise/attention layer to be developed in WP5 interacts with multiple modules of the GRASP project. It uses the results of the perception layer developed in WP4 to abstract the raw sensor data to geometric descriptions, human hand models and motions in the scene. This information is stored within the Attention-Surprise module to trigger processing to add information to the knowledge base. This requires interaction with the planning modules as well to optimize the knowledge acquisition by specifying next-best views and hypothesis about the observations. The goal is to map the apriori knowledge in an *Atlas* to the current observation and to adjust the parameters to match the sensor data.as well to optimize the knowledge acquisition by specifying next-best views and hypothesis about the observations. The goal is to map the apriori knowledge in an *Atlas* to the current observation and to adjust the parameters to match the sensor data.

Chapter 3

Knowledge Representation

Sensation and perception are key components of cognitive systems. Cognition can be defined as "generation of knowledge on the basis of perception, reasoning, learning and prior-models". Perception is the main source of information for reasoning and learning capabilities. Our goal is to understand, how a nervous system gathers and interprets information from the vast array of sensory stimulation that reaches us and to map it on physical systems available on our cognitive demonstrators.

Scene classification is an important task in cognitive systems. It helps in sensor-based 3D model generation to discriminate between objects interesting for missions (*foreground*) and *background* objects relevant merely for localization and obstacle avoidance. It is also used to trigger different behaviors of the robot depending on the scene type. Exemplary classification results relevant for the GRASP project are: factory environment, household environment, and table desk. The target selection task is a challenging part of the system and can be implemented as a manual or automatic process. Examples in 2D image space are described in [SD98, SMB98] in more detail. Interesting targets like single standing objects in the scene need to be separated from the supporting planes of the floor and walls that are merely relevant for collision avoidance.

Single standing objects are categorized as *foreground*. They need to be separated from the environment structure (*background*) first. In an additional step, the remaining *foreground* objects are classified according to their shape, extension and movement relative to the scene. The *background* structures are used in a subsequent classification process to classify the scene structure according to the criteria described above.

We consider the visual and haptic perception as the stimuli generating the input for our cognitive processing. This multi-modal sensor input will allow to extract the initial information about *foreground objects* in the scene, to classify them, and to match them to already known representations in the *Ontology (long-term memory)*.

We use for the knowledge representation in the *Ontology* in the GRASP project an analogy to the cognitive capabilities of the human brain and its different strategies, how to store and process the information in the most efficient way. The brain does not store memories in one unified structure. Instead, different types of memory are stored in different regions of the brain (Fig. 3.1). Long-term memory is typically divided up into two major headings: declarative memory and implicit memory (or procedural memory).

1. Declarative memory refers to all memories that are consciously available. These are encoded by the hippocampus, entorhinal cortex, and perirhinal cortex, but consolidated and stored elsewhere in the cortex. The precise location of storage is unknown, but the temporal cortex has been proposed as a likely candidate. Declarative memory also has two major subdivisions:
   - Episodic memory refers to memory for specific events in time
   - Semantic memory refers to knowledge about the external world, such as the function of a pencil.
2. Procedural memory refers to the use of objects or movements of the body, such as how exactly to use a pencil or ride a bicycle. This type of memory is encoded and probably stored by the cerebellum and the striatum.

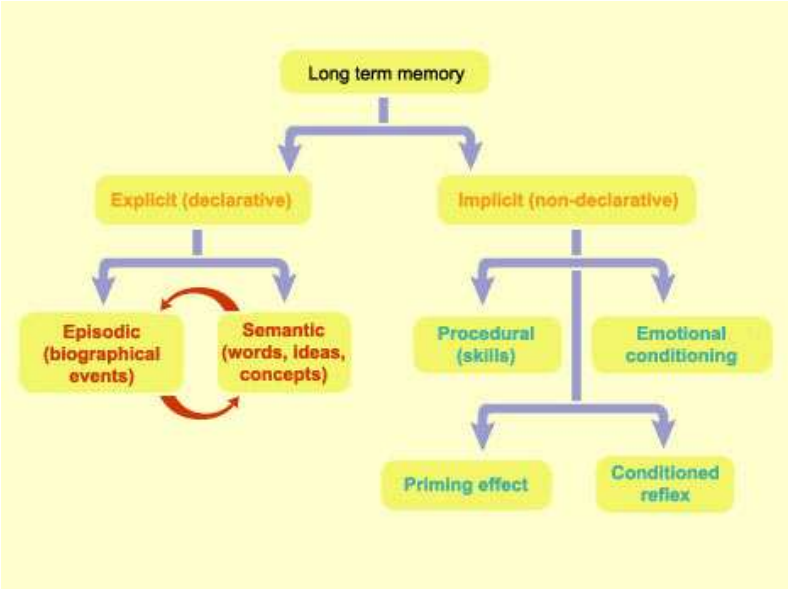

Figure 3.1: Different Types of Long-Term Memory (Source: http://thebrain.mcgill.ca)

There are various other categorizations of memory and types of memory that have captured research interest. Prospective memory (its complement: retrospective memory) is an example.

Long-term memory in the brain is memory that can last as little as a few days or as long as decades. It differs structurally and functionally from working memory or short-term memory, which ostensibly stores items for only a short time. Working memory (also referred to as short-term memory, depending on the specific theory) is a theoretical construct within cognitive psychology that refers to the structures and processes used for temporarily storing and manipulating information. There are numerous theories as to both the theoretical structure of working memory as well as to the specific parts of the brain responsible for working memory.

There have been numerous models proposed regarding how working memory works, both anatomically and cognitively. Of those, three have received the distinct notice of wide acceptance:

- **The Baddeley and Hitch model**[BH74] - Baddeley and Hitch (1974) introduced and made popular the multicomponent model of working memory. This theory proposes that two "slave systems" are responsible for short-term maintenance of information, and a "central executive" is responsible for the supervision of information integration and for coordinating the slave systems (Fig. 3.2). The central executive is, among other things, responsible for directing attention to relevant informa-

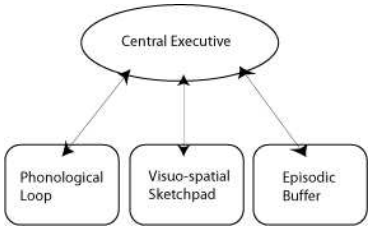

Figure 3.2: Schematic of Baddeley's Model.

tion, suppressing irrelevant information and inappropriate actions, and for coordinating cognitive processes when more than one task must be done at the same time.

The phonological loop (or "articulatory loop") as a whole deals with sound or phonological information. It consists of two parts: a short-term phonological store with auditory memory traces that are subject to rapid decay and an articulatory rehearsal component that can revive the memory

traces. This auditory example can be matched to a haptic information from the gripper to the phonological information and segmented primitives of an action.

Any auditory verbal information is assumed to enter automatically into the phonological store. Visually presented language can be transformed into phonological code by silent articulation and thereby be encoded into the phonological store. This transformation is facilitated by the articulatory control process. The phonological store acts as an 'inner ear', remembering speech sounds in their temporal order, whilst the articulatory process acts as an 'inner voice' and repeats the series of words (or other speech elements) on a loop to prevent them from decaying. The phonological loop may play a key role in the acquisition of vocabulary, particularly in the early childhood years.

The visuospatial sketchpad (Fig. 3.2) is assumed to hold information about what we see. It is used in the temporary storage and manipulation of spatial and visual information, such as remembering shapes and colors, or the location or speed of objects in space. Logie has proposed that the visuospatial sketchpad can be further subdivided into two components:

1. The visual cache, which stores information about form and color.
2. The inner scribe, which deals with spatial and movement information. It also rehearses information in the visual cache and transfers information to the central executive

In 2000 Baddeley added a fourth component to the model, called the 'episodic buffer'. This component is a third slave system, dedicated to linking information across domains to form integrated units of visual, spatial, and verbal information with time sequencing (or chronological ordering), such as the memory of a story or a movie scene. The episodic buffer is also assumed to have links to long-term memory and semantic meaning.

- **The theory of Cowan** - Cowan [Cow05] regards working memory not as a separate system, but as a part of long-term memory. Representations in working memory are a subset of the representations in long-term memory. Working memory is organized in two embedded levels. The first level consists of long-term memory representations that are activated. There can be many of these, there is no limit to activation of representations in long-term memory. The second level is called the focus of attention. The focus is regarded as capacity limited and holds up to four of the activated representations.
- **The theory of Ericsson and Kintsch** - Ericsson and Kintsch [KPE99] have argued that we use skilled memory in most everyday tasks. Tasks such as reading, for instance, require to maintain in memory much more than seven chunks - with a capacity of only seven chunks our working memory would be full after a few sentences, and we would never be able to understand the complex relations between thoughts expressed in a novel or a scientific text. We accomplish this by storing most of what we read in long-term memory, linking them together through retrieval structures. We need to hold only a few concepts in working memory, which serve as cues to retrieve everything associated to them by the retrieval structures

In GRASP, we follow the structure suggested by Baddeley with the long-term memory and the short-term memory maintained by the central executive. Our system consists of two databases storing a-priori knowledge about the world *the Ontology* corresponding to the long-term memory and a *Working Memory* representing the current visual an spatial representation of the world (visuospatial sketchpad). In this layer, the episodic buffer is implemented as a system storing the typical actions applied to a mission relevant object.

The two layers in GRASP (Fig. 3.3) have the following representation:

- **Ontology Representation (Experience of the System)** - this information represents a-priori knowledge given to the system from an expert or representations of the environment collected in previous operations in the same or similar environment. An important difference of the proposed system to many other systems suggested before is that it is supposed to interact with its environment in a cognitive way. This means that the system does not operate based on a set of pre-defined rules but it tries to learn from its own actions and actions of other agents in the environment (human or other robots). The information stored in the Ontology represents a generic knowledge about a class of object. In the initial implementation, this can be directly the information of specific objects in the scene. We try to generalize this information to entire object classes. This way, information about manipulation capabilities of one object (e.g., specific cup) will be transferred as possible hypothesis for a similar object type.

- **Working Memory**- Working memory is a theoretical construct within cognitive psychology that refers to the structures and processes used for temporarily storing and manipulating information. In our system, the *experience* needs to be grounded to a given environment. A typical environment of the cognitive system represents a geometric and semantic description of objects and actions in a given geometric environment. To implement the goals of GRASP, we expect to operate in highly complex environments, where the system must not try to analyse all elements of the scene as it is often the case in other current manipulation systems (*http://www.smerobot.org/*) but it needs to focus its *attention* on mission relevant objects whose properties need to be explored for a successful interaction with the world.

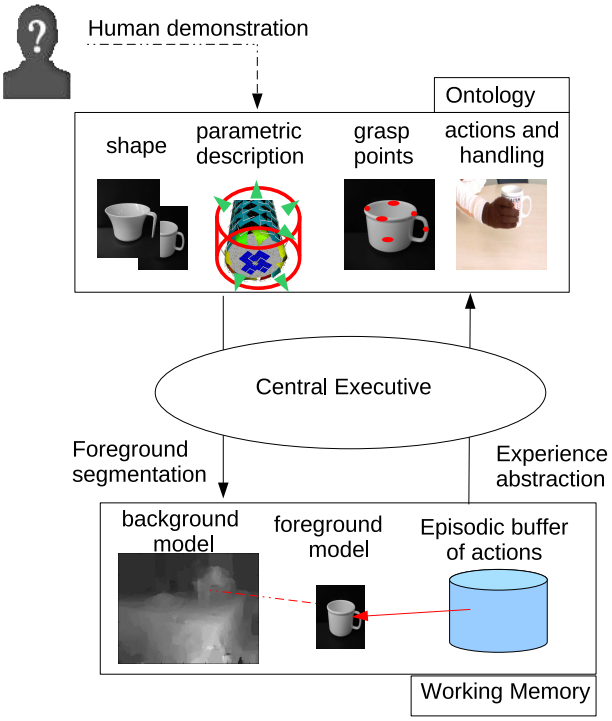

Figure 3.3: Knowledge Layers in GRASP.

Our system reconstructs the 3D information fusing the 3D perception of the stereo camera and possible additional camera systems in the wrist of the robot. The global view through the binocular system of the robot does not provide a complete information in the scene due to occlusions in the scene. This fact needs to be taken into account while fitting objects into the reconstructed 3D information.

The completeness of the model can be increased by active exploration of the environment (see Section 3.1.2.2) and by adding additional monocular information from a camera mounted in robot's wrist. This type of camera allows the system to look around the object without complex robot motions through the environment. The corresponding 3D reconstruction algorithms from motion from TUM are presented in the deliverables of WP4.

3.1 Working Memory - Representation of the Current Scene

The goal of the GRASP project is an operation in complex structured environments with occlusions, self-occlusions, and a high number of objects. In such environments it is not feasible to label all the reconstructed information to objects and it is not necessary, because only a very limited number of them is actually mission relevant. The geometry of the scene is necessary for motion planning and collision avoidance, therefore, it needs to be reconstructed with high accuracy. This information does not necessarily need to be labeled for the correct operation of the system.

Fig. 3.4 depicts an exemplary reconstruction of the scene from a stereo system looking at a simple table setup. Our stereo reconstruction system developed in WP4 reconstructs the corresponding point cloud.

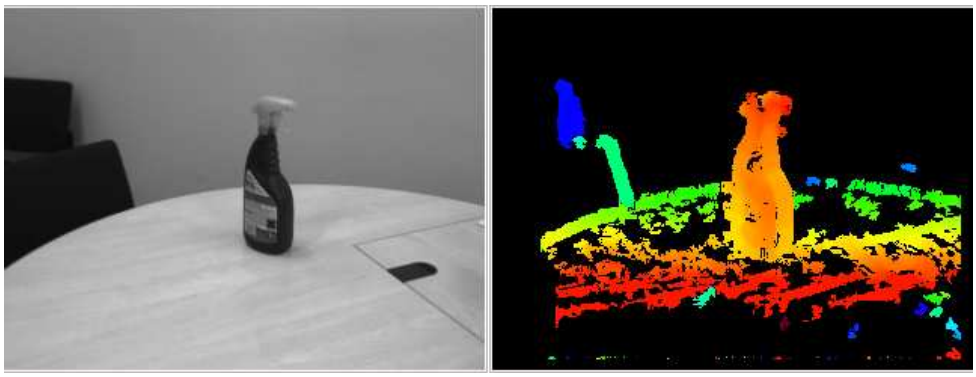

Figure 3.4: 3D information from a typical table setup.

We see that the input data does not distinguish between the cleaner-bottle on the table and the table itself. In the first step, the entire information is stored as background in our model. As we already mentioned before, this information can be used to plan actions and to avoid collisions while moving the manipulator in the table area.

The information stored in the *working memory* is used for an additional purpose in the system, which is novel for the manipulation system currently developed. We use this information not only for path planning but also to detect changes in the environment. It is theoretically possible to monitor for changes in 3D space, but a single camera view can cover a significant area in 3D space. Additionally, occlusions in the current view make it difficult to decide if an absence of an object in the current view is due to a change in the environment or due to the occlusion by an object closer to the camera. For this decision, objects from the 3D model need to be reprojected into an expected camera image which later can be compared with the actual view. Therefore, in addition to the geometric representation some information about the appearance of the object is necessary. This lead us to implementation of a hybrid structure of the working memory consisting of both the geometric and appearance information combined in a hybrid representation of the environment.

3.1.1 Hybrid Representation of the Environment

Essential novelty for simple detection of mismatches in the environment is a hybrid image-based and geometric representation of the environment that allows planning of actions using the geometry and the image-based portion to control attention and simplify the prediction directly as expected sensor images. While the construction of geometric models has been already sufficiently presented in multiple applications, our focus in this deliverable is on efficient detection of surprise events in the scene and support for path planning based on geometry of the scene.

Our goal in this reporting period was an implementation of the lower layers in the processing hierarchy of a surprise event (Fig. 3.5).

The initial trigger for a surprise event is a mismatch between the expectation of the system and the perception from the sensors on the robot. The hybrid structure of our model representation simplifies the processing here. The image-based information from our model is used directly to predict an appearance of the scene from a specific view-point while the geometry part is used for path planning and obstacle avoidance.

We evaluated two forms of the image-based prediction for their applicability for the GRASP project. They differ in the way, how the image information is stored in the model.

3.1.1.1 View-Point Based Image-Database

One type of image-based scene representation that recently has become very popular uses view-dependent geometry and texture. Instead of computing a global geometry model which is valid for any viewpoint and viewing direction, the geometry of the scene is locally estimated and only holds for a small region

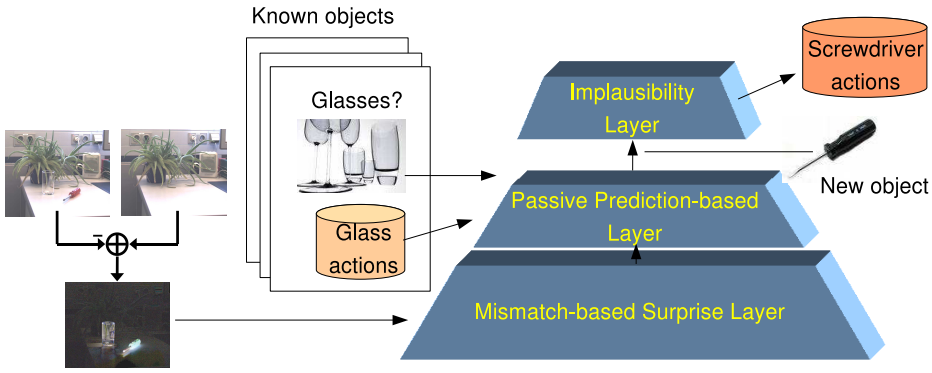

Figure 3.5: Hierarchical processing of a Surprise event (from the Technical Annex of the GRASP proposal).

in the viewpoint space. It has been shown that this approach is suitable especially when the scene contains specular and translucent objects. To extract local geometry information, per-pixel depth maps are calculated for each reference image, i.e., the left image of each captured stereo pair. Loopy belief propagation [FH04] minimizes a matching cost volume and yields the most probable depth value for each pixel, assuming that the scene is smooth between depth discontinuities. A triangulated mesh is reconstructed from each depth map and simplified using the algorithm in [GH98]. While these steps are done off-line, the view selection and view synthesis, as explained in the following, are performed on-line.

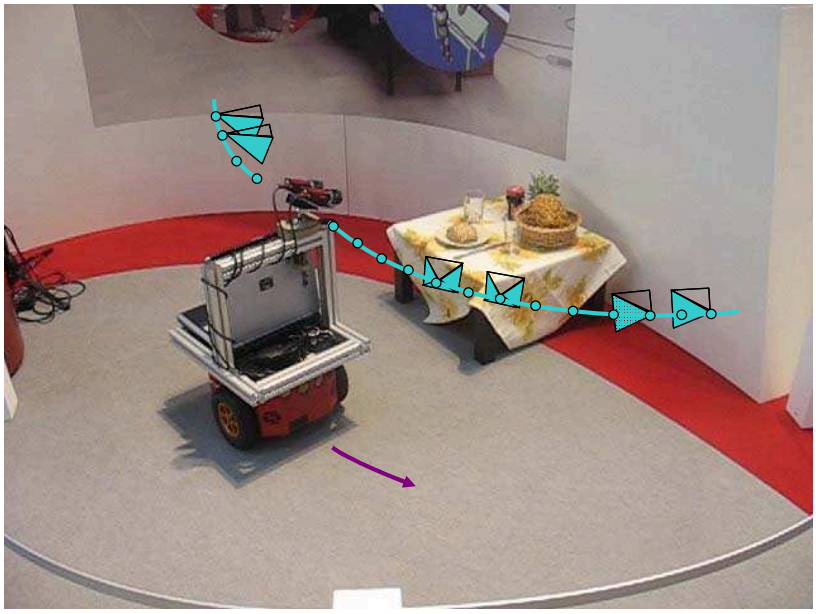

Figure 3.6: Acquisition of a set of images with a stereo camera (work together with DFG CoTeSys project 319).

This system acquires images along the traversed trajectory and saves together with the geometrical representation of the scene. The idea is to provide a dense set of image information along the trajectory that will allow a prediction of any additional virtual view from a position not seen in the initial image set (Fig. 3.7).

Any additional virtual image μ_{ML} from a direction similar to the already known views x_i can be generated using this approach. The synthesized image contains contributions from other images observing the scene. The task of the second rendering pass is to find an acceptable estimate for the true color value. The color values from the reference images are assumed to be samples from a Gaussian distribution whose mean is the true color value. Surprise detection: If the current observation yields a sample value which is, due

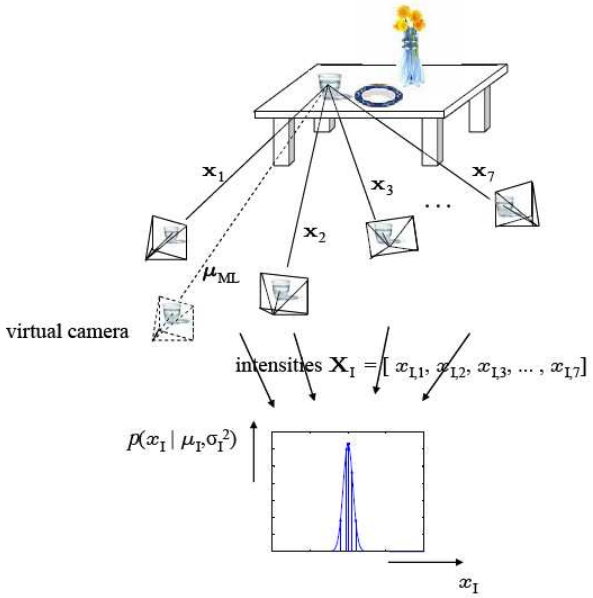

Figure 3.7: Virtual view synthesis for a virtual camera from existing image set.

to changes in the scene, largely different from the reference samples, a surprise trigger is generated in a given pixel region.

This approach has an advantage for applications, where the system operates only in a limited space, where only a limited set of images is necessary in total to predict a new view even from a previously unknown position. The information is at the level of detail at a given position.

3.1.1.2 Texture Modulation

An alternative approach, which was evaluated for the image-based part of the *working memory* is texture mapping on the geometry of the scene. A direct mapping of the texture on the faces of the object in the scene is a common approach in Computer Graphics. Unfortunately, it requires a detailed modeling of the filigrance structures in the scene to preserve the correct re-projection depending on the view. To allow an image differencing between the prediction of an image expectation and the current image requires a good match of the structural pixel values.

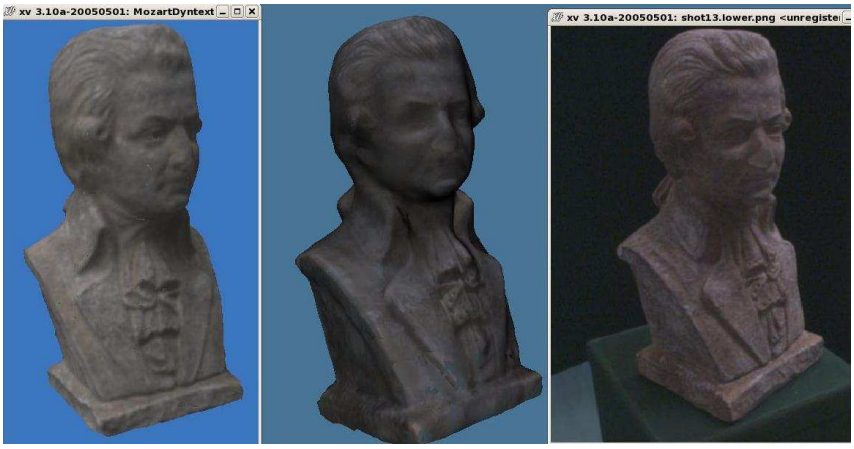

Figure 3.8: Virtual view synthesis for a virtual camera from existing image set.

Fig. 3.8 depicts the problem associated with too coarse geometrical modeling. The texture appears flat and does not follow correctly the details on the surfaces.

We chose to evaluate the Dynamic Texture approach [CYJ02] as an alternative for modeling of the appearance of an object. This approach allows an easy reconstruction of objects from very few images. In opposite to the approach in Section 3.1.1.1 the system stores *texture bases* attached directly to the surfaces and modulates them depending on the viewing angle.

3.1.2 Definition of the Foreground

Ground Plane Obstacle Detection (GPOD) using stereo disparity was first reported by Sandini et al. [FGM90], and refined by Mayhew et al. [MZC92] and by Brady et al. [SB97]. These approaches use orthogonal regression techniques to estimate the parameters of the ground plane. Approaches like the one from Brady et al. [SB97] use line features grouped in a Hough transform to detect obstacles in the environment. We plan to use in contrast directly the disparity information in the images. We pursue an approach that fits multiple surfaces into a dense disparity image to allow calibration, localization and object classification from a single image.

The segmentation of background point clouds into foreground objects is based on a-priori knowledge of the world and human activities in the scene. Scene, like the one depicted in Fig. 3.4, shows an object on a supporting plane.

3.1.2.1 Segmentation Strategy

Goal in the project is segmentation based on the motion induced through human action, for now - segmentation is based on supporting plane subtraction and contact with a human hand (actually skin) (Fig. 3.9).

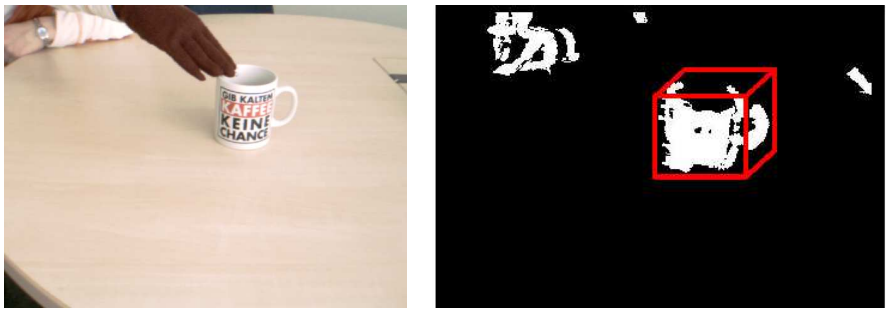

Figure 3.9: Object moves to Foreground description through an interaction with the human hand.

3.1.2.2 View-Planning for Efficient Exploration based on Attention

Visual 3-D reconstruction and registration methods are gaining immense importance in many practical applications in everyday life. Unfortunately, the applicability of such techniques is often limited because the accuracy requirements of procedures are too high, or if seen the other way around, the reconstruction quality is too low. One possibility to improve this situation is active planning of camera movements such that the maximum reconstruction quality is assured. To address this problem, we have developed a simple, easy to use approach that is directly inspired by geometric considerations.

This Section presents a geometrical approach to define efficient exploration strategies depending on the perception properties of the underlying sensor in the system. This research is a basic component in our collaborative sensing approach where the exploration of a local area is performed by several small cognitive agents in a joint action effort. We plan to map it onto biological exploration strategies of insects collaborating in exploration tasks, like wasps. Our current research shows, that although it is desirable to learn from biological systems, it is easier to explain biological behavior by mathematical models known from robotics and to use the biology to select the appropriate alternatives.

Vision-based technologies are becoming more and more important in everyday life. One of the most important tasks in that context is the 3-D reconstruction and modeling of objects and scenes. There are many problems that need to be dealt with in the context of that problem, and the accuracy of the

reconstruction is one of them. Of course, it is always possible to increase the accuracy by increasing the resolution of the employed sensors. But that will also lead to an increase of sensor size, cost, and probably also computation time, because more data needs to be processed. This Section is concerned with a different approach of improving reconstruction quality, by finding a strategy for sensor placement such that the accuracy of 3-D reconstruction tasks will be improved.

One possible application of our ideas is in computer-aided surgery, where, e.g., Burschka et al. [BCD+05, BLT+05] have developed techniques for 3-D reconstruction and active guidance of surgeons. It is clear that the accuracy of the 3-D reconstruction is of high importance, because a too low accuracy can in extreme cases mean injuries to the patient. It would be interesting to combine the methods developed herein with their system.

Another important application of our approach is view planning for flying systems and manipulation purposes, where additional information needs to be acquired from, e.g., camera-in-hand images or interacting flying agents from as few additional positions as possible.

3.1.2.3 Related Work

In the context of SLAM methods, many different approaches to the accuracy maximization problem have been examined. Vidal-Celleja et al. [VCDACM06] have developed a scheme for active control of a 6DOF camera, where possible movement actions are evaluated according to information-theoretic optimality criterions. It is assumed that the camera movement is chosen from a discrete set of actions at certain timesteps, and at each timestep the optimal action is determined.

Wenhardt et al. [WDAN07] use a different approach: They try to choose actions such that one of several characteristics of the covariance matrix are minimized. The minimization procedure is implemented as an exhaustive search, and the minimization criteria (examined as completely independent strategies) are the entropy, largest eigenvalue, and trace of the covariance matrix. As opposed to the approach of Vidal-Celleja et al. [VCDACM06], it is assumed that the camera will directly "jump" to the computed optimal position.

That it is not advisable to blindly use information maximization schemes has been shown by Sim [Sim05] in his work on bearings-only SLAM. It is shown that, when the sensor (e.g., a camera) is always driven to the "optimal" position (considering maximum information gain from the measurement), the update step for the Extended Kalman Filter becomes numerically unstable, and this in turn adversely affects the state estimation. The strategy developed in that work therefore aims at maximizing stability of the Kalman Filter update.

Another interesting method has been discussed by Whaite et al. [WF97]. Their approach is probably the one that is most similar to our work: A sensor that is constrained to move on a sphere surface is considered, and the optimal movement direction is computed with respect to the prediction variance. However, our idea uses a simplified approach to evaluate view points, and we also do not compute movement directions, but absolute positions instead.

An additional aspect of Active View Planning that is not discussed in this Section is dealing with self-occlusions of objects and assuring that all parts of an object are seen. Since the problem has been researched for several years, a lot of different methods have also been proposed [MB93, KD95, CK88]. A more recent contribution to solving that problem is the paper of Chen and Li [CL04].

3.1.2.4 3-D Reconstruction

Our method can generally be applied for any 3-D reconstruction technique that generates point position estimates as well as estimates of point position uncertainty in form of a covariance matrix. Typically, SLAM methods employing the Kalman Filter generate that kind of information, and one such method is what we have used for testing.

The technique of our reconstruction method is described by Davison et al. [DRMS07]. Their MonoSLAM algorithm, which is based on the Extended Kalman Filter, allows for simultaneous pose estimation and map building in an unknown environment. The modification we make to Davison's system is that we use two cameras instead of one, so we do not have to deal with complicated feature position initialization. We will, however, assume that only one of both cameras is able to move, and the movement of that

camera will be optimized. Only a very brief summary of the data structures of the reconstruction and pose estimation system will be given here, for a complete description the reader is referred to Davison's original work.

In our specific system, the state vector $\hat{x}$ contains the camera state estimates $\hat{x}_1, \hat{x}_2$ as well as the 3-D coordinates $\hat{y}_i$ of the n points under consideration. The state vector can be partitioned as follows:

$$\hat{x} = (\hat{x}_1, \hat{x}_2, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)^T \tag{3.1}$$

The camera state vectors $\hat{x}_i$ can be further partitioned into a 3-D position vector r_i^W, a rotation quaternion q_i^{WR} describing the camera's orientation, a translational velocity vector v_i^W and a rotational velocity vector ω_i^W:

$$x_i = (r_i^W, q_i^{WR}, v_i^W, \omega_i^R)^T \tag{3.2}$$

As the points of the scene are assumed to be static, the state transition for the point coordinates is the identity. The camera motion is assumed to be affected by constant linear and angular velocity in-between frames. Changes in the velocities are modeled as noise effects: We assume that unknown linear and angular accelerations a_i^W and α_i^W are applied at each time step, which can be seen as process noise. The corresponding total acceleration vectors n_1, n_2 are partitioned into linear and angular acceleration parts as follows:

$$n_i = \begin{pmatrix} V_i^W \\ \Omega_i^R \end{pmatrix} = \begin{pmatrix} a_i^W \Delta t \\ \alpha_i^R \Delta t \end{pmatrix} \tag{3.3}$$

With this notation, the total state update for the a pose of camera i can be expressed through a function f_v as follows:

$$f_v(x_i^v) = \begin{pmatrix} r_i^W + (v_i^W + V_i^W)\Delta t \\ \mathbf{qnorm}(q_i^{WR} \times q((\omega_i^R + \Omega_i^R)\Delta t)) \\ v_i^W + V_i^W \\ \omega_i^R + \Omega_i^R \end{pmatrix} \tag{3.4}$$

Here, $q(\cdot)$ is a function that maps a angle-axis rotation vector to the corresponding quaternion. The newly computed rotation quaternion is normalized by use of the **qnorm** function to assure that it always has unit length. Otherwise, errors introduced through the quaternion multiplication might add up over time such that the rotation description becomes unusable.

For the measurement function, we apply the standard pinhole camera model, and assume that the measurement vector consists of stacked 2-D measurements of the points in our model. From a combination of point coordinates y_j and camera parameters x_i, we can generate the corresponding expected measurement. First of all, note that the position of point j relative to camera i can be expressed as

$$h^R(x_i, y_j) = \mathrm{R}^{RW}(y_j^W - r_i^W), \tag{3.5}$$

where R^{RW} is the rotation matrix between the world reference coordinate frame W and the camera coordinate frame R. According to the pinhole model, the measurement then looks like this:

$$h(x_i, y_j) = \begin{pmatrix} u_0 - fk_u \frac{h^R(x_i,y_i)_x}{h^R(x_i,y_i)_z} \\ v_0 - fk_v \frac{h^R(x_i,y_i)_y}{h^R(x_i,y_i)_z} \end{pmatrix} \tag{3.6}$$

where fk_u, fk_v, u_0, v_0 are the usual camera calibration parameters.

The computation of the various Jacobian matrices required for the Extended Kalman Filter is now a straightforward matter that will not be discussed in detail. For our testing system, we used a Maple-based code generation system to perform the associated computations.

Because we are mainly concerned with uncertainties of our estimate in this work, we also make use of the convenient notation for relevant parts of the state covariance matrix, as introduced in Davison's [DRMS07] work:

$$P = \begin{pmatrix} P_{x_1x_1} & P_{x_1x_2} & P_{x_1y_1} & P_{x_1y_2} & \ldots \\ P_{x_2x_1} & P_{x_2x_2} & P_{x_2y_1} & P_{x_2y_2} & \ldots \\ P_{y_1x_1} & P_{y_1x_2} & P_{y_1y_1} & P_{y_1y_2} & \ldots \\ P_{y_2x_1} & P_{y_2x_2} & P_{y_2y_1} & P_{y_2y_2} & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{3.7}$$

We will primarily be interested in the sub-matrices $P_{y_iy_i}$, which provide a direct description of the uncertainty of the point y_i.

3.1.2.5 View Planning Method

As has been pointed out by, e.g., Sim [Sim05], maximizing the information gain from measurements is equivalent to moving the camera to a position that is orthogonal to the principal direction of an observed landmark's covariance ellipsis. This is also what one would intuitively expect, since by looking at a point feature from a specific direction, we can determine the feature position pretty well in the directions parallel to the camera plane, while we are not able to determine the 3-D depth of the feature. Figure 3.10 visualizes the concept.

Inspired by this idea, our approach to finding the optimal camera position can be described informally as follows: First of all, we compute the covariance axes for all points, establishing the principal axes of the covariance ellipsoid. If we place the camera on a plane that contains the point and has a normal that is orthogonal to one of the axes, we will maximally reduce the covariance in the direction of that axis after a measurement.

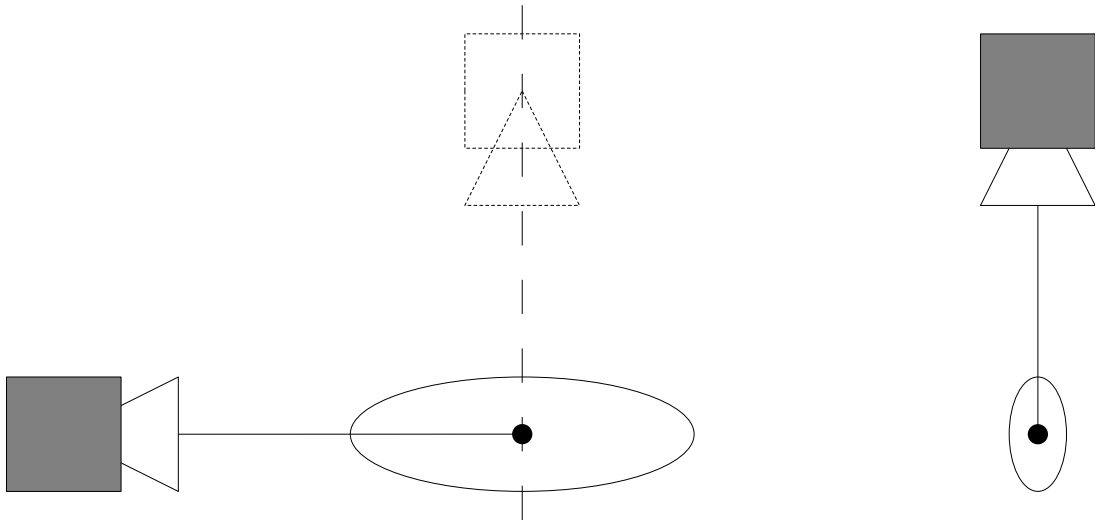

Figure 3.10: The left picture shows the following situation: A feature has been measured from the shown camera position, and the measurement lead to the shown point estimate and covariance. The dashed line is orthogonal to the major covariance ellipsis axis, and also with dashed lines we indicate a camera position on that line, which seems like a good choice intuitively. The right image shows how the situation could look like after a measurement from that position: The worst covariance has been maximally reduced.

Using this information, we can, for each point and one of its covariance axes, determine a plane on which the camera should be placed if we wish to minimize the covariance in that direction for that point. This method alone would lead to ill-conditioned filter updates as shown by Sim [Sim05], but we are not finished yet.

Based on the information that has been computed so far, we can introduce a penalty function for camera locations, depending on the location's distance to the "optimal" planes. A first, simple idea might be using, e.g., the squared distance to those planes. But that will not do: We would weigh all points evenly, which would apparently be suboptimal if some of them are localized already with high accuracy, and others are extremely inaccurate. Intuitively, the directions with high inaccuracy should have higher "weight" than the other points, so another thing to add to our penalty function are weights that should depend on the "imprecision" of the corresponding point's position estimate. This yields, all in all, a nice quadratic function that we need to optimize with certain side conditions.

These side conditions might be, e.g., visibility of all points, movement constraints due to limited physical mobility of our camera, etc. In our case, we focus on the point visibility constraint. It can easily be converted into some inequalities, but unfortunately, we find ourselves confronted with a quadratic programming problem with non-convex side conditions, which is probably not totally trivial to solve. It would be a lot nicer if it were possible to formulate the visibility side condition in form of a equality constraint, because this would allow us to use the concept of Lagrange multipliers.

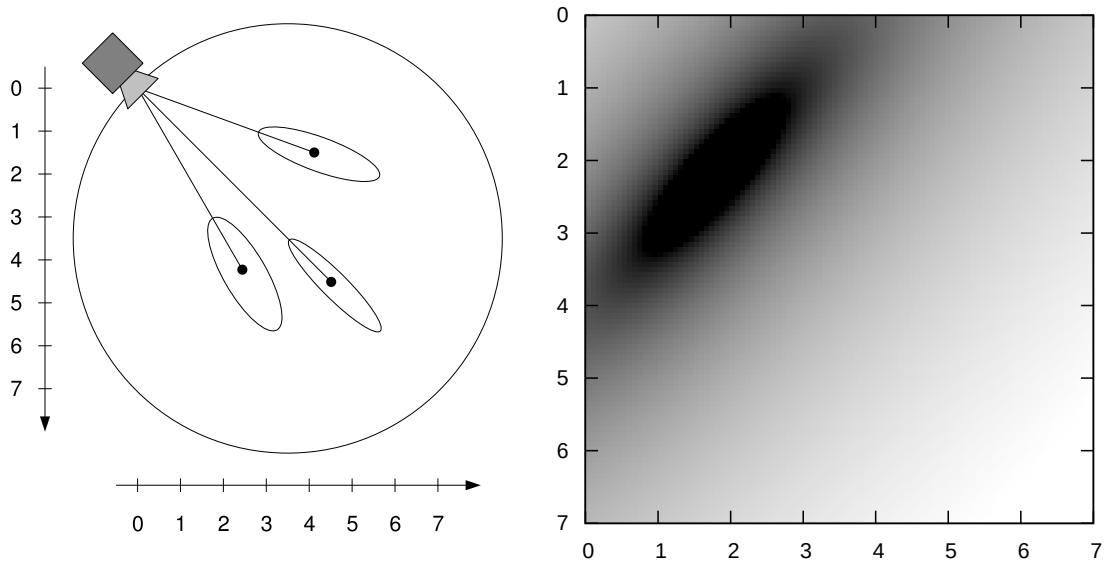

Figure 3.11: An illustration of the basic idea of our approach, demonstrated for a 2-D reconstruction problem. In the left image, we show an example situation, where the camera in the upper left corner has taken an image of the scene, and the reconstruction process leads to the point positions and covariances shown. The right image shows the sum of squared distances to the principal covariance planes, with a logarithmic scale, where light gray levels correspond to high distance, and dark levels to low distance.

3.1.2.5.1 Mathematical Formulation Now that we have given an informal description of our approach, it is time to move on to the mathematical treatment of the problem. As mentioned above, the first step is the determination of the covariance ellipsoid axes for each point y_i. This is equivalent to determining the eigenvectors and eigenvalues of the associated 3×3 covariance matrix $P_{y_i y_i}$, which can conveniently be done by computing the singular value decomposition of that matrix. For each point y_i, we denote the 3 normalized, orthogonal axis vectors by $n_{i,1}, n_{i,2}, n_{i,3}$, assuming that these vectors have been sorted according to their associated eigenvalues in descending order. Those eigenvalues will be denoted by $\lambda_{i,1}, \lambda_{i,2}, \lambda_{i,3}$. With this information, we can further determine values $d_{i,j}$ such that

$$n_{i,j} \cdot x - d_{i,j} = 0 \tag{3.8}$$

finally describes a plane with normal vector $n_{i,j}$ passing through y_i in Hessian Normal Form in 3-D Euclidean space.

The Hessian Normal Form of a plane has the convenient property that substituting any point $\bar{x}$ into the expression $n_{i,j} \cdot x - d_{i,j}$ will yield the distance of $\bar{x}$ to that plane. The value will be positive or negative indicating which side of the plane $\bar{x}$ is located on. Using this knowledge, we can already formulate our penalty function as follows:

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{3} w_{i,j} (x^T n_{i,j} - d_{i,j})^2 = x^T A x - 2 b^T x + c \tag{3.9}$$

Where A, b, c are defined as follows:

$$A := \left(\sum_{i=1}^{n} \sum_{j=1}^{3} w_{i,j} n_{i,j} n_{i,j}^T \right), \quad b := \left(\sum_{i=1}^{n} \sum_{j=1}^{3} w_{i,j} d_{i,j} n_{i,j}^T \right), \tag{3.10}$$

$$c := \sum_{i=1}^{n} \sum_{j=1}^{3} w_{i,j} (d_{i,j})^2 \tag{3.11}$$

It is clear that c can be left out for minimization purposes, so it will be dropped in future references to f. Here, $w_{i,j}$ is the weight assigned to the axis $n_{i,j}$, the choice of which will be discussed later.

Specifying the side condition is straightforward: If m is the center point of the sphere that the camera can move on, and r is the radius of that sphere (which has been determined by whatever means), then

the simple equation

$$g(x) = 0 \quad \text{with} \quad g(x) = (x-m)^T \cdot (x-m) - r^2 \tag{3.12}$$

defines the sphere surface. Applying the principle of Lagrange multipliers, we arrive at the following equations for finding candidates for extremal points of $f(x)$ on $g(x) = 0$:

$$\nabla f(x) = \mu \nabla g(x) \quad \Leftrightarrow \quad Ax - b = \mu(x - m) \tag{3.13}$$

We can simplify this somewhat by looking at functions $f'(x) = f(x+m), g'(x) = g(x+m)$ that are just shifted versions of f and g. This will change above equations as follows:

$$A(x+m) - b = \mu x \Leftrightarrow Ax + Am - b = \mu x \tag{3.14}$$

Defining $b' = -(Am - b)$, the equation finally becomes

$$Ax - b' = \mu x \tag{3.15}$$

Solving this problem is not altogether trivial. Fortunately, the problem of optimization of quadratic functions on a sphere surface has already been researched thoroughly. It is briefly discussed, e.g., as a special case in Hager's paper [Hag01] on quadratic optimization within a sphere. Using the explanation there, we can devise a simple algorithm that allows us to compute the desired optimum. We explain the basic idea here.

If we rearrange terms in the last equation, we see that x can be computed depending on μ:

$$x = (A - \mu I)^{-1} b' \tag{3.16}$$

The problem is then one of finding the right value of μ, such that the constraint $|x| = r$ is satisfied. Let ϕ_i be the eigenvectors of A, and λ_i the corresponding eigenvalues, sorted ascending. It is known that $(A - \mu I)^{-1}$ has the same eigenvectors ϕ_i as A, with eigenvalues $1/(\lambda_i - \mu)$. This observation allows us to express the right side of above equation as

$$\sum_{i=1}^{3} \frac{\beta_i}{(\lambda_i - \mu)} \phi_i, \tag{3.17}$$

where $\beta_i = \phi_i^T b$. If this representation is combined with the length constraint, we arrive at the following equation:

$$\sum_{i=1}^{3} \frac{\beta_i^2}{(\lambda_i - \mu)^2} = r^2 \tag{3.18}$$

This is essentially a polynomial of degree 6, which is in general not possible to solve. However, it can be shown that $\mu < \lambda_1$ must hold for a solution. We can also see that the left side of above equation is strongly convex. With this knowledge, it is possible to compute upper and lower bounds of μ, between which we can search for the solution using a bisection algorithm.

The bounds can be computed as follows: We have

$$\sum_{i=1}^{3} \frac{\beta_i^2}{(\lambda_i - \mu)^2} \leq \sum_{i=1}^{3} \frac{\beta_i^2}{(\lambda_1 - \mu)^2} \tag{3.19}$$

on the one hand, and

$$\sum_{i=1}^{3} \frac{\beta_i^2}{(\lambda_i - \mu)^2} \geq \sum_{i \in \mathcal{E}_1} \frac{\beta_i^2}{(\lambda_1 - \mu)^2} \tag{3.20}$$

on the other hand, where $\mathcal{E}_1$ is the set $\{i \mid \lambda_i = \lambda_1\}$. Both inequalities can be used to compute the bounds. One last difficulty are the so-called degenerate cases, where the boundary computation will fail. They correspond to cases where b' is orthogonal (or close to orthogonal) to the eigenvectors corresponding to the eigenvalues from $\mathcal{E}$. Fortunately, it is possible to use a simple alternate computation to compute the optimum.

3.1.2.5.2 Choice of Weights One question that remains to be answered is the choice of the weights $w_{i,j}$ that are assigned to the planes in our scenario. So far, we have only explained that those weights should somehow be connected to the uncertainty of the point — now we will actually explain how we choose the weights and why we do so.

The first idea one might come up with is weighing only those planes corresponding to the highest uncertainty direction of a point. Thus, our method would be equal to trying to maximally reduce the worst uncertainties in the estimation for each point. The rule for choosing weights can be formulated as follows:

$$w_{i,j} = \begin{cases} 1 & j = 1 \\ 0 & j \neq 1 \end{cases} \tag{3.21}$$

One flaw of this idea is obvious: It would weigh all planes evenly. This is, of course, not optimal, because it might be that some of the points we are looking at are already localized very well, while others are localized very bad.

This observation leads to the following, slightly modified rule:

$$w_{i,j} = \begin{cases} \lambda_{i,j} & j = 1 \\ 0 & j \neq 1 \end{cases} \tag{3.22}$$

Instead of choosing the weight 1 for each of the planes corresponding to the highest uncertainty, we choose as weight the eigenvalue corresponding to that plane. This is justified by the fact that the eigenvalue can be interpreted directly as measure of uncertainty in the direction of the associated eigenvector.

The last rule for choosing weights is already a clear improvement, but we can still see one problem: What if the eigenvalues associated with a point are very close together, or even equal? We might, e.g., think of a case where the uncertainty ellipsoid looks similar to a disc, which would happen when two eigenvalues are of equal size and very big compared to the last eigenvalue. Another interesting case is when the ellipsoid is sphere-shaped, meaning that all eigenvalues are equal. The solution to these problems is simple: We use the rule

$$w_{i,j} = \lambda_{i,j}. \tag{3.23}$$

To see why this rule helps with the problems outlined above, let us think about an example: Let $\lambda_{i,1} = \lambda_{i,2} = 1, \lambda_{i,3} = 0$. In that case, optimal camera positions are characterized by being contained in the planes corresponding to $n_{i,1}$ and $n_{i,2}$. The best camera positions are on the intersection of those planes, and thus on a line, which makes sense intuitively: By taking a measurement of the point from some place on the line, we can achieve maximum reduction of the two worst uncertainties. In the case where $\lambda_{i,1} = \lambda_{i,2} = \lambda_{i,3} = 1$, the best camera position would be the intersection of the three planes, thus we would consider the point itself as optimal position. In our evaluation function, this would mean that we would simply try to position the camera as close to the point as possible, which is an acceptable strategy.

Note that the last rule for choosing weights means that the computation of the matrix A is specifically simple, we have

$$A := \left(\sum_{i=1}^{n} P_{y_i y_i} \right). \tag{3.24}$$

3.1.2.6 Results

We have tested our approach in a simulated environment, where in each step, an optimal camera position is computed, and the camera position is set accordingly for the reconstruction step. We have compared the approach to a randomized viewpoint selection, and found that the results are significantly better.

Especially in the first steps of the reconstruction process, the advantage of using our view planning approach is striking. Figure 3.12 shows a diagram of our results. As a measure of overall estimation uncertainty, we have used the trace of the point covariances. This corresponds to the sum of eigenvalues, and thus seems to be a good measure. To make sure that the results from randomized view planning are not biased, we have built the average of the results of 10 reconstruction runs.

Also, a comparison of the different weight choosing schemes has been performed. It turns out that the final method really is the one that works best, as can be seen in the diagrams of Figure 3.13.

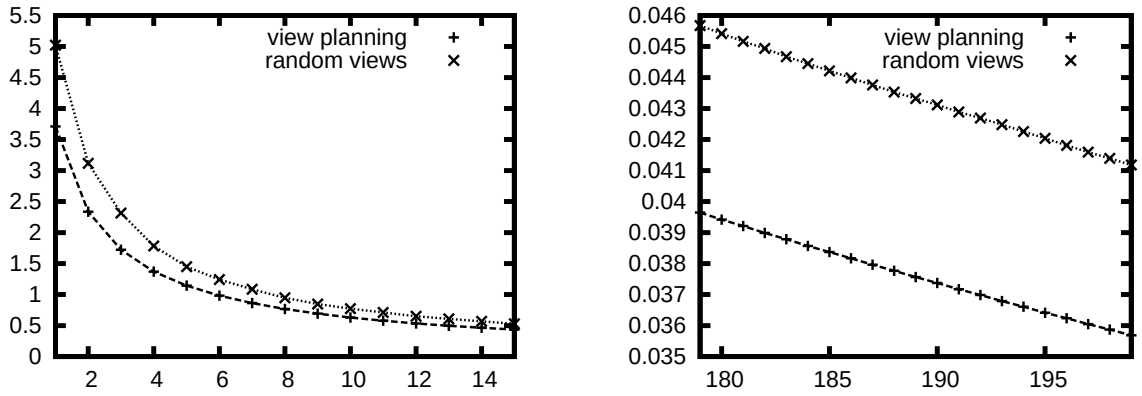

Figure 3.12: Traces of the covariance matrices produced with our view planning method, and with a randomized view point choice. The left diagram only show the first 15 steps, where the gain from using our method is most significant. The right diagram shows the covariance trace for steps 180 to 200, where the difference is smaller, but our strategy still performs better. The data shown for the random viewpoint selection has been obtained by averaging over 10 runs.

The evaluation has also been tried using other measures of uncertainty. In one case, this lead to an interesting observation: When using the spectral norm, the randomized view point selection still performed worse than our view planning method in the beginning. But surprisingly, after some time, the norm of the covariance matrix achieved with the randomized planner becomes smaller than that achieved with our view planning scheme.

The explanation for that phenomenon is simple: The spectral norm of a symmetric matrix is equal to its biggest eigenvalue. Consequently, this means that the worst uncertainty after using randomized views is better than after using view planning. Since the trace of a matrix can be interpreted as the sum of eigenvalues of that matrix, we can deduce the following conclusion: While the highest uncertainty produced with our algorithm might be worse, the average uncertainty is better, and also by an significant amount.

3.2 Discussion

The advantage of our approach versus other methods is that it is computationally very simple, and the view planning recommendation can be computed very efficiently. Most other approaches that rely on more complicated optimality measures also have to use expensive exhaustive sampling techniques, which leads to much higher complexity.

However, we have made some strongly simplifying assumptions. We totally ignore the uncertainty of the sensor. We ignore visibility problems, and we are only focusing on computing camera positions, while not considering the camera angle at all. To solve problems of visibility and self-occlusion, we hope to combine our approach with some other methods of Next Best View planning that have already been developed and can be used to assure that the object under consideration is explored completely.

We have constrained the sensor to a very simple surface, which is convenient mathematically, but also a too strong limitation for some applications. It would be interesting to constrain the camera position to mathematical bodies instead of surfaces. We might, e.g., allow the camera positions to be placed between an inner and an outer sphere. We could consider "cutting" parts of spheres by specifying a plane, and requiring that the camera is placed on a specific side of a plane. There are many possibilities to allow more general side conditions.

The method developed herein is for now constrained to situations where the camera position can be controlled directly, and the camera is able to "jump" to a recommended position between frames. It remains to examine how well it performs in settings where only direction indications are given.

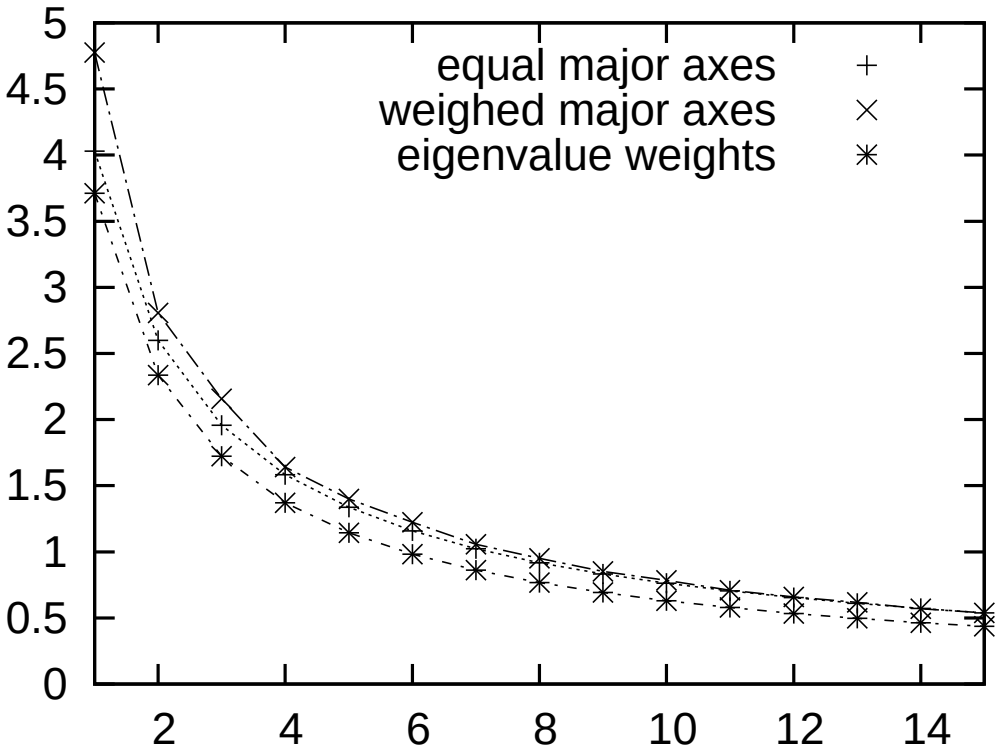

Figure 3.13: The three weight choice strategies in comparison: The weighing using the eigenvalues clearly performs best.

3.3 Object Ontology Representation - Long-Term Memory (Experience)

In collaboration with WP2, we implemented an initial version of the *ontology* (see also D4 chapter 3.2 for more details on the object representation).

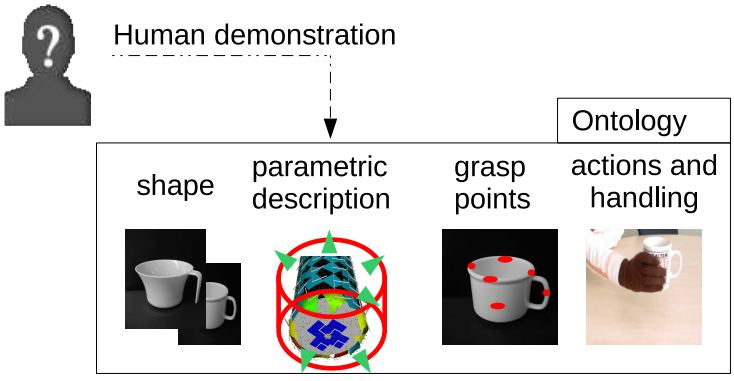

Figure 3.14: Ontology *Layer* in GRASP.

The ontology stores the different representation of a mission relevant object as depicted in Fig. 3.14. The basic representation of an object is its geometric representation as it is perceived by the sensor system. We store a complete shape representation of all mission relevant objects in our ontology and we use this information to improve the accuracy of the reconstructed objects in the environment. The registration is done together with WP4 and our current implementation is part of the deliverable D6 chapter 6. Since the sensor provides only noisy and partial information about the object, we use the shape representation

in the ontology to complete the missing parts of the object and to remove the noise from the sensor perception. This steps allows an identification of the reconstructed structure and a match to the set of the known objects in the ontology.

Our goal is to grasp also unknown objects if their shape is similar to already known geometries. An example can be here a cup of a different diameter, which was not previously handled. The ontology includes a parametric description of the object in the database. This parametric description is a result of a segmentation of the object into basic shape primitives. These primitives include also an additional descriptor indicating the allowed deformation of the basic segment. In case of a cup, a cylindrical descriptor with varying diameter can be a representation in the ontology. We currently work on a matching algorithm that will allow us to fit the current observation into this deformable description. A success in this step will allow us a simple classification of new object based on a segmentation into known basic parametric shapes in the ontology.

The ontology saves also information about the grasping points that were defined in previous missions or given as an apriori knowledge to the system. This information is currently not relevant for the surprise detection but it will provide valuable hints for manipulation of the object.

An important information stored in the ontology are the actions that are known for a specific object. In the current stage of the project, we store in this part the way how the object is removed from the supporting surface and how it is placed back on such a surface. Additionally, we detect constrains on the motion of the object while it is manipulated. It is not feasible and also not necessary to store all currently known trajectories for an object. We decided to inverse the problem and store which parameters stay fixed while an object is manipulated. We start with all the 6 DoF *frozen* at the beginning and allow motions in parameters that were observed during the human manipulation. A new allowed motion in currently *frozen* degree of freedom represents a surprise for the system. An example is a manipulation of the cup always in an upright position (because it contains liquid) and suddenly the system observes a free motion in all parameters which is new to it.

Chapter 4

Mismatch-Based Surprise Detection

According to the Figure 3.5, the first stage of the surprise detection got implemented. We tested an image-based approach to detect mismatches in the environment

The Maximum-Likelihood (ML) estimates for the mean and the covariance of the Gaussian distribution are point estimates which give one model which describes the statistical properties of the sample data. However, the estimates still deviate from their true values and there are other less probable parametrization for the Gaussian distribution. Unlike ML estimation, Bayesian inference takes into account all possible models and puts priors over the parameters of the probability distribution of the sample data. In [IB06], a Bayesian framework was presented for modeling and quantifying human surprise in a mathematical way. Inspired by that, we propose in the following a scheme for Bayesian visual surprise detection based on the probabilistic concept for view synthesis.

For surprise detection the set of samples consists of seven RGB-tripels from reference images captured in the past and an additional color value from the current observation. The virtual camera and the real camera capturing the current image have identical position and orientation. Hence, accurate localization of the cognitive system's camera is crucial for robust surprise detection. Similar to the processing of color information in the human visual system ([EZ97]), we compute from each RGB reference image a luminance signal and two color opponency signals (red-green and blue-yellow), respectively. Thus, surprise detection does not have to be performed jointly in RGB-space but can be done independently in three decoupled pathways. For the luminance of a pixel in the virtual image the following likelihood function for a univariate Gaussian model results:

$$p(\mathbf{X}_\mathrm{I} \mid \mu_\mathrm{I}, \lambda_\mathrm{I}) = \prod_{k=1}^{7} \left(\frac{\lambda_\mathrm{I}}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\tfrac{\lambda_\mathrm{I}}{2}\left(x_{\mathrm{I},k} - \mu_\mathrm{I}\right)^2\right\}. \tag{4.1}$$

$\mathbf{X}_\mathrm{I} = [x_{\mathrm{I},1}, \ldots, x_{\mathrm{I},7}]$ is a vector containing the luminance samples from the reference images. μ_I denotes the true luminance value at the pixel in the virtual image which is also the mean of the Gaussian distribution. For the choice of the prior distributions it is more convenient to use the precision λ_I, which is defined by the reciprocal of the variance ($\lambda_\mathrm{I} \equiv \frac{1}{\sigma_\mathrm{I}^2}$). Assuming that the mean is given by its ML estimate $\mu_{\mathrm{I,ML}} = \sum_{k=1}^{7} x_{\mathrm{I},k}$, we put a prior over the precision which has the form of a gamma distribution

$$p(\lambda_\mathrm{I}) = \frac{1}{\Gamma(a_0)} b_0^{a_0} \lambda_\mathrm{I}^{a_0 - 1} \exp\left\{-b_0 \lambda_\mathrm{I}\right\}. \tag{4.2}$$

Here $\Gamma(a_0) = \int_0^\infty t^{a_0 - 1} \exp\left\{-t\right\} \mathrm{d}t$ denotes the gamma function which serves as a normalization constant. The shape of the distribution thus depends on the two hyperparameters a_0 and b_0.

With Bayes' formula the posterior distribution of the precision given the sample data is calculated from the likelihood function and the prior up to a scaling factor by

$$p(\lambda_\mathrm{I} \mid \mathbf{X}_\mathrm{I}) \propto p(\mathbf{X}_\mathrm{I} \mid \mu_{\mathrm{I,ML}}, \lambda_\mathrm{I}) \cdot p(\lambda_\mathrm{I}) \tag{4.3}$$

Note that the posterior is again a gamma distribution with the hyperparameters $a = a_0 + \frac{7}{2}$ and $b = b_0 + \frac{1}{2}\sum_{k=1}^{7}\left(x_{\mathrm{I},k} - \mu_{\mathrm{I,ML}}\right)^2$ which depend on the sample data. The kind of prior whose posterior has

the same functional form is called a conjugate prior. The advantage of conjugate priors is that their posteriors can again be used as priors for further analysis.

Now we augment our set of luminance samples by the luminance value which the current observation of the cognitive technical system provides ($\mathbf{X}'_{\mathrm{I}} = [x_{\mathrm{I},1}, \ldots, x_{\mathrm{I},7}, x_{\mathrm{I,ob}}]$). The posterior distribution over λ_{I} is then calculated by

$$p(\lambda_{\mathrm{I}} \mid \mathbf{X}'_{\mathrm{I}}) \propto p(x_{\mathrm{I,ob}} \mid \mu_{\mathrm{I,ML}}, \lambda_{\mathrm{I}}) \cdot p(\lambda_{\mathrm{I}} \mid \mathbf{X}_{\mathrm{I}}) \tag{4.4}$$

which results in a gamma distribution with the hyperparamters $a' = a + \frac{1}{2}$ and $b' = b + \frac{1}{2}\left(x_{\mathrm{I,ob}} - \mu_{\mathrm{I,ML}}\right)^2$.

In [IB05], the Kullback-Leibler divergence (KLD) as the difference between the posterior distribution over the model parameters given a new observation and the prior distribution is proposed as a quantitative measure for surprise

$$\begin{aligned} &\mathrm{KLD}\left(p(\lambda_{\mathrm{I}} \mid \mathbf{X}'_{\mathrm{I}});\ p(\lambda_{\mathrm{I}} \mid \mathbf{X}_{\mathrm{I}})\right) = \\ &\quad = \int_{\lambda_{\mathrm{I}}} p(\lambda_{\mathrm{I}} \mid \mathbf{X}'_{\mathrm{I}}) \log\left(\frac{p(\lambda_{\mathrm{I}} \mid \mathbf{X}'_{\mathrm{I}})}{p(\lambda_{\mathrm{I}} \mid \mathbf{X}_{\mathrm{I}})}\right) \mathrm{d}\lambda_{\mathrm{I}}. \end{aligned} \tag{4.5}$$

It can be shown that the KLD between two gamma distributions is a function of their hyperparameters

$$\begin{aligned} &\mathrm{KLD}\left(p(\lambda_{\mathrm{I}} \mid \mathbf{X}'_{\mathrm{I}});\ p(\lambda_{\mathrm{I}} \mid \mathbf{X}_{\mathrm{I}})\right) = \\ &\quad = a \cdot \log\left(\frac{b'}{b}\right) + \log\left(\frac{\Gamma(a)}{\Gamma(a')}\right) + b \cdot \frac{a'}{b'} \\ &\qquad + (a' - a) \cdot \psi(a') \end{aligned} \tag{4.6}$$

where $\psi(a') = \frac{\frac{\mathrm{d}}{\mathrm{d}x}\Gamma(x)\big|_{x=a'}}{\Gamma(a')}$ is the digamma function. We evaluate (4.6) for each pixel in the virtual image and as a result get a pixel-wise surprise trigger.

For fast and parallel calculation of pixel-wise surprise triggers, modern graphics hardware can be used. Since common graphics APIs like Direct3D and OpenGL do not allow for the direct calculation of gamma and digamma functions, (4.6) has to be modified. In our pixel shader implementation, we approximate the gamma function using the Stirling series

$$\Gamma(z) \approx \sqrt{\frac{2\pi}{z}} \cdot \left(\frac{z}{e}\right)^z \cdot \exp\left(\frac{1}{12z} - \frac{1}{360z^3} + \frac{1}{1260z^5}\right) \tag{4.7}$$

where $e = 2.71828\ldots$ is the Euler's number.

The digamma function is approximated by

$$\psi(z) \approx -\frac{1}{z} - \gamma + \sum_{n=1}^{5}\left(\frac{1}{n} - \frac{1}{z+n}\right) \tag{4.8}$$

where $\gamma = 0.57721\ldots$ denotes the Euler's constant. With the approximations in (4.7) and (4.8), we obtained the surprise trigger in Fig. 4.1 which was calculated on the graphics hardware by a pixel shader implemented in Direct3D. For better visualization the surprise trigger was amplified by a factor of 10. For a static observation, we measured an average frame rate of 14 frames per second (at a resolution of 320×240 pixels). Since the pose is not that accurate in case of automatic localization, the surprise trigger is higher in regions where indeed no changes occurred compared to Fig. 4.1. However, there is still a pronounced region around the missing glasses with high surprise trigger compared to the rest of the surprise map.

4.1 Self-Localization in the Environment

The surprise detection in the Mismatch-Based Layer of the GRASP project requires a highly accurate registration of the current perception to the coordinate frame from the knowledge base of the system. We developed a localization system in collaboration with CoTeSys project (German DFG funded) for highly accurate real-time localization.

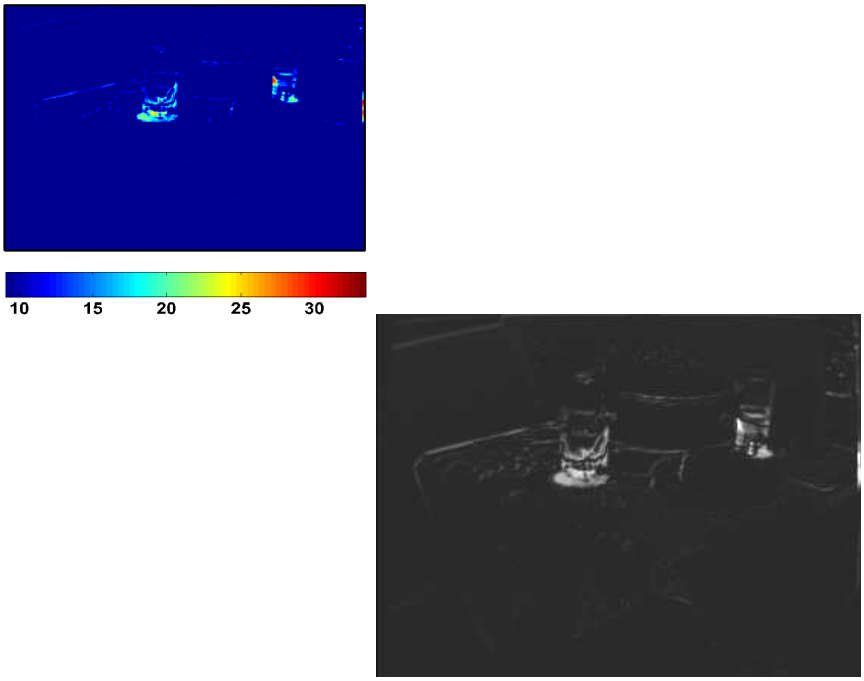

Figure 4.1: (a) Surprise trigger obtained from the pixel-wise calculation of the KLD between prior and posterior distribution over the precision of the color samples. (b) Approximated surprise trigger computed on the graphics hardware.

Our localization method provides acceptable results with respect to position and orientation as long as the tracker finds enough matches between the images. However, it fails as soon as there are too few tracked landmarks for pose estimation. This may happen if the torsional moment of the camera is too high, so that all references leave its field of view or just the tracker's search range. After a simple reboot, the relation to the prior run can also be lost. Even if all detected features were saved on a hard drive, the cameras could not be registered within the prior world coordinate frame, as soon as the robot moves outside the known trajectory. We need to register the new sequence with respect to the reference coordinate system in order to establish a relationship to the previously acquired data. Since we do not use external markers as reference, which could be used to determine the origin of the reference frame, we need to initially specify an arbitrary origin. All the information which is necessary t o refer to this origin, whenever required, has to be stored – a so called "snapshot" has to be taken. In this chapter, we present two different approaches to solve this problem.

4.1.1 Homing based on three images

In our first homing approach (further on Homing1) we use RVGPS for extrinsic parameter estimation. RVGPS is now used to estimate the rotation and translation between the current and the reference frame. We only need $I_{1.1}$ and its initialized points of interest (POIs), provided by SURF. This 3D structure forms the so called snapshot of the origin. SURF and KLT use different detectors, hence the stereo-registration method used by the visual localization scheme cannot be used between $I_{1.1}$ and $I_{2.1}$. Instead the POIs are initialized by a so called *structure from motion* approach: The distance of the camera between $I_{1.1}$ and $I_{1.2}$, which is estimated by our visual localization routine, is used as baseline for stereo triangulation. Once the SURF-features are initialized, we only need at least 3 SURF correspondences between $I_{1.1}$ and $I_{2.1}$ to apply RVGPS in order to estimate the six degrees of freedom (DOFs). Of course, the robustness and accuracy rapidly increases if more matching features are available. Thus, big parts of the same scene should be seen by these three images to ensure that enough matches are found. Otherwise one can also use more than one image in S_1 to initialize more POIs in $I_{1.1}$. The more points are available, the higher is the probability that correspondences in $I_{2.1}$ are found and the higher is also the accuracy of the motion estimation. Fig. 4.1.2 illustrates the principle of the Homing1 algorithm, which needs only 3 images.

4.1.2 Homing based on four images

The Homing1 variant has shown that its results strongly depend on the accuracy of the POIs' structure. Our second approach has been developed with the aim of avoiding that inconvenience by not using RVGPS for localizing $I_{2.1}$ with respect to $I_{1.1}$. Instead we are looking for an optimal matching of two 3D structures in the different coordinate frames of S_1 and S_2 in order to estimate the six DOFs. To calculate two corresponding structures for our second homing algorithm (Homing2) we need for each sequence S_1 and S_2 two images, their extrinsic parameters and the SURF correspondences in all 4 images. The extrinsic parameters for structure initialization in $I_{1.1}$ resp. $I_{2.1}$ are estimated in the same way as in the Homing1 method - by the visual localization routine and subsequent stereo triangulation. Using Arun's algorithm ([AHB87]) we can calculate the transformation matrix between the two frames of S_1 and S_2. The result of this method is obviously more robust, because we do not estimate the transformation matrix and the structure of the point set at the same time, like in Homing1. On the other hand we need to find SURF matches in 4 images, which is more problematic than with 3 images due to the smaller common feature intersection. Fig. 4.1.2 depicts the principle of the Homing2 algorithm based on 4 images.

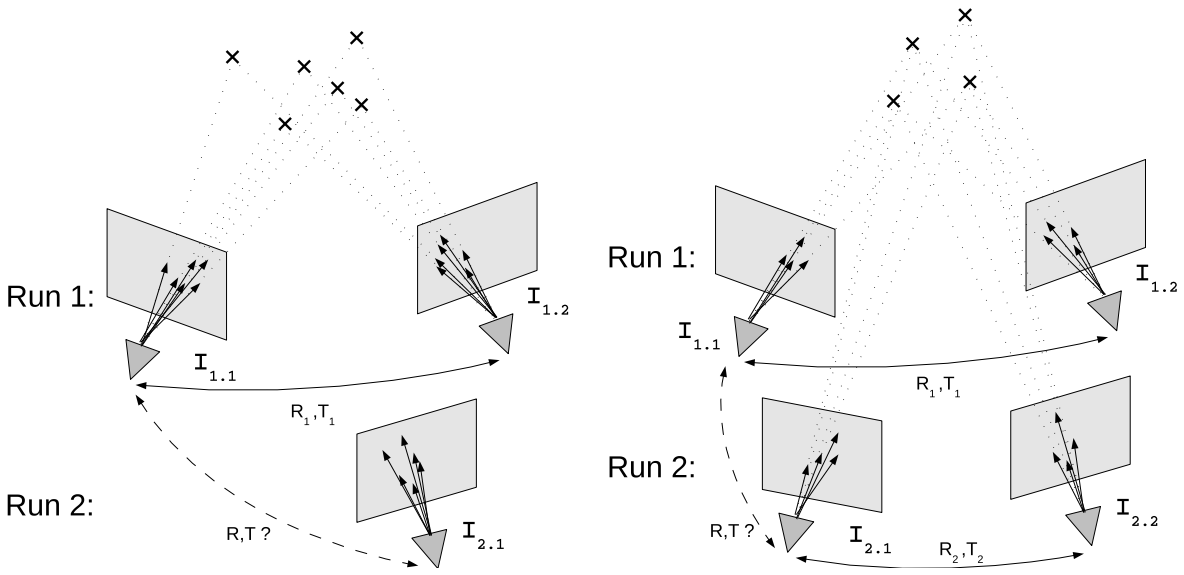

Figure 4.2: *Fig. 4.1.2*: The transformation between sequence 1 and 2 is estimated using the RVGPS algorithm. Thus, only one image of run 2 is necessary. *Fig. 4.1.2*: The point structure in run 2 is initialized independently of the reference sequence (run 1), so that a higher accuracy is provided at the cost of the robustness (usually fewer common features are found).

Which algorithm to use therefore strongly depends on the application and the scene. Since the errors do not vary much, the more robust but less accurate Homing1 algorithm is preferable in most cases.

4.2 Synthesis of the Expectation (Mismatch-Based Surprise)

In this section, we show some test results of our visual navigation algorithm and the visual output obtained from our image-based modeling technique applied to a household scene. We further tested our methods for visual homing and surprise detection. Fig. 3.6 shows the acquisition of an image sequence S1 with a stereo camera head (640x480 pixels) mounted on a Pioneer 3-DX robot. The robot went along an approximately circular trajectory around a table set with household objects like glasses, plates etc. The stereo camera was looking towards the objects and captured 213 pairs of images. The visual localization of image sequence S1 in a world coordinate frame is illustrated in Fig. 4.3.

In order to test our algorithm for surprise detection we captured another image sequence S2 on a trajectory which was close to the first one but not identical. We changed the scene before by removing the two glasses. The task of the cognitive system is to detect these changes. This is usually quite challenging for an artificial cognitive system due to the difficulties involved with building up an internal representation

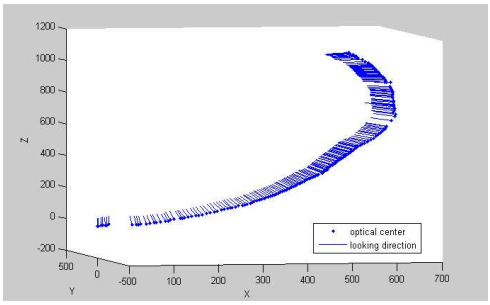

Figure 4.3: 3D plot of localization results.

of the glasses. One image from S2, which is the current observation of the cognitive system, was localized with respect to the world coordinate system of S1.

Figure 4.4: (left) Observation of the cognitive system. (right) Virtual image rendered from a set of reference images from S1 at the current position of the observing camera.

The observation is depicted in Fig. 4.4left) together with a photo-realistic virtual image rendered from reference images which were selected only from S1 (Fig. 4.4right). The virtual image was rendered with our method described in Section IV. Note that there is no real camera image from S1 which was acquired exactly at the position of the observation. Applying our algorithm from Section V on the luminance signals of the two images, we obtained the surprise trigger shown in Fig. 4.5. The surprise trigger was calculated in MATLAB using (7). The figure clearly shows a region of high KLD values around the missing glasses.

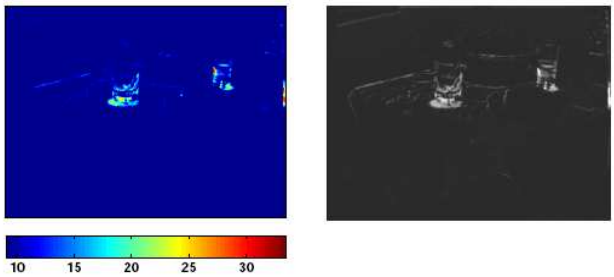

Figure 4.5: Surprise trigger obtained from the pixel-wise calculation of the KLD between prior and posterior distribution over the precision of the color samples

Chapter 5

Conclusions and Future Work

The initial representation developed in the current workpackage is the first testbed how to represent the knowledge in the GRASP project and how to define action representations that are necessary for a successful surprise detection. We focused on the implementation of the parts of the ontology that are necessary for a successful surprise detection and on implementation of low-level algorithms that provided the required information about the position of the system in the world this information is essential for a successful prediction of expectation. In our definition actions are part of the object description and we observe the changes induced by the human or other agents in the world. We try to predict them in our to be developed surprise detection framework and important part of the task is the estimation of the ego position in the world and the segmentation of mission relevant objects. We provided an initial framework to solve this tasks that allows us to detect changes in the environment and to segment out structures that need to be manipulated.

Our next goal is to focus more on the representation of actions in the local environment and to include them in the predictions of the system. We started already the work on registration of generic shape descriptions that will allow a classification of objects to a global category, e.g., a cup. This will allow to provide a-priori suggestion about the manipulation capabilities of an object which may still be unknown to the system but where the ontology will help to generate *suggestions* how to handle it based on the similarities to know objects in the *object ontology*. A fusion of information about human gestures from WP1 will further enhance the knowledge acquisition in our framework.

References

[BCD+05] Darius Burschka, Jason J. Corso, Maneesh Dewan, William Lau, Ming Li, Henry Lin, Panadda Marayong, Nicholas Ramey, Gregory D. Hager, Brian Hoffman, David Larkin, and Christopher Hasser. Navigating inner space: 3-d assistance for minimally invasive surgery. *Robotics and Autonomous System*, 2005.

[BH74] A.D. Baddeley and G.J. Hitch. *Working Memory*, volume 8. New York: Academic Press, 1974.

[BLT+05] Darius Burschka, Ming Li, Russell Taylor, Gregory D. Hager, and Masaru Ishii. Scale-invariant registration of monocular endoscopic images to ct-scans for sinus surgery. *Medical Image Analysis*, 9(5):413–439, October 2005. Best Paper Award at MICCAI 2005.

[Bra88] M.E. Bratman. *Intentions, plans, and practical reason*. Cambridge, MA: Harvard University Press, 1988.

[CK88] Cregg K. Cowan and Peter D. Kovesi. Automatic sensor placement from vision task requirements. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(3):407–416, 1988.

[CL04] S. Y. Chen and Y. F. Li. Active viewpoint planning for model construction. In *ICRA*, pages 4411–4416. IEEE, 2004.

[Cow05] N. Cowan. Working memory capacity. In *New York, NY: Psychology Press*, 2005.

[CYJ02] D. Cobzas, K. Yerex, and M. Jagersand. Dynamic Textures for Image-Based Rendering of Fine-Scale 3D Structure and Animation of Non-Rigid Motion, 2002.

[Dey01] A.K. Dey. Understanding and Using Context. In *Personal and Ubiquitous Computing*, volume 5(1), pages 4–7, 2001.

[DRMS07] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):1052–1067, 2007.

[FH04] P. Felsenszwalb and D. Huttenlocher. Efficient Belief Propagation for Early Vision. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, Washington, DC, USA*, pages I261–I268, 2004.

[GH98] M. Garland and P.S. Heckbert. Simplifying Surfaces with Color and Texture Using Quadric Error Metrics. In *Proc. IEEE Conf. on Visualization, Durham, USA*, pages 263–269, 1998.

[Hag01] William W. Hager. Minimizing a quadratic over a sphere. *SIAM J. on Optimization*, 12(1):188–208, 2001.

[KD95] Kiriakos N. Kutulakos and Charles R. Dyer. Global surface reconstruction by purposive control of observer motion. *Artif. Intell.*, 78(1-2):147–177, 1995.

[KPE99] W. Kintsch, V. Patel, and A. Ericsson. The role of long-term working memory in text comprehension. In *Psychologia*, volume 42, page 186198, 1999.

[MB93] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(5):417–433, 1993.

[MC02] M. Miceli and Castelfranchi. The Mind and the Future. The (Negative) Power of Expectations. In *Theory & Psychology*, volume 12(3), pages 335–366, 2002.

[OP87] A. Ortony and Partridge. Surprisingness and Expectation Failure. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 106–108, 1987.

[RG92] A.S. Rao and M.P. Georgeff. An Abstract Architecture for Rational Agents. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, pages 439–449, 1992.

[Sim05] Robert Sim. Stable exploration for bearings-only slam. In *ICRA*, pages 2411–2416. IEEE, 2005.

[VCDACM06] T. Vidal-Calleja, A.J. Davison, J. Andrade-Cetto, and D.W Murray. Active control for single camera slam. In *IEEE Int Conf on Robotics and Automation, Orlando, May 2006*, 2006.

[WDAN07] Stefan Wenhardt, Benjamin Deutsch, Elli Angelopoulou, and Heinrich Niemann. Active visual object reconstruction using d-, e-, and t-optimal next best views. In *CVPR*. IEEE Computer Society, 2007.

[WF97] Peter Whaite and Frank P. Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(3):193–205, 1997.

[Win] T. Winograd. Architecture of Context. In *Human Computer Interaction*, volume 16, pages 401–419.

[WMS97] R. Reisenzein W.U. Meyer and A. Schützwohl. Towards a Process Analysis of Emotions: The Case of Surprise. In *Motivation and Emotion*, volume 21, pages 251–274, 1997.

[SD98] S. Simhon and G. Dudek. Selecting targets for local reference frames. In Proc. IEEE Int. Conf. on Robotics and Automation, pages 2840–2845, 1998.

[SMB98] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. In *In Proc. of International Conference on Computer Vision, Bombay, January*, 1998.

[FGM90] F.Ferrari, E. Grosso, G. Sandini, and M. Magrassi. A stereo vision system for real-time obstacle avoidance in unknown environment. In Proc. of IEEE International Workshop on Intelligent Robots and Systems IROS'90, pages 703–708, 1990.

[MZC92] J. Mayhew, Y. Zheng, and S. Cornell. The adaptive control of a four-degrees-of-freedom stereo camera head. In *Natural and Artificial Low-level Seeing Systems, The Royal Society, London*, pages 63–74, 1992.

[SB97] S. Se and M. Brady. Vision-based detection of kerbs and steps. In *Eighth British Machine Vision Conference BMVC '97*, pages 410–419, 1997.

[IB05] L. Itti and P. Baldi, "A Principled Approach to Detecting Surprising Events in Video," *in Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, San Diego, USA, 2005, pp. 631-637.

[IB06] L. Itti and P. Baldi, "Bayesian Surprise Attracts Human Attention," *in Adv. in Neural Information Processing Systems*, vol. 19, 2006, pp. 547-554.

[EZ97] S. Engel and X. Zhang, "Colour Tuning in Human Visual Cortex Measured with Functional Magnetic Resonance Imaging," *Nature*, vol. 388, no. 6637, 1997, pp. 68-71.

[AHB87] K. S. Arun, T. S. Huang and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *in IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, 1987, pp. 698-700.